Supervised ranking framework for relationship prediction in heterogeneous information networks

Wenxin Liang[1] · Xiao Li[1] · Xiaosong He[2] · Xinyue Liu[1] · Xianchao Zhang[1]

Published online: 15 August 2017

Abstract In recent years, relationship prediction in heterogeneous information networks (HINs) has become an active topic. The most essential part of this task is how to effectively represent and utilize the important three kinds of information hidden in connections of the network, namely *local structure information* (Local-info), *global structure information* (Global-info) and *attribute information* (Attr-info). Although all the information indicates different features of the network and influence relationship creation in a complementary way, existing approaches utilize them separately or in a partially combined way. In this article, a novel framework named Supervised Ranking framework (S-Rank) is proposed to tackle this issue. To avoid the class imbalance problem, in S-Rank framework we treat the relationship prediction problem as a ranking task and divide it into three phases. Firstly, a Supervised PageRank strategy (SPR) is proposed to rank the candidate nodes according to Global-info and Attr-info. Secondly, a Meta Path-based Ranking method (MPR) utilizing Local-info is proposed to rank the candidate nodes based on their meta path-based features. Finally, the two ranking scores are linearly integrated into the final ranking result which combines all the Attr-info, Global-info and Local-info together. Experiments on DBLP data demonstrate that the proposed S-Rank framework can effectively take advantage of all the three kinds of information for relationship prediction over HINs and outperforms other well-known baseline approaches.

Keywords Relationship prediction · Ranking strategy · Meta path · Heterogeneous information networks

1 Introduction

Relationship Prediction is known as *Link Prediction* initially. Link prediction problem is formally defined by Liben-Nowell and Kleinberg [15]. Given a snapshot of a network at time t, link prediction aims to predict new links created in future time interval $[t, t']$. There are various applications that substantially take advantage of link prediction frameworks or algorithms. In social networks, individuals can efficiently and effectively find companions, assistants, or colleagues [11]. In academic communities, researchers can easily apply link prediction methods to discover potential collaborators by predicting the co-author relationship [25], or find highly related publications by mining the citation prediction problem [34].

A preliminary version of this article appeared in proceedings of the 29th International Conference on Industrial, Engineering and Other Applications of Applied Intelligence Systems (IEA/AIE 2016) [14].

✉ Wenxin Liang
wxliang@dlut.edu.cn

Xiao Li
lixiao_dlut@163.com

Xiaosong He
hexiaosong@abitai.com

Xinyue Liu
xyliu@dlut.edu.cn

Xianchao Zhang
xczhang@dlut.edu.cn

[1] School of Software, Dalian University of Technology, No. 321 Tuqiang Street, Jinzhou District, Dalian 116620, China

[2] A Bit AI Co., Ltd, Danleng SOHO, No. 6 Danleng Street, Haidian District, Beijing 100060, China

Most link prediction approaches are proposed based on traditional ideal networks, i.e., homogeneous networks, which contain only one type of link and node. However, in the real world, the relations between objects mostly generate more complicated networks, namely *Heterogeneous Information Networks* (HINs), in which nodes and links are of multiple types [8]. Researchers [25, 27] conclude that the link prediction problem could be extended to an analogous but more general problem, namely the relationship prediction problem in the context of HINs. Relationship prediction aims to predict a specific relation between two types of nodes. In contrast of a single link, a relation might be represented by a sequence of links, which also makes the relationship prediction tasks more complex than the link prediction ones. The emergence of relationship is predictable due to such assumption that there exists latent information encoded in links and nodes, which is the main reason why the relationship was generated in networks. Therefore, the key issue of relationship prediction in HINs is how to effectively represent, extract and model the latent information. The latent information in HINs can be generally divided into two types: structure information and attribute information.

Firstly, topological features between objects can be referred to as structure information. In this article, we divide the structure information into two categories by reviewing existing methods [15, 25, 33], namely *local structure information* (Local-info) and *global structure information* (Global-info), which reflect different aspects of structural features. Figure 1 shows a small citation network, where the arrows from circles to squares mean authors write papers and the arrows from squares to squares mean one paper cites another paper. From this figure we illustrate the difference between Local-info and Global-info through a simple relationship prediction problem: predicting which paper will probably be cited by author A in the future, P_1 or P_6?

