Noname manuscript No.
(will be inserted by the editor)

SemRec: A Personalized Semantic Recommendation Method based on Weighted Heterogeneous Information Networks

Chuan Shi · Zhiqiang Zhang · Yugang Ji · Weipeng Wang · Philiph S. Yu · Zhiping Shi

Received: date / Accepted: date

Abstract In past few years, heterogeneous information network (HIN) analysis has received much attention, and HIN has seen many explorations in the field of data mining. Recommender system is composed of many object types, including users, movies, actors, and interest groups in movie recommendation, for example. It also has the rich relations among object types, which naturally constitutes a HIN. HIN has the comprehensive information integration and rich semantic information. Therefore, it is a promising tool to generate better recommendations. However, the attribute values on links are not considered in HINS. As a result, semantic relations among objects may not be captured by the widely used meta path in HIN, resulting from the existence of rating scores (usually ranging from 1 to 5) between users and items in recommender system. In this paper, we introduce the weighted HIN and weighted meta path concepts to subtly depict the path semantics through distinguishing different link attribute values. In addition, we develop a personalized recommendation method SemRec. It utilizes semantic path to estimate user rating scores on items. Setting meta paths allows SemRec integrates heterogeneous information flexibly. Furthermore it also helps to generate prioritized and personalized weights that represent user preferences on paths. We conduct experiments on three real datasets to show that SemRec can achieve better recommendation performance by flexible information integrating via weighted meta paths. Moreover, extensive experiments validate the benefits of weighted meta paths.

Keywords heterogeneous information network · recommendation, similarity · meta path

1 Introduction

Recently, more and more work have been focused on Heterogeneous Information Network (HIN). In HIN, objects are of different types and links among objects represent different relations [24, 28]. The heterogeneity and rich relation of information network make it a better data representation in many scenarios. As a unique characteristics of HIN, the meta path [29,36], connecting two objects through a

C. Shi
Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University
Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications

Z. Zhang
Ant Financial Services Group

Y. Ji
Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications

W. Wang
University of Southern California

P. S. Yu
University of Illinois at Chicago

Z.Shi
Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University

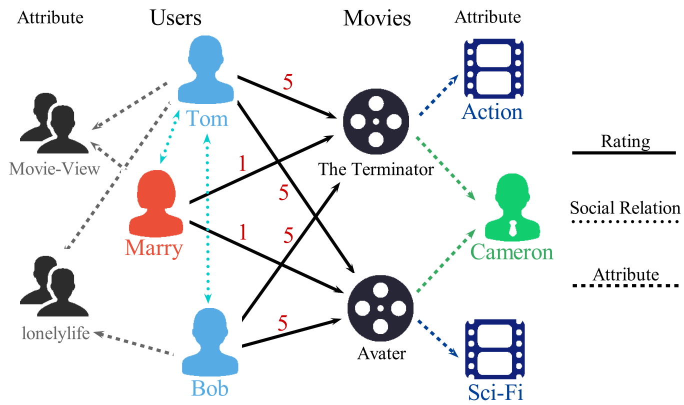


Fig. 1: The objects and relations in movie recommender system is organized as a weighted heterogeneous information network.

sequence of relations between object types, is widely used to exploit rich semantic information. Many meta path based data mining tasks have been done in the past couple of years, including similarity search [23, 29], clustering [30], and classification [7] etc.

More recently, some works [6, 18, 36] have adopted HIN for recommendation, where the objects and their relations in recommender system constitute a HIN. Fig. 1 shows such an example. The HIN not only contains different types of objects in movie recommendation (e.g., users and movies) but also illustrates all kinds of relations among objects, such as viewing information, social relations, and attribute information. Constructing heterogeneous networks for recommendation can effectively integrate all kinds of informations, which can be potentially utilized for recommendation. Moreover, the objects and relations in the networks have different semantics, which can be explored to reveal subtle relations among objects. For example, the meta path "User-Movie-User" in Fig. 1 means users viewing the same movies, and can be used to find the similar users according to viewing records. If we recommend movies following this meta path, it will recommend the movies that are seen by users having the same viewing records with the given user. It corresponds to the collaborative filtering model in essence. Similarly, the "User-Interest Group-User" path can find the similar users with similar interests. This path corresponds to the member recommendation [37]. So we can directly recommend items based on the similar users generated by different meta paths connecting users. Moreover, it can realize different recommendation models through properly setting meta paths. However, this idea faces the following two main challenges.

Firstly, conventional HIN and meta path cannot be directly applied to recommender system. As we know, conventional HIN and meta path do not consider the attribute values on links. However, this movie recommendation network can contain attribute values on links. Concretely, in recommender system, the users can provide a rating score to each movie viewed. The rating scores usually range from 1 to 5 as indicated on the link between user and movie in Fig. 1, where higher score means stronger preference. Ignoring the rating scores may result in bad similarity discovery on users. For example, according to the path "User-Movie-User", Tom has the same similarity with Mary and Bob, since they view the same movies. However, they may have totally different tastes due to different rating scores. In fact, Tom and Bob should be more similar, since they both like the same movies very much with high scores. Mary may have totally different tastes, because she does not like these movies at all. The conventional meta path does not allow links to have attribute values (e.g., rating scores in the above example) [29, 36], and hence it cannot reveal this subtle difference. However, this difference is very important, especially in recommender system, to more accurately reveal relations of objects. So we need to extend existing HIN and meta path for considering attribute values on links. Moreover, the new similarity measures are urgently needed for development.

Secondly, it is difficult to effectively combine information from multiple meta paths for recommendation. As we have said, different types of similar users will be generated through different meta paths, and these different types of similar users will recommend different items. A weight learning method can be designed to combine these recommendations and each path can be assigned with a learned weight preference. A good weight learning method should obtain prioritized and personalized weights. That is, the learned weights can represent the importance of paths, and each user should have personalized weights to embody his preferences on paths. The prioritized and personalized weights are very important for recommendation, since they can deeply reveal the characteristics of users. Much more than this, it makes the recommendation more explainable, since meta paths contain semantics. For example, if a user has high weight preference on the "User-Interest Group-User" path, we can explain that the recommendation results stem from movies viewed by users in the interest groups he joined in. Unfortunately, the personalized weights may suffer from the rating sparsity problem, especially for users with little rating information. The reasons lie in that so many parameters are needed to be learned and rating information are usually not sufficient.

In this paper, we extend HIN and meta path for widely-existing attribute values on links in information networks, and firstly propose the weighted HIN and weighted meta path concepts to more subtly reveal object relations through distinguishing link attribute values. Instead of designing an ad hoc similarity measure for weighted meta paths, we design a novel similarity computation strategy that can make existing path-based similarity measures still usable. Furthermore, the semantic path based personalized recommendation method SemRec is proposed to flexibly integrate heterogeneous information through setting meta paths. In SemRec, we design a novel weight regularization term to obtain personalized weight preferences on paths and alleviate the rating sparsity through employing the consistency rule of weight preferences of similar users. The major contributions of this paper are summarized as follows:

- We propose weighted HIN and weighted meta path to consider attribute values on links in information networks. Furthermore, we propose the similarity measure strategy on weighted meta path.
- We design a novel SemRec method, which not only effectively integrates all kinds of information contained in recommender system but also flexibly represents different kinds of recommendation models through properly setting meta paths. In addition, SemRec can obtain the prioritized and personalized weight preferences on multiple meta paths, which are important for real applications, e.g., user characteristics analysis and recommendation explanation.
- Empirical studies on three real datasets, Douban Movie, Yelp and Douban Book, demonstrate the power of SemRec. SemRec outperforms the state of the arts, especially for cold-start users and items, and the personalized weights learned by SemRec are able to reflect user preferences on paths. In addition, SemRec can achieve better performances with the help of weighted meta paths.

The original work of this study has been published in [26]. This paper significantly extends the original work in the following aspects. (1) As the most important contribution of this study, this paper more detailedly describes the weighted meta path and provides its similarity measure strategy. The added experiments and applications confirm the benefits of the weighted meta path. (2) The added Douban Book dataset furtherly validates the proposed method. (3) The added parameter experiment reveals the characteristics of SemRec. (4) The related work is greatly extended to comprehensively introduce the newest work.

The remainder of this paper is organized as follows. Section 2 introduces the related work, and then the preliminaries for this study are presented in Section 3. In Section 4, we propose the SemRec method in detail. Experiments and analysis are in Section 5. Finally, we conclude the study in Section 6.

2 Related Work

As one of the most popular recommendation approaches, collaborative filtering techniques have attracted a lot of attention from different perspectives. Among different techniques, matrix factorization has shown its effectiveness and efficiency in recommender systems, which factorizes the user-item rating matrix into two low rank user-specific and item-specific matrices, and then utilize the factorized matrices to make further predictions [27]. With the prevalence of social media, more and more research study social recommender system, which utilizes social relations among users. Many researchers utilized trust information among users. Ma et al. [20] fused the user-item matrix with the users' social trust networks by sharing a common latent low-dimensional user feature matrix. Furthermore, they [19] coined the social trust ensemble to represent the formulation of the social trust restrictions. More and more researches began to use the friend relation among users. In [21], the additional social regularization term ensures that the distance of latent feature vectors of two friends with similar tastes become closer. The membership among users are also explored to boost collaborative filtering [37]. Most existing relation-based collaborative filtering methods usually utilize information from single or multiple homogeneous networks which only have single type of nodes and links. However, the widely available attribute information of users and items is useful for recommendations, which is seldom exploited in this kind of methods.

Taking the attribute information of users and items into consideration, Yin et al. [34] proposed a joint probabilistic generative model based on Latent Dirichlet Allocation to mimic user check-in behaviors in a process of decision. By exploiting both the c-occurrence pattern of spatial items and the content of spatial items, Wang el al. [32] paid attention to user interests and the preference of the crowd in the target region, and then proposed a new graphical model called Geo-SAGE. However, these methods just focused on obtaining user profile or item profile but ignored the structural information between users and items.

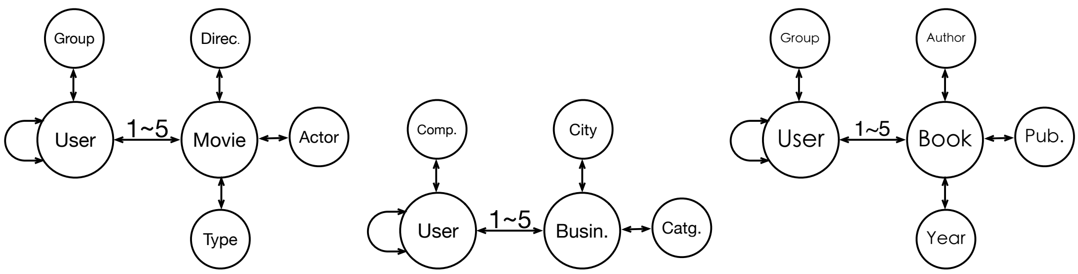
As a simple HIN, bipartite and multipartite graph can be used to show different views of relations between users and items. Links in bipartite/multipartite graph show various views. Zhang et al. [38] utilized user-item-tag tripartite graphs to optimize personalized recommendation in which they integrated the diffusions on user-item and item-tag relation. And considering flexibility of a recommender system, Lee et al. [12] proposed a graph-based multidimensional recommendation method, in which bipartite graph model can represent various relations not only the user-item relation. Considering recommendation problem as link prediction in graph, Li et al. [13] took advantage of structure of user-item bipartite graph and proposed a kernel-based recommendation approach. Later, Personalized PageRank [5] was extended on heterogeneous multipartite graph [11], through utilizing the semantics of different types of nodes and edges. Xie et al. [33] embedded four relational graphs (POI-POI, POI-Region, POI-Time and POI-Word) into a shared low dimensional space and then developed a time-decay method to dynamically compute the users latest preferences based on the embedding of his/her checked-in POIs learnt in the latent space. These methods attempt to integrate more information with bipartite or multi-partite graph. However, as one type of the simplest HINs, bipartite or multi-partite graph fails to fuse more information and richer semantics.

Recently there is a surge of research on heterogeneous information network, in which objects are of different types and links among objects represent different relations [4]. The heterogeneity and rich relation of information network make it a better data representation in many scenarios. As a unique characteristics of HIN, meta path not only represents rich semantics but also can be utilized to integrate different types of informations in HIN. Many meta path based data mining tasks have been done in the past couple of years, including clustering [30], classification [7], and link prediction [8] etc. Among these tasks, similarity measure in HIN is an important and basic function. Several path-based similarity measures have been proposed. Sun et al. proposed PathSim [29] to measure relevance of same-typed objects based on symmetrical meta paths in HIN. Lao and Cohen proposed PCRW [10] to measure relevance of arbitrary entities in directed graph. Shi et al. proposed HeteSim [23], a symmetric measure, to evaluate similarity of arbitrary node pairs in heterogeneous network. Conventional HIN does not consider the link attribute values, while these link attribute information may be very important in some applications. In this paper, we propose the weighted HIN and explore its benefit for recommender system.

Although meta path has been widely used in HIN to capture semantics, some researchers have noticed the shortcomings of meta paths in failing to capture more subtle semantics in some applications. Li et al. [14] addressed the problem and proposed the constrained meta path concept, which can confine some constraints on objects. Moreover, Liu et al. [17] propose the concept "restricted meta path" which enables in-depth knowledge mining on the heterogeneous bibliographic networks by allowing restrictions on the node set. In order to capture refined semantics, these work consider the constraints on objects, while our work consider the constraints on links. Recently, Vahedian et al. [31] introduce a random walk method to generate meta-paths in heterogeneous network, which considers more on highly rated links for recommendation. Although their method considers the effect of ratings on links, our method insightfully investigate this effect and design corresponding similarity measure strategy.

Some researchers have begun to be aware of the importance of heterogeneous information for recommendations. Jamali and Lakshmanan [6] proposed HETEROMF to integrate a general latent factor and context-dependent latent factors. Wang et al. [3] proposed the OptRank method to alleviate the cold start problem by utilizing heterogeneous information contained in social tagging system. Lippert et al. [16] proposed a collective matrix factorization method which shares the latent factor of same object types in different relations. Yu et al. [35] proposed a matrix factorization method with entity similarity regularization. More recently, Luo et al. [18] proposed a collaborative filtering-based social recommendation method using heterogeneous relations, while Burke et al. [1] present an approach for recommendation which incorporates multiple relations in a weighted hybrid. Cao et al. [2] focus on link prediction with automatic meta path and designed an algorithm called automatic meta-path generation to automatically extract meta-paths from schema-rich HIN for link prediction. Zheng et al. [40] proposed a dual-similarity-regularization based recommendation method, which can simultaneously impose the constraint on users and items with high and low similarities. These methods usually focus on integrating heterogeneous information and only consider partial information available in HIN. The proposed SemRec considers more comprehensive information and focuses on exploiting path semantics.

The most similar work is HeteRec proposed by Yu et al. [36] which employs an implicit feedback recommendation model with systematically extracted latent features from heterogeneous network. However, SemRec is different from HeteRec in many aspects. In order to solve the top k recommendation problem, HeteRec adopts the Bayesian ranking optimization technique to realize the personalized recommendation, and utilizes the meta path to capture path semantics. In order to solve the rating score prediction problem, SemRec employs the novel weight regularization to realize personalized recommendation and avoid the rating sparsity. More importantly, SemRec firstly applies the proposed weighted meta path to delicately depict the path semantics.



(a) Douban Movie **(b) Yelp** **(c) Douban Book**

Fig. 2: Network schema of weighted heterogeneous information networks constituted by Douban Movie/Yelp/Douban Book datasets.

3 Heterogeneous network framework for recommendation

In this section, we describe notations used in this paper and present some preliminary knowledge.

3.1 Basic concepts

A HIN is a special type of information network with the underneath data structure as a directed graph, which either contains multiple types of objects or multiple types of links. Traditionally, HIN does not consider the attribute values on links. However, many real networks contain attribute values on links. For example, users usually rate movies with a score from 1 to 5 in movie recommender system, and the "author of" relations between authors and papers in bibliographic networks can take values (e.g., 1, 2, 3) which means the order of authors in the paper. In this paper, we formally propose the weighted heterogeneous information network concept to handle this condition.

Definition 1 ***Weighted Information Network***. *Given a schema $S = (\mathcal{A}, \mathcal{R}, \mathcal{W})$ which consists of a set of object types $\mathcal{A} = \{A\}$, a set of relations connecting object pairs $\mathcal{R} = \{R\}$, and a set of attribute values on relations $\mathcal{W} = \{W\}$, a* ***weighted information network*** *is defined as a directed graph $G = (V, E, W)$ with an object type mapping function $\varphi : V \to \mathcal{A}$, a link type mapping function $\psi : E \to \mathcal{R}$, and an attribute value type mapping function $\theta : W \to \mathcal{W}$. Each object $v \in V$ belongs to one particular object type $\varphi(v) \in \mathcal{A}$, each link $e \in E$ belongs to a particular relation $\psi(e) \in \mathcal{R}$, and each attribute value $w \in W$ belongs to a particular attribute value type $\theta(w) \in \mathcal{W}$. When the types* *of objects $|\mathcal{A}| = 1$ and the types of relations $|\mathcal{R}| = 1$, it is a* ***homogeneous information network****. When the types of objects $|\mathcal{A}| > 1$ (or the types of relations $|\mathcal{R}| > 1$) and the types of attribute values $|\mathcal{W}| = 0$, the network is called* ***unweighted heterogeneous information network****. When the types of objects $|\mathcal{A}| > 1$ (or the types of relations $|\mathcal{R}| > 1$) and the types of attribute values $|\mathcal{W}| > 0$, the network is called* ***weighted heterogeneous information network*** *(WHIN).*

Conventional HIN is an unweighted HIN, where there are no attribute values on relations or we do not consider them. For a WHIN, there are attribute values on some relation types, and these attribute values may be discrete or continuous values.

Example 1 *A movie recommender system can be organized as a weighted heterogeneous information network, whose network schema is shown in Fig. 2a. The network contains objects from six types of entities (e.g., users, movies, groups, actors) and relations between them. Links between objects represent different relations. For example, links exist between users and users denoting the friendship relations, between users and movies denoting rating and rated relations. In addition, the network also contains one type of attribute value on the rating relation between users and movies, which take values from 1 to 5.*

Table 1: The meanings and corresponding recommendation models of meta paths.

No.	Meta Path	Semantic Meaning	Recommendation Model
1	UU	friends of the target user	Social recommendation
2	UGU	users in the same group of the target user	Member recommendation
3	UMU	users who view the same movies with the target user	Collaborative recommendation
4	UMTMU	users who view the movies having the same types with that of the target user	Content recommendation

Table 2: Examples and their semantic meanings of weighted meta paths.

No.	Weighted Meta Path	Semantic Meaning
1	U(1)M(1)U	Users having the same rating score 1 on some movies as the target user
2	U(1,2)M(1,2)U	Users disliking the same movies as the target user
3	U(i)M(j)U$\|i=j$	Users having exactly the same rating on some movies as the target user
4	U(i)M(j)U$\|\,\|i-j\|\leq 1$	Users having rating within a difference of one on some movies as the target user
5	U(1)MTM(1)U	Users having the same rating score 1 on movies that have the same types with movies viewed by target user

Two objects in a HIN can be connected via different paths and these paths have different meanings. As an example shown in Fig. 2a, users can be connected via "User-User" (UU) path, "User-Group-User" (UGU) path, "User-Movie-User" (UMU) and so on. These paths are called meta paths that are the combination of a sequence of relations between object types. Although meta path is widely used to reveal semantics among objects [28], it fails to distinguish the attribute values between two objects in WHIN. For example, if ignoring the different rating scores of users on items in above movie recommendation, we may obtain incorrect results. Consider a scenario that we use the UMU path to find the similar users of Tom according to their viewing records in Fig. 1. We can infer that Tom is very similar to Mary and Bob, since they have the same viewing records. However, it is obvious that Tom and Mary have totally different tastes. So the UMU path cannot subtly reveal the different ratings of users on the same movies. In order to effectively exploit semantics in WHIN, we extend the conventional meta path to consider attribute values on relations. Without loss of generality, we assume the attribute values on relations in WHIN are discrete. For continuous attribute values on relations, we can convert the continuous attribute values into discrete ones.

Definition 2 ***Extended meta path on WHIN****. Extended meta path is a meta path based on a certain attribute value constraint on relations, which is denoted as $A_1 \xrightarrow{\delta_1(R_1)} A_2 \xrightarrow{\delta_2(R_2)} \cdots \xrightarrow{\delta_l(R_l)} A_{l+1}|\mathcal{C}$ (also denoted as $A_1(\delta_1(R_1))A_2(\delta_2(R_2))\cdots(\delta_l(R_l))$*
$A_{l+1}|\mathcal{C}$). If the relation R has attribute values on links, the attribute value function $\delta(R)$ is a set of values from the attribute value range of relation R, else $\delta(R)$ is an empty set. $A_i \xrightarrow{\delta_i(R_i)} A_{i+1}$

represents the relation R_i between A_i and A_{i+1} based on the attribute values $\delta_i(R_i)$. The constraint $\mathcal{C}$ on attribute value functions is a set of correlation constraints among attribute value functions. If all attribute value functions in a meta path are empty set (the corresponding constraint $\mathcal{C}$ is also an empty set), the path is called an ***unweighted meta path****, else the path is called a* ***weighted meta path****.*

Note that, the conventional meta path is an unweighted meta path that can be considered as the special case of a weighted meta path.

Example 2 *Taking Fig. 2a as an example, the rating relation between users U and movies M can take scores from 1 to 5. the weighted meta path $U \xrightarrow{1} M$ (i.e., $U(1)M$) means movies rated by users with score 1, which implies that users dislike the movies. The weighted meta path $U \xrightarrow{1,2} M \xrightarrow{1,2} U$ (i.e., $U(1,2)M(1,2)U$) means users disliking the same movies as the target user, while the unweighted meta path UMU can only reflect that users have the same viewing records. Furthermore, we can flexibly set the correlation constraints of attribute value functions on different relations in weighted meta paths. For example, the path $U(i)M(j)U|i=j$ means users having exactly the same ratings on some movies as the target user. Under this path, we can easily find that, in Fig. 1, Tom is very similar to Bob, while they are totally dissimilar to Mary. Table 2 shows more examples and corresponding semantics of weighted meta paths.*

3.2 Recommendation on heterogeneous networks

For a target user, recommender systems usually recommend items according to his similar users. In HIN, there are a number of meta paths connecting users, such as "User-User" and "User-Moive-User". Based on these paths, users have different similarities. Here we define the path based similarity as follows.