It is obvious that there are more paths between A and P_6 than those between A and P_1. Considering the existence of more common neighbors, P_6 is more likely to be cited by A in the future. This kind of information (number of paths, common neighbors between two nodes, etc.) is represented as Local-info. Meta path [26], connecting two types of objects through different object type composition, is one of the typical methods using the Local-info in HINs. On the other hand, P_1 has higher degrees (indegrees or outdegrees) than P_6, which means P_1 may have higher influence and reputation in the network. In reality, P_1 may be a famous paper in the related academic field of A. Hence, it is of high probability that A will cite P_1 in the future. This kind of feature (degree, global reputation, etc.) is viewed as Global-info. The most representative method using Global-info is PageRank [20] which assigns high value to the nodes with high degrees. Therefore, Local-info and Global-info are both useful but represent two totally different insights when solving the relationship prediction problem using network topologies.

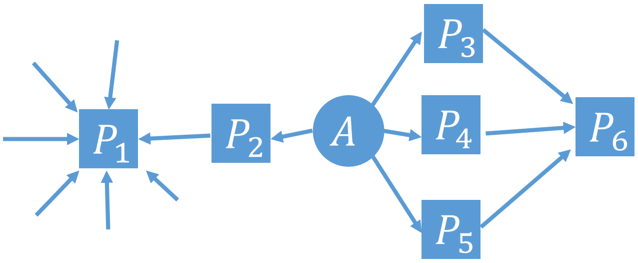


Fig. 1 A simple example of citation networks

Secondly, emergence of new relations are usually generated by particular reasons because HINs represent real-world networks. This means there are abundant semantic meanings hidden in links. In such bibliography networks as Fig. 1, there are many reasons for an author to cite a particular paper: the paper is the latest research; the paper is highly related to the author's papers, etc. There are many features related to edges and objects in HINs, such as published year and relevancy between papers in the above-mentioned example. Such kind of features are denominated as *attribute information* (Attr-info). Thus, Attr-info also has significant impact on the creation of new relationships.

Global-info, Local-info and Attr-info are all useful information for relationship prediction and they work in a different but complementary way. Local-info may lose the position of node in the whole network, Global-info is biased to highly visible objects, and Attr-info has deep significance in modeling real-world behaviors. Therefore, it is essential to combine all three kinds of information together to represent the creation of links in a better way.

To the best of our knowledge, there is no existing method that combines all the three kinds of information together in HINs, since it is not an easy task to effectively integrate all the information in one framework. To tackle this issue, in this article, a novel supervised ranking framework (S-Rank) is proposed for the relationship prediction in HINs, which completely combines Global-info, Local-info and Attr-info together. S-rank framework carries out in three phases. In the first phase, a Supervised PageRank method (SPR) is firstly utilized to capture the rich Attr-info and Global-info hidden in HINs. A set of feature vectors is defined to represent the Attr-info of different types of links. Then, the capacity of each link can be computed by assigning weights to the feature vector. Finally, all nodes will obtain a score by iteratively computing PageRank which can capture Global-info simultaneously. In the second phase, a Meta Path-based Ranking method (MPR) is proposed to score nodes utilizing meta path-based measures which can capture the Local-info. In the final phase, the results of SPR and MPR are integrated together to obtain the final ranking result. The differences between our S-Rank framework and the existing methods are summarized in Table 1.

The main contributions of this article are summarized as follows.

- We deeply analyze and discuss how the three kinds of information (Global-info, Local-info and Attr-info) impact on the relationship prediction over HINs.
- We propose a novel three-phase Supervised Ranking framework (S-Rank) for the relationship predication problem in HINs. To the best of our knowledge, our work is the first to completely combine Global-info, Local-info and Attr-info together.
- In the first phase of S-Rank, we propose a Supervised PageRank strategy (SPR) to rank the candidate nodes. In SPR, the capacity of each link is computed according to attribution features of the link. Therefore, both the Global-info and the Attr-info can be utilized simultaneously.
- In the second phase of S-Rank, we propose a Meta Path-based Ranking method (MPR) which uses Local-info to rank the candidate nodes based on their meta path-based features.
- In the final phase of S-Rank, we linearly integrated the two ranking scores into the final ranking result which combines all the Attr-info, Global-info and Local-info together.
- Experiments on DBLP show that our framework can effectively incorporate three kinds of information and significantly outperforms the baseline methods.

In the rest of the article, we first introduce the preliminaries about heterogeneous information network and formalize the problem in Section 2. The proposed S-Rank framework is described in Section 3. Experiments and results are presented in Section 4. Finally, we review the related work in Section 5 and conclude our work in Section 6.

A preliminary version of this article was presented at "the 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems" (IEA/AIE 2016) [14]. In this article, we extend the following contents. In Section 1, a table (Table 1) is added to summarize the difference between our proposed method and the existing ones. In Section 2.1, we explain main concepts about heterogeneous information network and particularly introduce DBLP network that is used in this article. In Section 2.2, we explain why S-Rank framework can deal with the imbalanced dataset. An additional graph is presented in this section to show an overview of S-Rank framework. More detailed derivation and computing methodology of some complicated formulas are supplemented in Section 3. In the end of Section 3, we discuss the complexity of S-Rank framework. Then, we reorganize the experiments part in Section 4, in which more details are declared. In Section 4.2, we add one more measure named PathSim that is an excellent method to measure the relationship between two nodes and present an example to illustrate the measures. Additional experiments that discuss the impact of training time and restart parameters are performed and discussed in Sections 4.4.4 and 4.4.5, respectively. Finally, related work is conducted as an individual section in Section 5.

2 Problem definition

In this section, we introduce the heterogeneous information network and formulate the relationship prediction problem in this network context.

2.1 Heterogeneous information network

A heterogeneous information network (HIN) is an information network that involves multiple types of objects and relations. In this section, in order to describe the HIN more precisely and formally, we slightly modify the definition of **Information Network** and **Network Schema** in [26] to follow the context of this article.

Definition 1 (**Information Network**) An Information Network is defined as a directed graph $G = (V, E, F)$ with an object type mapping function $\phi : V \to \mathcal{A}$, an edge type mapping function $\varphi : E \to \mathcal{R}$ and a feature type mapping function $\delta : F \to \mathcal{T}$, where each object $v \in V$ belongs to a particular object type $\phi(v) \in \mathcal{A}$, each edge $e \in E$ belongs to a particular relation type $\varphi(e) \in \mathcal{R}$, and each feature $F \in F$ belongs to a particular feature type associated with an edge $\delta(f) \in \mathcal{T}$.

Table 1 Comparison of some existing methods and our s-rank framework

Method	Attr-info	Local-info	Global-info	Supervised	HIN	Universal
PathPredict [25]		√		√	√	√
Bucket [34]	√	√		√	√	
RW_ALL [13]			√		√	
SRW [1], SSP [7]	√		√	√		√
S-Rank	√	√	√	√	√	√

When both the types of objects $|\mathcal{A}| > 1$ and the types of edges $|\mathcal{R}| > 1$, the network is called *heterogeneous information network* (HIN); otherwise it is a *homogeneous information network*, which is the model of traditional network. Set F denotes the rich attribute information encoded in HINs (we attach both object attributes and edge attributes information to form the feature vector f). Since f is associated with edges, we have $|\mathcal{R}| = |\mathcal{T}|$, so feature vector for the each edge type has the corresponding feature representation. Formally, we can denote F as $F = \{f_t | t \in \mathcal{T}\}$, and $f_t = (f_t^1, f_t^2, ..., f_t^n)$ where n is the number of features of edge type t.

Definition 2 (**Network Schema**) The network schema is a meta template of a HIN $G = (V, E, F)$ with the object type mapping function $\phi : V \rightarrow \mathcal{A}$, the edge mapping function $\varphi : E \rightarrow \mathcal{R}$ and the feature type mapping function $\delta : F \rightarrow \mathcal{T}$, which is a directed graph defined over object types $\mathcal{A}$, with edges as relations from $\mathcal{R}$ as well as attribute template $\mathcal{T}$ and denoted as $T_G = (\mathcal{A}, \mathcal{R}, \mathcal{T})$.

The definition of network schema is similar to the ER (Entity-Relationship) model in database systems. It serves as a template for a specific network, and defines the rules of how entities exist, how relationship should be created and how features in both edge and node should be extracted. Based on the definition of network schema, we can apply basic graph search methods (such as BFS) to obtain the *meta path* [26, 34].