Definition 3 ***Path based similarity****. In HIN, the path based similarity of two objects is the similarity evaluation based on the given meta path connecting these two objects.*

After obtaining the path based similarity of users, we can recommend items according to the similar users of the target user. More importantly, the meta paths connecting users have different semantics, which can represent different recommendation models. As an example shown in Fig. 2a, "User-User" (UU) means friends of the target user. If we recommend movies according to the similarity of users generated by that path, it will recommend the movies viewed by friends of the target user. Indeed, it is the social recommendation. Another example is that "User-Movie-User" (UMU) means users who view the same movies with the target user. Following that path, it will recommend the movies viewed by users having the similar viewing records with the target user. it is collaborative recommendation in essential. Table 1 shows the other representative paths and the corresponding recommendation models. Based on the HIN framework, we can flexibly represent different recommendation models through properly setting meta paths.

3.3 Similarity measure based on weighted meta path

Similarity measure on meta paths have been well studied, and several path based similarity measures have been proposed on HIN, such as PathSim [29], PCRW [9], and HeteSim [23]. However, these similarity measures cannot be directly applied to weighted meta path, because they do not consider the attribute value constraint on relations. As we know, the essential of the path based similarity measure is to evaluate the proportion of the number of paths connecting two objects on all possible paths along the meta path [29], so the similarity measure based on weighted meta path should only consider those paths that satisfy the attribute value constraint. Moreover, the attribute value on relations may be a variable, even correlated. Taking the U(i)M(j)U$|i=j$ path as an example, the attribute values i and j are variables from 1 to 5, and they satisfy constraint $i=j$. The implement of most of existing path based similarity measures are based on matrix multiplication, measuring the number of path instants or probability of random walk. However, matrix multiplication can not represent attribute value constraint on multiple relations (e.g., $i=j$). So for this kind of paths, existing path based similarity measures cannot handle it.

In order to address the variable, even correlated, attribute value constraints in a weighted meta path, we extend the meta path concept and propose a general strategy to make existing path based similarity measure still usable, instead of proposing an ad hot similarity measure. Specifically, we can decompose the weighted meta path into a group of atomic meta paths with fixed attribute value constraint. For an atomic meta path, the existing path based similarity measures can be used directly.

Definition 4 ***Atomic meta path***. *If all attribute value functions $\delta(R)$ in a weighted meta path take a specific value, the path is called an* ***atomic meta path****. A weighted meta path is* ***a group of atomic meta paths*** *which contain all atomic meta paths that satisfy the constraint $\mathcal{C}$.*

Example 3 *Taking Fig. 2a as an example, $U(1)M(1)U$ and $U(1)M(2)U$ both are atomic meta paths. The weighted meta path $U(i)M(j)U|i = j$ is a group of five atomic meta paths (e.g., $U(1)M(1)U$ and $U(2)M(2)U$).*

3.4 Similarity measure strategy

Since a weighted meta path is a group of corresponding atomic meta paths, the similarity measure based on a weighted meta path can be considered as the sum of the similarity measure based on the corresponding atomic meta paths. So the similarity based on a weighted meta path can be evaluated based on the following two steps: (1) evaluate the similarity based on each atomic meta path with existing path based measures; (2) sum up similarities on all atomic meta paths in the weighted meta path. Note that, similarity measure needs to consider the effect of the normalized term existing in some path based similarity measures, such as PathSim [29] and HeteSim [23]. Then we will specifically illustrate how to adapt existing path based similarity measure for weighted meta path.

PathCount is a basic similarity measure, which counts the number of path instants connecting two nodes along a meta path. Since a weighted meta path can decompose into a group of atomic meta paths, the number of path instants connecting two nodes along weighted meta path is the sum of the number of path instants along each atomic meta path in it. That is,

$$S(x, y|\mathcal{P}_\mathcal{C}) = \sum_{\mathcal{P}_a \in \mathcal{P}_\mathcal{C}} |\{p_{x \rightsquigarrow y} : p_{x \rightsquigarrow y} \in \mathcal{P}_a\}|, \tag{1}$$

where $\mathcal{P}_\mathcal{C}$ is a weighted meta path with attribute value constraint $\mathcal{C}$, $\mathcal{P}_a \in \mathcal{P}_\mathcal{C}$ is an atomic meta path satisfies constraint $\mathcal{C}$, and $p_{x \rightsquigarrow y} \in \mathcal{P}_a$ means a path instant connecting x and y along atomic meta path $\mathcal{P}_a$.

PathSim [29] is a path-count based similarity measure, which is able to find peer objects in network. Essentially, PathSim is the normalized version of PathCount.

$$S(x, y|\mathcal{P}_\mathcal{C}) = \frac{2 \times \sum_{\mathcal{P}_a \in \mathcal{P}_\mathcal{C}} |\{p_{x \rightsquigarrow y} : p_{x \rightsquigarrow y} \in \mathcal{P}_a\}|}{\sum_{\mathcal{P}_a \in \mathcal{P}_C} |\{p_{x \rightsquigarrow x} : p_{x \rightsquigarrow x} \in \mathcal{P}_a\}| + \sum_{\mathcal{P}_a \in \mathcal{P}_C} |\{p_{y \rightsquigarrow y} : p_{y \rightsquigarrow y} \in \mathcal{P}_a\}|}. \tag{2}$$

PCRW [9] measures similarity based on path-constraint random walk, which utilizes the reaching probability of random walk along given meta path. According to the proposal similarity measure strategy, PCRW can be revised as follow

$$S(x, y|\mathcal{P}_\mathcal{C}) = \sum_{\mathcal{P}_a \in \mathcal{P}_\mathcal{C}} \sum_{t = x \rightsquigarrow y \in \mathcal{P}_a} RW(t), \tag{3}$$

where $t = (a_1, a_2, \ldots, a_{l+1})$ is a tour from a_1 to a_{l+1} along an atomic meta path, and $RW(t) = \prod_{i=1}^{l} \frac{1}{d(a_i)}$ is the random walk probability of t where $d(a_i)$ is the degree of node a_i to all qualified nodes in entity type A_{i+1}.

HeteSim [23] measures similarity based on a path-constraint pair-wise random walk with normalization. Note that there is a path decompose process in HeteSim. That is, a path will be decomposed into two equal-length paths (i.e., $\mathcal{P} = \mathcal{P}_L\mathcal{P}_R$). Assume that $t = (a_1, a_2, \ldots, a_m, \ldots, a_{l+1})$ is a tour along an atomic meta path, where a_m is a node in the middle type A_m. $t_L = (a_1, a_2, \ldots, a_m)$ is the tour from a_1 to a_m along the left part of $\mathcal{P}_a$, denoted as $\mathcal{P}_{aL}$, and $t_R = (a_{l+1}, \ldots, a_m)$ is the tour from a_{l+1} to a_m against the right part of $\mathcal{P}_a$, denoted as $\mathcal{P}_{aR}$. Reaching probability of random walk along t_L and t_R can be calculated as $LRW(t_L) = \prod_{i=1}^{\frac{l}{2}} \frac{1}{O(a_i)}$ and $RRW(t_R) = \prod_{i=l+1}^{\frac{l}{2}+1} \frac{1}{I(a_i)}$, where $O(a_i)$ and $I(a_i)$ means the out-degree and in-degree of a_i. For conveniency, assume $PM_{\mathcal{P}_{aL}}(a_1, :$

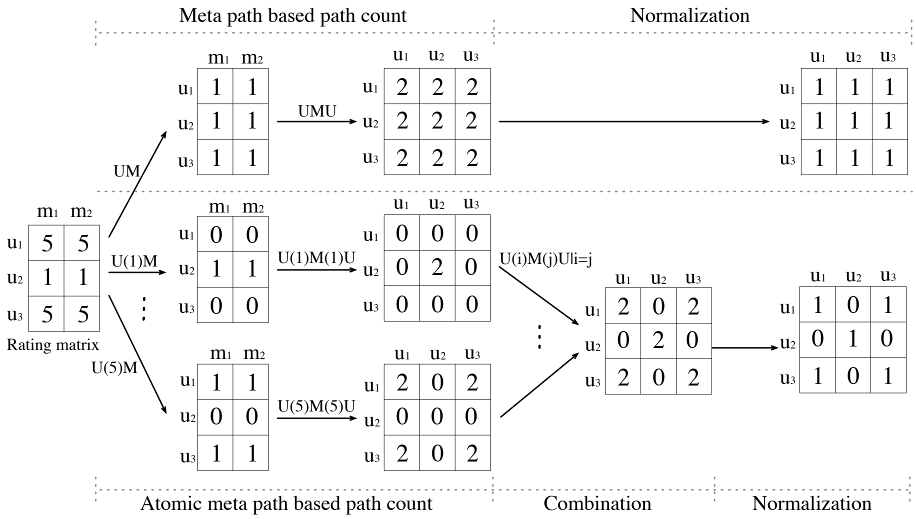


Fig. 3: PathSim similarity measure based on conventional and weighted meta path.

$) \in \mathbf{R}^{|A_m|}$ means the reaching probability vector from a_1 to all nodes of the middle type A_m, while $PM_{\mathcal{P}_{aR}}(a_{l+1}, :) \in \mathbf{R}^{|A_m|}$ means the reaching probability vector from a_{l+1} to all nodes of A_m. $PM_{\mathcal{P}_{aL}}(a_1, a_m) = \sum_{t_L \in \mathcal{P}_{aL}} LRW(t_L)$, while $PM_{\mathcal{P}_{aR}}(a_{l+1}, a_m) = \sum_{t_R \in \mathcal{P}_{aR}} RRW(t_R)$. And thus the HeteSim score between x and y along weighted meta path $\mathcal{P}_{\mathcal{C}}$ is

$$S(x, y|\mathcal{P}_c) = \sum_{\mathcal{P}_a = \mathcal{P}_{aL}\mathcal{P}_{aR} \in \mathcal{P}_{\mathcal{C}}} \frac{PM_{\mathcal{P}_{aL}}(x, :)^T PM_{\mathcal{P}_{aR}}(y, :)}{\sqrt{||PM_{\mathcal{P}_{aL}}(x, :)||||PM_{\mathcal{P}_{aR}}(y, :)||}} \tag{4}$$

Taking PathSim as an example, we illustrate its calculation process along conventional and weighted meta path in Fig. 3, where the rating matrix between 3 users and 2 movies are from Fig. 1. We know that PathSim counts the number of path instances connecting two objects along conventional meta path with a normalized term (shown in the upper half of Fig. 3), and thus it regards that the users all are the same. As shown in the lower half of Fig. 3, PathSim along weighted meta path firstly counts the number of path instances along each atomic meta path, and then sums up the the number of path instances along all atomic meta paths before normalization. And thus it can more accurately discover that only u_1 and u_3 are similar, since they have the same tastes in movies.

4 The SemRec Solution

4.1 Basic Idea

In this section, we proposed a **Sem**antic path based personalized **Rec**ommendation method (**SemRec**) to predict the scores of items. Specifically, SemRec first evaluates the similarity of users based on weighted or unweighted meta paths, and then infers the predicted scores on items according to the rating scores of similar users. Under different meta paths, the users can obtain different recommendation results. How to effectively combine these recommendations generated by different meta paths is challenging. We need to put different preferences on the various meta paths. This results in assigning preference weight to each meta path. We abbreviate the preference weight as weight when the context is clear without confusion with the link weight in the weighted meta path. There are two aspects of difficulties on learning the weights. (1) Prioritized weights. That is, the weights learned should embody the importance of paths and reflect users' preferences. However, the similarity evaluations based on different paths have significant bias, which makes path preference hard to reflect the path importances. For example, the similarity evaluations may all be high based on a path with dense relations, while the similarity evaluations may all be low based on another path with sparse relations. So the similarity evaluations based on different paths cannot reflect the similarity of two objects. SemRec designs a normalized rating intensity operation to eliminate the similarity bias, which makes the weight better reflect path importances. (2) Personalized weights. That is, it is better to learn weight preferences for each user. However, personalized weight learning may suffer from the rating sparsity problem, since many users have little rating informations. In order to alleviate the rating sparsity problem for personalized weight learning, we propose *the consistency rule of weight preferences of similar users*. That is, we assume that two similar users have consistent weight preferences on meta paths. While it is reasonable, it is seldom used before. Two users are similar based on a path, which implies the path has similar impacts on these two users. That is to say, these users have the consistent preferences on the path. Following this principle, we design a novel weight regularization term, which effectively alleviate rating sparsity in personalized weight learning.

In following sections, we firstly design the basic recommendation method based on a single path. And then we propose three levels of personalized recommendation methods based on multiple paths: unified weights for all users, personalized weights for each user, and personalized weights with weight regularization.

4.2 Recommendation with single path

Based on the path based similarity of users, we can find the similar users of a target user under a given path, and then the rating score of the target user on an item can be inferred according to the rating scores of his similar users on the item. Assume that the range of rating scores are from 1 to N (e.g., 5); $\mathcal{P}$ is a set of unweighted or weighted meta paths; $R \in \mathbf{R}^{|U|\times|I|}$ is the rating matrix, where $R_{u,i}$ denotes the rating score of user u on item i; and $S \in \mathbf{R}^{|U|\times|U|}$ is the path based similarity matrix of users, where $S^{(l)}_{u,v}$ is the similarity of users u and v under path $\mathcal{P}_l$. Here we define the *rating intensity* $Q \in \mathbf{R}^{|U|\times|I|\times N}$, where $Q^{(l)}_{u,i,r}$ represents the intensity of user u rating item i with score r given path $\mathcal{P}_l$. $Q^{(l)}_{u,i,r}$ is determined by two aspects: the number of similar users rating the item i with score r, and the similarity of users. So we calculate $Q^{(l)}_{u,i,r}$ as the sum of similarity of users rating i with r.

$$
\begin{aligned}
Q^{(l)}_{u,i,r} &= \textstyle\sum_v S^{(l)}_{u,v} \times E_{v,i,r} \\
E_{v,i,r} &= \begin{cases} 1 & R_{v,i} = r \\ 0 & \text{others,} \end{cases}
\end{aligned}
\tag{5}
$$

where $E_{v,i,r}$ indicates whether user v rates item i with score r.

Under a meta path $\mathcal{P}_l$, the rating of a user u on an item i range from 1 to N with different rating intensity $Q^{(l)}_{u,i,r}$. So the *predicted rating score*, denoted as $\hat{R}^{(l)}_{u,i}$, of user u on item i under the path $\mathcal{P}_l$ can be the average of rating scores weighted by corresponding normalized intensity.

$$
\hat{R}^{(l)}_{u,i} = \sum_{r=1}^{N} r \times \frac{Q^{(l)}_{u,i,r}}{\sum_{k=1}^{N} Q^{(l)}_{u,i,k}}, \tag{6}
$$

and $\hat{R}^{(l)} \in \mathbf{R}^{|U|\times|I|}$ means the predicted rating matrix under path $\mathcal{P}_l$.

According to Eq. (6), we can predict the rating score of a user on an item under a given path, and then recommend the item with the high score for a target user. Moreover, the Eq. (6) has an additional advantage that it eliminates the similarity bias existing in different meta paths. As we know, the similarity of users under different meta paths have different scales, which makes similarity evaluation and rating intensity incomparable among different paths. The normalized rating intensity in Eq. (6) is able to eliminate those scale differences.

4.3 Recommendation with multiple paths

Under different meta paths, there are different predicted rating scores. In order to calculate the compositive score, we propose three different weight learning methods corresponding to different levels of personalized weights of users.

4.3.1 Unified weight learning for all users

For all users, we assign each meta path with a unified weight, which means the user preference on the path. This weight vector is denoted as $\boldsymbol{w} \in \mathbf{R}^{1\times|\mathcal{P}|}$, and $\boldsymbol{w}^{(l)}$ means the weight on path $\mathcal{P}_l$. The final predicted rating score under all meta paths, denoted as $\hat{R}_{u,i}$, can be the weighted sum of predicted rating score under each meta path.

$$\hat{R}_{u,i} = \sum_{l=1}^{|\mathcal{P}|} \boldsymbol{w}^{(l)} \times \hat{R}^{(l)}_{u,i}. \tag{7}$$

Hopefully, the predicted rating matrix $\hat{R} \in \mathbf{R}^{|U|\times|I|}$ should be as close as to the real rating matrix R. So a direct optimization objective can be defined as the square error between the real scores and the predicted scores.

$$\begin{aligned} \min_{\boldsymbol{w}} \mathcal{L}_1(\boldsymbol{w}) = \quad & \tfrac{1}{2}||Y \odot (R - \sum_{l=1}^{|\mathcal{P}|} \boldsymbol{w}^{(l)} \hat{R}^{(l)})||_2^2 + \tfrac{\lambda_0}{2}||\boldsymbol{w}||_2^2 \\ s.t. \quad & \boldsymbol{w} \geq 0, \end{aligned} \tag{8}$$

where the notation $\odot$ is the Hadamard product (also know as the entrywise product) between matrices, and $||\cdot||_p$ is the matrix L_p-norm. Y is an indicator matrix with $Y_{u,i} = 1$ if user u rated item i, and otherwise $Y_{u,i} = 0$. λ_0 is the regularization parameter.

4.3.2 Personalized weight learning for individual user

The above optimization objective has a basic assumption: all users have the same path preferences. However, in many real applications, each user has his personal interest preferences. Unified weights cannot provide personalized recommendations for users. To realize personalized recommendation, each user is assigned with weight vector on meta paths. The weight matrix is denoted as $W \in \mathbf{R}^{|U|\times|\mathcal{P}|}$, in which each entry, denoted as $W^{(l)}_u$, means the preference weight of user u on path $\mathcal{P}_l$. The column vector $W^{(l)} \in \mathbf{R}^{|U|\times 1}$ means the weight vector of all users on path $\mathcal{P}_l$. So the predicted rating $\hat{R}_{u,i}$ of user u rating item i under all paths is as follow.

$$\hat{R}_{u,i} = \sum_{l=1}^{|\mathcal{P}|} W^{(l)}_u \times \hat{R}^{(l)}_{u,i}. \tag{9}$$

Similarly, we can define the optimization objective as follows.

$$\begin{aligned} \min_{W} \mathcal{L}_2(W) = \quad & \tfrac{1}{2}||Y \odot (R - \sum_{l=1}^{|\mathcal{P}|} diag(W^{(l)}) \hat{R}^{(l)})||_2^2 + \tfrac{\lambda_0}{2}||W||_2^2 \\ s.t. \quad & W \geq 0, \end{aligned} \tag{10}$$

where $diag(W^{(l)})$ means the diagonal matrix transformed from a vector $W^{(l)}$.

4.3.3 Personalized weight learning with weight regularization

Although Eq. (10) considers user's personalized weights, it may be hard to effectively learn weights for those users that have little rating information. There are $|U| \times |\mathcal{P}|$ weight parameters to learn, while the training samples are usually much smaller than $|U| \times |I|$. The training samples are usually not sufficient for the weight learning, specially for those cold-start users and items. According to the consistency rule of weight preferences of similar users mentioned above, the path weights of a user should be consistent to that of his similar users. For users with little rating information, their path weights can be learnt from the weights of their similar users, since the similarity information of users are more available through meta paths. So we design a weight regularization term as follows, which compels the weights of a user consistent to the average of weights of his similar users.

$$\sum_{u=1}^{|U|} \sum_{l=1}^{|\mathcal{P}|} (W^{(l)}_u - \sum_{v=1}^{|U|} \bar{S}^{(l)}_{u,v} W^{(l)}_v)^2, \tag{11}$$

ALGORITHM 1: Framework of SemRec

Input:
G: weighted heterogeneous information network
$\mathcal{P}$: meta paths connecting users
λ_0 and λ_1: controlling parameter
α: step size for updating parameters
ϵ: convergence tolerance
Output:
W: the weight matrix of all users on all paths.
for $\mathcal{P}_l \in \mathcal{P}$ **do**
    Evaluate user similarity $S^{(l)}$
    Calculate rating intensity $Q^{(l)}$ with Eq. (5)
    Calculate predicted rating score $\hat{R}^{(l)}$ with Eq. (6)
end
Initialize $W > 0$ **repeat**
    $W_{old} := W$
    Calculate $\frac{\partial \mathcal{L}_3(W)}{\partial W}$ with Eq. (14)
    $W := max(0, W - \alpha \frac{\partial \mathcal{L}_3(W)}{\partial W})$
until $|W - W_{old}| < \epsilon$;

where $\bar{S}^{(l)}_{u,v} = \frac{S^{(l)}_{u,v}}{\sum_v S^{(l)}_{u,v}}$ is the normalized user similarity based on path $\mathcal{P}_l$. For convenience, the weight regularization term can be written as the following matrix format.

$$\sum_{l=1}^{|\mathcal{P}|} ||W^{(l)} - \bar{S}^{(l)} W^{(l)}||_2^2. \tag{12}$$

And thus the optimization objective is defined as follows.

$$\begin{aligned} \min_W \mathcal{L}_3(W) = \quad & \tfrac{1}{2}||Y \odot (R - \sum_{l=1}^{|\mathcal{P}|} diag(W^{(l)})\hat{R}^{(l)})||_2^2 \\ + \quad & \tfrac{\lambda_1}{2} \sum_{l=1}^{|\mathcal{P}|} ||W^{(l)} - \bar{S}^{(l)} W^{(l)}||_2^2 + \tfrac{\lambda_0}{2}||W||_2^2 \\ s.t. \qquad & W \geq 0. \end{aligned} \tag{13}$$

The above optimization objective is a non-negative quadratic programming problem, a simple special case of non-negative matrix factorization. Projected gradient method for non-negative bound-constrained optimization [15] can be applied to solve this problem. The gradient of Eq. (13) with respect to $W_u^{(l)}$ can be calculated as follows,

$$\begin{aligned} \frac{\partial \mathcal{L}_3(W)}{\partial W_u^{(l)}} = \quad & -(Y_u \odot (R_u - \sum_{l=1}^{|\mathcal{P}|} W_u^{(l)} \hat{R}_u^{(l)}))\hat{R}_u^{(l)T} + \lambda_0 W_u^{(l)} \\ + \quad & \lambda_1 (W_u^{(l)} - \bar{S}_u^{(l)} W^{(l)}) - \lambda_1 \bar{S}_u^{(l)T} (W^{(l)} - \bar{S}^{(l)} W^{(l)}). \end{aligned} \tag{14}$$

$W_u^{(l)}$ can be updated as follows,

$$W_u^{(l)} = max(0, W_u^{(l)} - \alpha \frac{\partial \mathcal{L}_3(W)}{\partial W_u^{(l)}}), \tag{15}$$

where α is the step size and can be set according to [15]. Algorithm 1 shows the framework of this version of SemRec.

4.4 Discussion

From the optimization objectives, we find that the unified weight learning method ($\mathcal{L}_1$ in Eq. (8)) is a special case of personalized weight learning ($\mathcal{L}_2$ in Eq. (10)), when the weights of all users on path $\mathcal{P}_l$ (i.e., $W^{(l)}$) have the same value. Furthermore, they both are the special cases of personalized weight learning with weight regularization. The objective $\mathcal{L}_3$ converts to $\mathcal{L}_2$ when λ_1 is 0, and $\mathcal{L}_3$ converts to

$\mathcal{L}_1$ when λ_1 converges to $+\infty$. So the λ_1 parameter controls the personalized level in fact. The smaller λ_1 means the more personalized weights for users, while it may lead to more difficult learning task. So a proper λ_1 is needed to be set in real applications. Algorithm 1 is a flexible algorithm framework. Through setting different meta paths (weighted or unweighted), SemRec can flexibly realize different recommendation models and generate different recommendations complying with path semantics.

The time complexity of SemRec is analyzed as follows. As shown in Algorithm 1, SemRec includes two main parts: (1) Calculation of basic information (Lines 1-5). The main time-consuming component lies in similarity evaluation, while it can be done offline and many strategies [23] and parallel computing can speed it up. (2) Weight learning (Lines 6-11). It is an quadratic programming problem with complexity $O((|R| + |U|^2) \times |\mathcal{P}|)$.

5 Experiments

In this section, extensive experiments on three real datasets illustrate the traits of SemRec from five aspects. We first validate the effectiveness of SemRec, especially for cold-start problem. Then we thoroughly explore the meanings of weights learned and validate benefits of the proposed weighted meta path. Finally, we illustrate the effect of λ_1 on performances.

Table 3: Statistics of Datasets

Dataset	Relations (A-B)	Number of A	Number of B	Number of (A-B)	Ave. Degrees of A/B
Douban Movie	User-Movie	13367	12677	1068278	79.9/84.3
	User-User	2440	2294	4085	1.7/1.8
	User-Group	13337	2753	570047	42.7/207.1
	Movie-Director	10179	2449	11276	1.1/4.6
	Movie-Actor	11718	6311	33587	2.9/5.3
	Movie-Type	12676	38	27668	2.2/728.1
Yelp	User-Business	16239	14284	198397	12.2/13.9
	User-User	10580	10580	158590	15.0/15.0
	User-Compliment	14411	11	76875	5.3/6988.6
	Business-City	14267	47	14267	1.0/303.6
	Business-Category	14180	511	40009	2.8/78.3
Douban Book	User-Book	13024	22347	792026	60.8/35.4
	User-User	12748	12748	169150	13.3/13.3
	User-Group	13024	2936	1189271	91.3/405.1
	Book-Author	21907	10805	21907	1.0/78.3
	Book-Publisher	21773	1815	21773	1.0/11.9
	Book-Year	21192	64	21192	1.0/331.1

5.1 Datasets

In order to get more comprehensive heterogeneous information, we crawled a user-movie dataset from Douban[1], a well known social media network in China. The dataset includes 13367 users and 12677 movies with 1068278 movie ratings ranging from 1 to 5. The dataset includes the social relation among users and the attribute information of users and movies. The second dataset is the Yelp challenge dataset[2]. This dataset contains user ratings on local business and attribute information of users and businesses. We ignore users and businesses which has no related rating and finally get 16239 users and 14284 local businesses with 198397 ratings ranging from 1 to 5. The last dataset is Douban Book dataset[3]. This dataset includes 13024 users and 22347 books with 792026 ratings

[1] http://movie.douban.com/
[2] http://www.yelp.com/dataset_challenge/
[3] http://book.douban.com/

ranging from 1 to 5, with containing social relation among users and attribute information of users and books. The detailed description of these three datasets can be seen in Table 3, and their network schemas are shown in Fig. 2. We can find that these three datasets have different properties. Douban Movie dataset has dense rating relations but sparse social relations, Yelp dataset has sparse rating relations but dense social relations, while Douban Book dataset has relatively medium dense rating information with dense social relations.

5.2 Metrics

We use two widely used metrics, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), to measure the rating prediction quantity.

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_{test}} (R_{u,i} - \hat{R}_{u,i})^2}{|R_{test}|}}, \tag{16}$$

$$MAE = \frac{\sum_{(u,i)\in R_{test}} |R_{u,i} - \hat{R}_{u,i}|}{|R_{test}|}. \tag{17}$$

where $R_{u,i}$ denotes the real rating user u gave to item i and $\hat{R}_{u,i}$ denotes the predicted rating. R_{test} denotes whole test set. A smaller MAE or RMSE means a better performance.

5.3 Comparison Methods

In order to show the effectiveness of the proposed SemRec, we compare four variations of SemRec with the state of the arts. Besides the personalized weight learning method with weight regularization (called $\mathrm{SemRec_{Reg}}$), we include three special cases of SemRec: single path based method (called $\mathrm{SemRec_{Sgl}}$), unified weight learning method for all users (called $\mathrm{SemRec_{All}}$), and personalized weight learning method for individual user (called $\mathrm{SemRec_{Ind}}$). As the baselines, four representative rating predication methods are illustrated as follows. Note that the top k recommendation methods [6, 36] are not included here, since the problem they solve is different from the rating prediction in this paper.

- PMF [22]: It is the basic matrix factorization method using only user-item matrix for recommendations.
- SMF [21]: It adds the social regularization term into PMF, which aims at getting the users' latent factor closer to their friends' latent factors.
- CMF [16]: A collective matrix factorization method, which factorizes all relations in HIN and shares the latent factor of same object types in different relations.
- HeteMF [35]: A matrix factorization method with entity similarity regularization, which also utilizes the relations in HIN.

We employ 5 meaningful meta paths whose lengths are not longer than 4 for three datasets, since the longer meta paths are not meaningful and they fail to produce good similarity measures [29]. How to select meta paths for real applications is an open problem in HIN [23]. Recently, some works [25,39] show that the meaningful meta paths are potential to achieve better performances. Table 4 shows those paths which include the weighted and unweighted meta paths. Since our method is based on collaborative filtering which is to detect similar users, these meta paths are formatted as User-*-User to measure the similarity of users based on different meta paths. For SemRec, we use PathSim [29] as the similarity measure to calculate the similarity between users. Note that we employ the adapted PathSim (see Section 3.4) as similarity measure for weighted meta path. The parameter λ_0 in SemRec is 0.01 and λ_1 is 10^3 for the best performance. The parameters in other methods are set with the best performances on these datasets.

Table 4: Meta Paths used in Experiments for Three Datasets

Dataset	Meta paths
Douban Movie	User-Group-User, User-(i)-Movie-(j)-User$\|i=j$, User-(i)-Movie-Director-Movie-(j)-User$\|i=j$ User-(i)-Movie-Actor-Movie-(j)-User$\|i=j$, User-(i)-Movie-Type-Movie-(j)-User$\|i=j$
Yelp	User-User, User-Compliment-User, User-(i)-Business-(j)-User$\|i=j$ User-(i)-Business-Category-Business-(j)-User$\|i=j$, User-(i)-Business-City-Business-(j)-User$\|i=j$
Douban Book	User-Group-User, User-(i)-Book-(j)-User$\|i=j$, User-(i)-Book-Author-Book-(j)-User$\|i=j$ User-(i)-Book-Publisher-Book-(j)-User$\|i=j$, User-(i)-Book-Year-Book-(j)-User$\|i=j$

Table 5: Effectiveness Experiments on RMSE Performances for Three Datasets

Dataset	Training Settings	PMF	SMF	CMF	HeteMF	SemRec$_{\mathbf{Sgl}}$	SemRec$_{\mathbf{All}}$	SemRec$_{\mathbf{Ind}}$	SemRec$_{\mathbf{Reg}}$
Douban movie		0.9750	0.9743	0.9285	0.8513	0.8434	0.8125	0.8753	**0.7844**
	20%		0.07%	4.77%	12.69%	13.50%	16.67%	10.23%	19.55%
		0.8455	0.8449	0.8273	0.7796	0.8138	0.7814	0.8083	**0.7452**
	40%		0.07%	2.15%	7.79%	3.75%	7.58%	4.40%	11.86%
		0.7975	0.7967	0.8042	0.7601	0.7937	0.7709	0.7729	**0.7296**
	60%		0.10%	-0.84%	4.69%	0.48%	3.34%	3.08%	8.51%
		0.7673	0.7674	0.7741	0.7550	0.7846	0.7656	0.7540	**0.7216**
	80%		-0.01%	-0.89%	1.60%	-2.25%	0.22%	1.73%	5.96%
Yelp		1.6779	1.4843	1.6161	1.2333	1.3252	1.2166	1.3654	**1.2025**
	60%		11.54%	3.68%	26.50%	21.02%	27.49%	18.62%	28.33%
		1.5931	1.4017	1.5731	1.2090	1.2889	1.1906	1.3229	**1.1760**
	70%		12.01%	1.26%	24.11%	19.09%	25.27%	16.96%	26.18%
		1.5323	1.3678	1.5194	1.1895	1.2576	1.1665	1.2922	**1.1559**
	80%		10.74%	0.84%	22.37%	17.93%	23.87%	15.67%	24.56%
		1.4833	1.3377	1.4793	1.1755	1.2331	1.1496	1.2658	**1.1423**
	90%		9.82%	0.27%	20.75%	16.87%	22.50%	14.66%	22.99%
Douban book		1.1781	0.9431	0.9741	0.9256	0.9205	0.9201	0.9861	**0.9132**
	20%		19.95%	17.32%	21.43%	21.87%	21.90%	16.30%	22.49%
		0.8583	0.7903	0.8080	0.7852	0.7849	0.7759	0.8148	**0.7718**
	40%		7.92%	5.86%	8.52%	8.55%	9.60%	5.07%	10.08%
		0.7827	0.7520	0.7573	0.7480	0.7588	0.7449	0.7543	**0.7391**
	60%		3.92%	3.25%	4.43%	3.05%	4.83%	3.63%	5.57%
		0.7454	0.7294	0.7490	0.7309	0.7516	0.7356	0.7296	**0.7190**
	80%		2.15%	-0.48%	1.95%	-0.83%	1.31%	2.12%	3.54%

Table 6: Effectiveness Experiments on MAE Performances for Three Datasets

Dataset	Training Settings	PMF	SMF	CMF	HeteMF	SemRec$_{\mathbf{Sgl}}$	SemRec$_{\mathbf{All}}$	SemRec$_{\mathbf{Ind}}$	SemRec$_{\mathbf{Reg}}$
Douban movie	**20%**	0.7198	0.7192	0.6971	0.6342	0.6506	0.6309	0.6412	**0.6054**
			0.08%	3.15%	11.89%	9.61%	12.35%	10.92%	15.89%
		0.6319	0.6313	0.6263	0.5927	0.6351	0.6149	0.6032	**0.5808**
	40%		0.09%	0.89%	6.20%	-0.51%	2.69%	4.54%	8.09%
		0.6010	0.6002	0.6090	0.5800	0.6172	0.6098	0.5840	**0.5698**
	60%		0.13%	-1.33%	3.49%	-2.70%	-1.46%	2.83%	5.19%
		0.5812	0.5815	0.5900	0.5758	0.6142	0.6072	0.5739	**0.5639**
	80%		-0.05%	-1.51%	0.93%	-5.68%	-4.47%	1.26%	2.98%
Yelp		1.2997	1.0830	1.2628	0.9268	0.9657	0.9040	1.0029	**0.8901**
	60%		16.67%	2.84%	28.69%	25.70%	30.45%	22.84%	31.51%
		1.2262	1.0547	1.2224	0.9107	0.9420	0.8873	0.9728	**0.8696**
	70%		13.99%	0.31%	25.73%	23.18%	27.64%	20.67%	29.08%
		1.1740	1.0282	1.1740	0.8969	0.9224	0.8723	0.9517	**0.8548**
	80%		12.42%	0.00%	23.60%	21.43%	25.70%	18.94%	27.19%
		1.1324	1.0085	1.1405	0.8878	0.9067	0.8616	0.9322	**0.8442**
	90%		10.94%	-0.72%	21.60%	19.93%	23.91%	17.68%	25.45%
Douban book		0.8147	0.6732	0.768	0.6859	0.6855	0.68	0.7087	**0.6791**
	20%		17.30%	5.70%	15.81%	15.86%	16.53%	13.01%	16.64%
		0.6363	0.6141	0.6477	0.6078	0.622	0.6102	0.6162	**0.6001**
	40%		3.49%	-1.79%	4.48%	2.25%	4.10%	3.16%	5.69%
		0.6008	0.5902	0.6089	0.5822	0.6095	0.5948	0.5822	**0.5768**
	60%		1.76%	-1.35%	3.10%	-1.45%	1.00%	3.10%	3.99%
		0.5754	0.5748	0.6023	0.5709	0.5973	0.5892	0.5673	**0.5638**
	80%		0.10%	-4.68%	0.78%	-3.81%	-2.40%	1.41%	2.02%

Table 7: Running Time for Three Datasets with 60% Training Setting (second)

Dataset	PMF	SMF	CMF	HeteMF	SemRec$_{\mathbf{Sgl}}$	SemRec$_{\mathbf{All}}$	SemRec$_{\mathbf{Ind}}$	SemRec$_{\mathbf{Reg}}$
Douban Movie	260.25	266.78	509.31	736.85	0	1.44	155.98	293.14
Yelp	31.8	51.19	375.38	619.25	0	0.25	57.22	374.57
Douban Book	201.10	233.21	610.32	755.10	0	0.61	103.21	264.02

5.4 Effectiveness Experiments

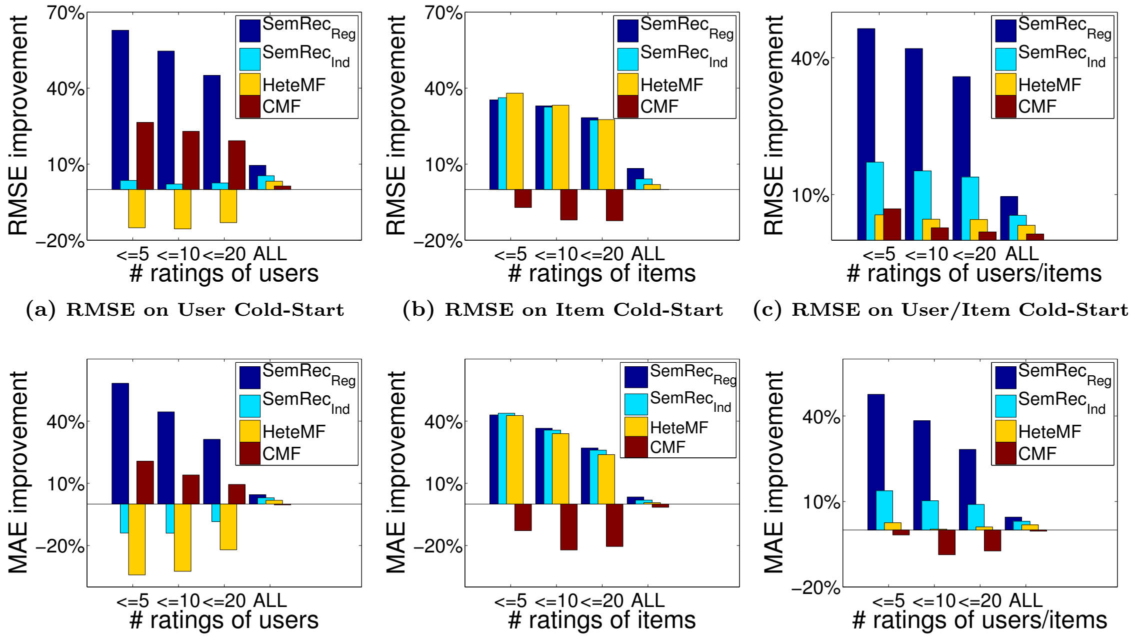
For Douban Movie dataset and Douban Book dataset, we use different training data settings (20%, 40%, 60%, 80%) to show the comparison results in different data sparseness. Training data 20%, for example, means that 20% of the ratings from user-item rating matrix is randomly selected as the training data to predict the remaining 80%. From Table 3, we can find that the two Douban datasets have dense rating relations, while Yelp has very sparse rating relations. So we utilize more training data (60%, 70%, 80%, 90%) on Yelp.

The random selection was repeated 10 times independently and the average results are reported in Table 5 and Table 6. Note that SemRec$_{\mathrm{Sgl}}$ reports the best performances on these five paths.

The main phenomena and conclusions are listed as follows:

- From the results, we can observe that all versions of SemRec outperform other approaches in most conditions. Particularly, SemRec$_{\mathrm{Reg}}$ always achieves the best performances on all conditions. For example, on 20% training set of Douban Movie, SemRec$_{\mathrm{Reg}}$ outperforms PMF up to 19.55% on RSME and 15.89% on MAE, and on 20% training set of Douban Book, SemRec$_{\mathrm{Reg}}$ outperforms PMF up to 22.49% on RSME and 16.64% on MAE. On Yelp dataset, SemRec$_{\mathrm{Reg}}$ outperforms PMF over 20% on all chosen conditions. As compared to PMF, CMF improves the recommendation performances through integrating heterogeneous information with matrix factorization. However, its performances are much worse than the proposed SemRec on all conditions, especially on less training set. As the most similar method to SemRec, HeteMF also has good performances, while its performances are still worse than the proposed SemRec$_{\mathrm{Reg}}$. These all imply that the proposed SemRec has better mechanism to integrate heterogeneous information.
- Different versions of SemRec have different performances. Generally, SemRec with multiple paths (e.g., SemRec$_{\mathrm{All}}$, and SemRec$_{\mathrm{Reg}}$) have better performances than SemRec with single path (i.e., SemRec$_{\mathrm{Sgl}}$) except SemRec$_{\mathrm{Ind}}$, which indicates that the weight learning of SemRec can effectively integrate the similarity information generated by different paths. Because of rating sparsity, SemRec$_{\mathrm{Ind}}$ has worse performances than SemRec$_{\mathrm{All}}$ on most conditions. In addition, the better performances of SemRec$_{\mathrm{Reg}}$ over SemRec$_{\mathrm{Ind}}$ confirm the benefit of the weight regularization term. In all, SemRec$_{\mathrm{Reg}}$ always achieves best performances in all conditions. The reason lies in that SemRec$_{\mathrm{Reg}}$ not only realizes personalized weight learning for all users but also avoids the rating sparsity through the weight regularization in it.
- The results also show that SemRec has more obvious superiority with less training set. We think the reason is that meta paths consisting of different types of objects and relations contain more informations and multiple paths reveal similarity of users from different aspects. These additional informations are very useful for those users who have a small quantity of rating records. When comparing the performances of SMF to that of PMF, we notice that the performance improvement of SMF is very marginally on Douban dataset, while its improvement is more obvious on Yelp dataset. The different behaviors of SMF stem from the different density of social relations on Douban and Yelp dataset. The experiments show that the social recommendations (i.e., SMF) is more suitable for dataset with dense social relations. However, the density of social relations have less effect on the proposed SemRec, since SemRec can integrate more information from attributes through path based similarity.

Furthermore, we record the average running time of these methods on the learning process in Table 7. For two similarity based methods (e.g., SemRec and HeteMF), we do not consider the running time on similarity evaluation, since it can be done offline beforehand. For the four versions of SemRec, their running times increase when the weight learning tasks become more complex. Both SemRec$_{\mathrm{Sgl}}$ and SemRec$_{\mathrm{All}}$ are very fast, which can be applied for online learning. The running times of SemRec$_{\mathrm{Ind}}$ and SemRec$_{\mathrm{Reg}}$ are still acceptable when comparing to CMF and HeteMF. We can select a proper model through balancing the efficiency and effectiveness of SemRec in real applications.



(a) RMSE on User Cold-Start (b) RMSE on Item Cold-Start (c) RMSE on User/Item Cold-Start

(d) MAE on User Cold-Start (e) MAE on Item Cold-Start (f) MAE on User/Item Cold-Start

Fig. 4: Performance improvements of four HIN based methods against PMF on different levels and types of cold-start problems.

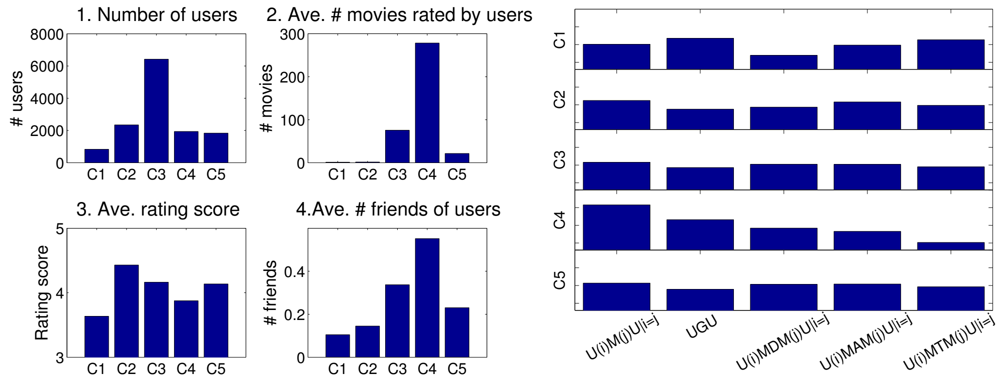
5.5 Study on Cold-Start Problem

The better performances of SemRec with less training set imply that SemRec has the potential to alleviate the cold-start problem. In this section, we will exploit the ability of SemRec on alleviating the cold-start problem through observing its performances on different levels of cold-start users and items. We run PMF, CMF, HeteMF, SemRec$_{\mathrm{Ind}}$, and SemRec$_{\mathrm{Reg}}$ on Douban Movie dataset with users having the different numbers of rated movies. We select four types of users: three types of cold-start users with different numbers of rated movies (e.g., users with the number of rated movies no more than 5, denoted as ≤ 5 in Fig. 4) and all users (called ALL in Fig. 4). In addition, we also do the similar experiments on cold-start items and users&items (contain both cold-start users and items). We record the performance improvement of other four algorithms against PMF on two metrics RMSE and MAE in Fig. 4.

It is clear that SemRec$_{\mathrm{Reg}}$ always achieves the best performance improvements on almost all conditions and both metrics, and its superiority is more significant for less rating information. On the contrary, CMF only achieves improvements on cold-start users and HeteMF's improvements are only on items. We think the reason lies in that the collective matrix factorization of all relations in CMF may introduce much noises, especially for items. HeteMF only utilizes the similarity information of items, ignoring that of users. Generally, integrating heterogeneous information is helpful for alleviating cold-start problem (see Fig. 4c and Fig. 4f), while the integrating mechanisms may have different impact on cold-start items and users. The overall performance improvements of SemRec$_{\mathrm{Reg}}$ are attributed to multiple meta paths that not only contain rich attribute information but also provide comprehensive and complementary similarity evaluation of users and items. In addition, the better performances of SemRec$_{\mathrm{Reg}}$ over SemRec$_{\mathrm{Ind}}$ further validate that the weight regularization term employed in SemRec$_{\mathrm{Reg}}$ is really helpful for the weight learning of cold-start users from similar users.

5.6 Study of Weight Preferences

In this section, we illustrate the meanings of weights learned by SemRec through a case study. Based on the results of SemRec$_{\mathrm{Reg}}$ on Douban Movie dataset with 60% data for training in the above experiments, we cluster users' weight vectors into 5 groups using K-means, and then show the



(a) Statistics of clusters **(b) Weights on meta paths**

Fig. 5: Analysis of clusters' characteristics and path preferences of results returned by SemRec$_{\mathbf{Reg}}$ on Douban dataset. C1-C5 represents the index of five clusters.

statistics information of users in five clusters in Fig. 5a. Moreover, the weight preferences of the five cluster centers on 5 meta paths are also shown in Fig. 5b.

Let's observe the relationship of the statistics information of users in different clusters and their weight preferences on paths from Fig. 5a and 5b. As we know, Douban is a unique social media platform in China, in which the major active users are young people who love culture and arts. As the typical and major users in Douban, the users in C3 view a good number of movies, give relatively good rating scores, and have a moderate number of friends. So they also have close weight preferences on all paths. As the top movie fans, the users in C4 who view a great many movies tend to give lower rating scores due to critical attitude, and most of them have many friends. And they obviously like to get recommendation from viewing records of other users (i.e., UMU) and interest group (i.e., UGU), but less paying attentions to movies' content (e.g., UMTMU and UMAMU). In addition, the users in C1 and C2 are two types of inactive users, and they view few movies and have few friends. Because of not being fond of movies, these users tend to give much high or low rating scores. These users comparatively prefer to follow movie content (e.g., UMTMU and UMAMU). The picky users in C1 is more likely to get recommendation from interest group (i.e., UGU), while the idealess users in C2 give more preferences to viewing records of other users (i.e., UMU).

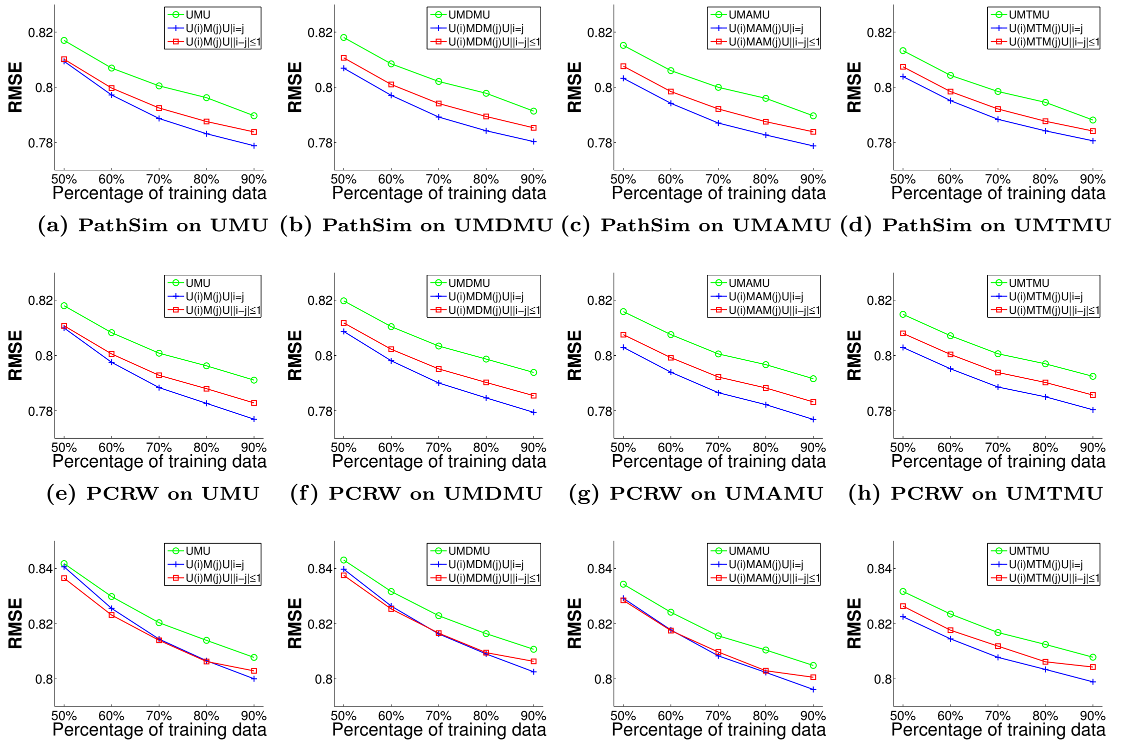
In all, the weights of paths learned by SemRec can reflect the users' path preferences, and these path preferences are able to reveal the users' characteristics to a large extent. More importantly, the meaningful weight preferences are very useful for recommendation explanation. We know that the meta path has semantics, so we can tell users the recommendation reason according to the path semantics of the high weight path. Although some weight learning methods on paths have been proposed [18, 36], their weights fail to reflect users' preferences on paths. We think two strategies adopted in RecSem contribute to its good properties. (1) We design the predicted rating score in Eq. (2), which can eliminate the similarity bias on different meta paths by the adoption of normalized rating intensity. (2) We employ the weight regularization term in Eq. (9) according to the consistency rule of weight preferences of similar users. The consistency rule makes similar users have similar weight preferences. In other words, weights also reveal users' similarity and preferences.

5.7 Study on Weighted Meta Path

In this section, experiments performed on weighted meta path show its effectiveness on improving the performances of SemRec, as well as the accurate semantic meaning of similarity measure.

5.7.1 Effectiveness Experiments on Recommendation

In this section, we study the effectiveness of weighted meta path on improving the performances of SemRec, through more accurately revealing relations among objects. We observe the performances of SemRec under different meta paths and similarity measures.



(a) PathSim on UMU (b) PathSim on UMDMU (c) PathSim on UMAMU (d) PathSim on UMTMU

(e) PCRW on UMU (f) PCRW on UMDMU (g) PCRW on UMAMU (h) PCRW on UMTMU

(i) HeteSim on UMU (j) HeteSim on UMDMU (k) HeteSim on UMAMU (l) HeteSim on UMTMU

Fig. 6: The performances of SemRec with different weighted meta paths and different similarity measure.

Four meta paths are employed to demonstrate that the improvement can be achieved along weighted meta paths. For the meta path UMU, we design two weighted paths U(i)M(j)U$|i = j$ and U(i)M(j)U$||i - j| \leq 1$. U(i)M(j)U$|i = j$ means users rating the exact same scores on the same movies, while U(i)M(j)U$||i - j| \leq 1$ means users rating close scores. Similarly, we design two corresponding weighted paths for UMDMU, UMAMU, and UMTMU. Based on the similarity generated by these meta paths, we employ SemRec$_{\mathrm{Sgl}}$ to make recommendations. We compare the performances of SemRec$_{\mathrm{Sgl}}$ with different paths and record the results in Fig. 6a to Fig. 6d.

Moreover, three adapted similarity measures (i.e., PathSim [29], PCRW [9], and HeteSim [23]) are employed to show that the improvement can also be achieved with different similarity measures. Again, SemRec$_{\mathrm{Sgl}}$ is utilized as recommendation approach. The performances of SemRec$_{\mathrm{Sgl}}$ with different similarity measures along four meta paths are shown in Fig. 6e to Fig. 6l.

The experimental results clearly show that SemRec with weighted meta paths (e.g., U(i)M(j)U$|i = j$ and U(i)M(j)U$||i - j| \leq 1$) significantly outperform SemRec with unweighted meta paths (e.g., UMU) along different paths or with different similarity measures. Let's take the UMU path as an example to analyze the reasons. Failing to distinguish the different rating scores of users on the same movies, UMU cannot accurately reveal user similarity, so it has bad performances. The path U(i)M(j)U$|i = j$ and U(i)M(j)U$||i - j| \leq 1$ not only consider the differences of rating scores but also keep dense relations, so they can achieve better performances than UMU. Moreover, the results show that the proposed similarity measures on weighted meta path (i.e., PathSim, PCRW, and HeteSim) all achieve better performances through measuring more accurate similarity along weighted meta path. This experiments furtherly confirm that the weighted meta paths are really helpful to improve recommendation performances by more accurately revealing object relations.

5.7.2 Case Study on Semantic Meaning of Meta paths

Weighted heterogeneous information network is ubiquitous and has various real world applications. Bibliographic network extracted from DBLP is a widely used HIN, shown in Fig 7, constituted by

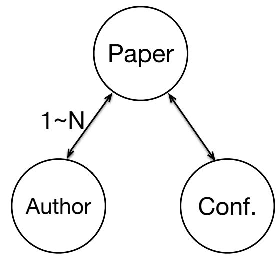


Fig. 7: Network scheme of DBLP dataset

Table 8: Top-5 most similar authors to "Yizhou Sun" and "Charu C. Aggarwal"

Rank	Yizhou Sun		Charu C. Aggarwal	
	APCPA	**A(i)PCP(j)A$\|i=j$**	**APCPA**	**A(i)PCP(j)A$\|i=j$**
1	Yizhou Sun	Yizhou Sun	Charu C. Aggarwal	Charu C. Aggarwal
2	Wilfred Ng	Rainer Gemulla	Jian Pei	Graham Cormode
3	Yannis Throdoridis	Daniele Braga	Hans-Peter Kriegel	Hans-Peter Kriegel
4	Ira Assent	Yiping Ke	Vipin Kumar	Fei Wang
5	Panayiotis Tsaparas	Gagan Agrawal	Eamonn J. Keogh	Wei Fan

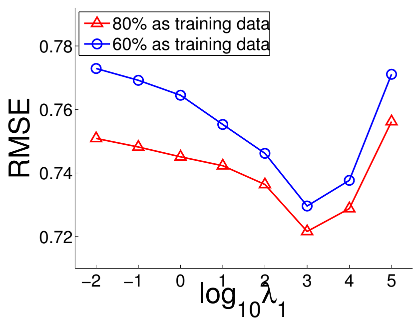
authors (A), papers (P) and conferences (C). Many work performed on bibliographic network ignored the order of authors to their papers, which can be considered as the attribute value on "author-paper" relation and makes bibliographic network a weighted HIN. A basic task in bibliographic network is top-k similarity search for a given author. In order to evaluate the performance of weighted meta paths in top-k similarity search, we perform the following experiments on a DBLP dataset including 23661 papers, 26741 authors, 20 conference and 73603 links connecting authors and papers. In this experiment, we compare the similarity search results based on APCPA path, a conventional meta path which connects authors publishing papers in the same conference, and A(i)PCP(j)A$|i = j$ path, a weighted meta path which connects authors publishing papers in the same conference with the same author order. The attribute values (i.e, i and j) in the A(i)PCP(j)A$|i = j$ path represent the order of co-authors in their publication.

We use PathSim as the similarity measure and measure the similarity between two target authors and other authors. One target author is Yizhou Sun, an assistant professor of University of California, Los Angeles, and the other one is Charu C. Aggarwal, a distinguished research staff member at the IBM Watson Research. Top-5 most similar authors to Sun and Aggarwal are shown in Table 8 respectively, ordered by the similarity scores measured by PathSim.

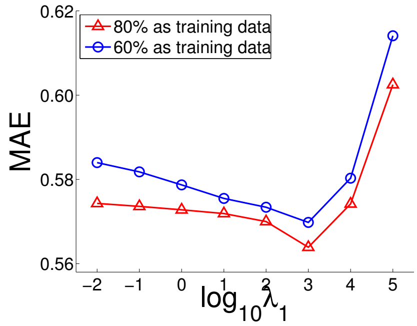
As Table 8 shows, the most similar authors to Yizhou Sun, based on these two meta paths, are different. Under the conventional meta path APCPA, Wilfred Ng and Yannis Throdoridis, who published a lot of papers in the conference that Yizhou Sun has attended, have the best similarity with Sun, whereas under the weighted meta path A(i)PCP(j)A$|i = j$, it is Rainer Gemulla and Daniele Baniele. Rainer Gemulla and Daniele Braga not only share numbers of conferences with Yizhou Sun, but also are the first or second author in most of their papers, meaning that they are prior author of papers like Sun. In addition, Wilfred Ng and Yannis Throdoridis acted as the third or less important author, so they are more like a supervisor when involved in a paper. In this query, weighted meta path can capture not only information about attendance of same venues, like conventional meta path does, but also the role in their publications.

The top-5 most similar authors to Charu C. Aggarwal also hold the same situation. Aggarwal published most of his papers as the first author according to our dataset, so did Graham Cormode, a professor at University of Warwick. However, Jian Pei, a professor of Simon Fraser University, acted as a supervisor in most of his publications, so he is in the ranking list of APCPA, but not in the list of A(i)PCP(j)A$|i = j$.

As results show, similarity measured on conventional meta path (e.g., APCPA) can find the peer authors, while the weighted meta path (e.g., A(i)PCP(j)A$|i = j$) can not only find the peer authors, but also consider more subtle semantic information in the "paper-author" relation. Let's talk about the reason behind this result. The conventional meta path APCPA fails to distinguish contribution of different co-authors to their publication, so similarity along this path is measured only based on



(a) RMSE



(b) MAE

Fig. 8: Impact of λ_1 parameter on performances.

attending same conferences. The path A(i)PCP(j)A$|i = j$ takes co-authors' order into consideration and can distinguish different roles of co-authors to their publication. Such a semantic meaning in the weighted meta path results in more accurate path-based similarity.

5.8 Study on Parameters

As we have said, the unique parameter λ_1 in SemRec controls the personalized levels in our model (see Eq. (9)). Here we analyze how the changes of λ_1 affect the recommendation accuracy. Fig. 8 shows the impact of λ_1 on RMSE and MAE in the SemRec$_{\mathrm{Reg}}$ method. We use 60% and 80% training data of Douban dataset to perform the experiments. From Fig. 8, we can see that on both training settings, the RMSE and MAE values both decrease (i.e., performances improve) at first. After λ_1 reaches a specific threshold around 1000, the RMSE and MAE values increase (i.e., performances drop) with the increase of λ_1. As we have mentioned, the SemRec$_{\mathrm{Reg}}$ changes from personalized weight learning to unified weight learning when λ_1 alters from 0 to $+\infty$. So through setting proper $\lambda_1 \in [0, +\infty)$, SemRec$_{\mathrm{Reg}}$ can utilize the benefits of personalized weight learning and avoid the rating sparsity. As the experiment suggested, we set λ_1 as 1000 in the above experiments.

6 CONCLUSION

In this paper, we extend conventional HIN and meta path for information networks with attribute values on links, and apply them on recommender system. We propose weighted HIN and weighted meta path to more subtly depict object relations through distinguishing link attribute values, and put forwards the similarity measure strategy based on weighted meta path. Furthermore, we design a novel semantic path based personalized recommendation method SemRec. The SemRec method not only flexibly integrates heterogeneous information through setting meta paths, but also obtains the prioritized and personalized weights representing user preferences on paths. Extensive experiments illustrate the effectiveness of SemRec. Since weighted meta paths contain rich semantics, future work includes a recommender system based on our SemRec method. We can realize semantic recommendation via choosing different meta paths for different recommendation models such as UMU path for collaborative recommendation and UU path for social recommendation. Personalized weight preferences on paths indicate recommendation explanation, which is another key feature of the system. The system will tell users which meta path is of high weight and the most similar user to them according to the weights.

References

1. Burke, R., Vahedian, F., Mobasher, B.: Hybrid recommendation in heterogeneous networks. In: UMAP, pp. 49–60 (2014)
2. Cao, X., Zheng, Y., Shi, C., Li, J., Wu, B.: Meta-path-based link prediction in schema-rich heterogeneous information network. International Journal of Data Science and Analytics **3**(4), 285–296 (2017)

3. Feng, W., Wang, J.: Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In: KDD, pp. 1276–1284 (2012)
4. Han, J.: Mining heterogeneous information networks: the next frontier. In: KDD, p. Keynote speech (2012)
5. Haveliwala, T.H.: Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. IEEE TKDE pp. 784–796 (2003)
6. Jamali, M., Lakshmanan, L.V.: Heteromf: Recommendation in heterogeneous information networks using context dependent factor models. In: WWW, pp. 643–653 (2013)
7. Ji, M., Han, J., Danilevsky, M.: Ranking-based classification of heterogeneous information networks. In: KDD, pp. 1298–1306 (2011)
8. Kuo, T.T., Yan, R., Huang, Y., Kung, P.H., Lin, S.D.: Unsupervised link prediction using aggregative statistics on heterogeneous social networks. In: SIGKDD, pp. 775–783 (2013)
9. Lao, N., Cohen, W.: Fast query execution for retrieval models based on path constrained random walks. In: KDD, pp. 881–888 (2010)
10. Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. Machine Learning **81**(2), 53–67 (2010)
11. Lee, S., Park, S., Kahng, M., Lee, S.: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. Expert Systems with Applications **40**, 684–697 (2013)
12. Lee, S., Song, S., Kahng, M., Lee, D., Lee, S.: Random walk based entity ranking on graph for multidimensional recommendation. In: RecSys, pp. 93–100 (2011)
13. Li, X., Chen, H.: Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach, decision support systems. Decision Support Systems **54**, 880–890 (2013)
14. Li, Y., Shi, C., Philip, S.Y., Chen, Q.: Hrank: A path based ranking method in heterogeneous information network. In: WAIM, pp. 553–565 (2014)
15. Lin, C.J.: Projected gradient methods for non-negative matrix factorization. In: Neural Computation, pp. 2756–2279 (2007)
16. Lippert, C., Weber, S.H., Huang, Y., Tresp, V., Schubert, M., Kriegel, H.P.: Relation prediction in multi-relational domains using matrix factorization. In: NIPS Workshop on Structured Input Structure Output (2008)
17. Liu, X., Yu, Y., Guo, C., Sun, Y.: Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation. In: CIKM, pp. 121–130 (2014)
18. Luo, C., Pang, W., Wang, Z., Lin, C.: Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations. In: ICDM, pp. 917–922 (2014)
19. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: SIGIR, pp. 203–210 (2011)
20. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: Social recommendation using probabilistic matrix factorization. In: CIKM, pp. 931–940 (2008)
21. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM, pp. 287–296 (2011)
22. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS, vol. 20 (2008)
23. Shi, C., Kong, X., Huang, Y., Yu, P.S., Wu, B.: Hetesim: A general framework for relevance measure in heterogeneous networks. IEEE Transactions on Knowledge and Data Engineering **26**(10), 2479–2492 (2014)
24. Shi, C., Li, Y., Zhang, J., Sun, Y., Yu, P.S.: A survey of heterogeneous information network analysis. CoRR abs/1511.04854
25. Shi, C., Liu, J., Zhuang, F., Philip, S.Y., Wu, B.: Integrating heterogeneous information via flexible regularization framework for recommendation. Knowledge and Information Systems **49**(3), 835–859 (2016)
26. Shi, C., Zhang, Z., Luo, P., Yu, P.S., Yue, Y., Wu, B.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In: CIKM, pp. 453–462 (2015)
27. Srebro, N., Jaakkola, T.: Weighted low-rank approximations. In: ICML, pp. 720–727 (2003)
28. Sun, Y., Han, J.: Mining heterogeneous information networks: A structural analysis approach. SIGKDD Explorations **14**(2), 20–28 (2012)
29. Sun, Y., Han, J., Yan, X., Yu, P., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In: VLDB, pp. 992–1003 (2011)
30. Sun, Y., Norick, B., Han, J., Yan, X., Yu, P.S., Yu, X.: Integrating meta-path selection with user guided object clustering in heterogeneous information networks. In: KDD, pp. 1348–1356 (2012)
31. Vahedian, F., Burke, R., Mobasher, B.: Weighted random walks for meta-path expansion in heterogeneous networks. In: RecSys 2016 Poster Proceedings (2016)
32. Wang, W., Yin, H., Chen, L., Sun, Y., Sadiq, S., Zhou, X.: Geo-sage: A geographical sparse additive generative model for spatial item recommendation. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1255–1264. ACM (2015)
33. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 15–24. ACM (2016)
34. Yin, H., Zhou, X., Shao, Y., Wang, H., Sadiq, S.: Joint modeling of user check-in behaviors for point-of-interest recommendation. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1631–1640. ACM (2015)
35. Yu, X., Ren, X., Gu, Q., Sun, Y., J.Han: Collaborative filtering with entity similarity regularization in heterogeneous information networks. In: IJCAI HINA (2013)
36. Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., Han, J.: Personalized entity recommendation: A heterogeneous information network approach. In: WSDM, pp. 283–292 (2014)
37. Yuan, Q., Chen, L., Zhao, S.: Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In: RecSys, pp. 245–252 (2011)
38. Zhang, Z., Zhou, T., Zhang, Y.: Personalized recommendation via integrated diffusion on usercitemctag tripartite graphs. Physica A: Statistical Mechanics and its Applications **389**, 179–186 (2010)
39. Zheng, J., Liu, J., Shi, C., Zhuang, F., Li, J., Wu, B.: Dual similarity regularization for recommendation. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 542–554. Springer (2016)
40. Zheng, J., Liu, J., Shi, C., Zhuang, F., Li, J., Wu, B.: Recommendation in heterogeneous information network via dual similarity regularization. International Journal of Data Science and Analytics **3**(1), 35–48 (2017)