Definition 3 (**Meta Path**) A meta path $\mathcal{P} = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} ... \xrightarrow{R_l} A_l$ is a path defined on the graph of network schema $T_G = (\mathcal{A}, \mathcal{R})$, where $A_i \in \mathcal{A}$ and $R_i \in \mathcal{R}$ for $i = 0, ..., l$, l is called the length of meta path. For $i = 1, ..., l-1$, $A_0 = dom(R_1)$, $A_l = range(R_l)$ and $A_i = range(R_i) = dom(R_{i+1})$, where the start node type and end node type of $\mathcal{P}$ are defined as $dom(\mathcal{P}) = A_0$ and $range(\mathcal{P}) = A_l$, respectively.

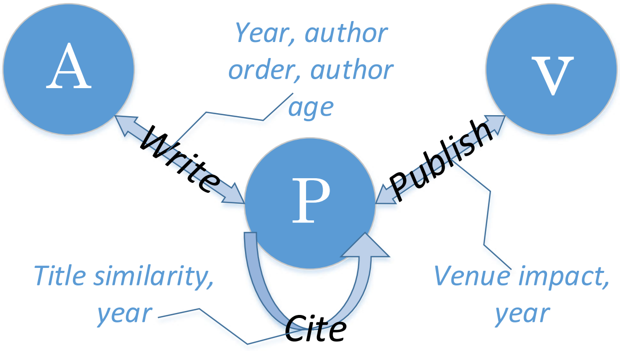


Fig. 2 Network schema of DBLP

Meta path is a kind of pattern. A mass of path instances in a network can be obtained by following a given meta path. For convenience, we remove the edge type and denote meta path as: $P = A_0 \rightarrow A_1 \rightarrow ... \rightarrow A_l$. To make the definition of network schema and meta path more clear, here we present an exquisite example.

Figure 2 is the network schema of DBLP,[1] in which there are three kinds of objects. A indicates authors, P represents papers and V denotes venues (conferences or journals). In addition, there exist three types of edges ($|\mathcal{R}| = 3$) as well as edge features ($|\mathcal{T}| = 3$) in the networks. In this article, both edge attributes (title similarity and author order) and object attributes (year, venue impact and author age) are included in set F. Comparing with [7], we do not treat attributes in edge and object separately, since the same attribute (e.g., year) may have various impacts when it is placed in diverse types of relations for HINs. Examples of meta path exists in DBLP are given in Table 2.

Previous studies [25, 27] have defined the **target relation** as either a relation in $\mathcal{R}$ or a composite relation described by a meta path. In the example network schema of DBLP (Fig. 2), the citation relation can be defined by meta path $A - P \rightarrow P$ while path $A - P - V - P - A$ denotes the relation that two authors publish papers in the same venue. The **relationship** between objects can be referred as instances of the target relation. Relationship prediction is to predict whether there will generate a relationship instances that follows target relation. In addition, Objects may have relationships several times. For instance, Jiawei Han and Philip S. Yu have co-authored papers many times, which indicates that it is of high probability for Philip S. Yu to have the co-author target relation with Jiawei Han. In other words, a path instance carries relationship information that follows the relation defined by its meta path in the network.

2.2 Problem formulation

Generally, given a HIN $G = (V, E, F)$, a source node $s \in V$ and a set of candidate nodes $\mathbf{C} \subset V$ to which s may create relationships. A meta path $\mathcal{P}$ represents the target relation and the function $\phi(s)$ gets the node type of s (notations used in definitions and formulation as well as the rest part of the article can be found in Table 3). Then the relationship prediction task is to predict whether there will be a relationship between two nodes $s \in V$ and $v \in \mathbf{C}$ in the future, where $\phi(s) = dom(\mathcal{P})$ and $\phi(v) = range(\mathcal{P})$. Some details about how $\mathbf{C}$ is selected are stated as following: $\phi(s)$ and $\phi(v)$ can either be identical or not; the self-relation is not taken into consideration, i.e., $s \notin \mathbf{C}$; we are interested in predicting new relationships rather than repeated relationships,

[1] http://www.informatik.uni-trier.de/~ley/db/.

Table 2 Examples of meta paths

Meta Path	Description	Length
$A - P$	An author writes a paper	1
$P \rightarrow P$	A paper cites another paper	1
$A - P \rightarrow P$	An author cites a paper	2
$A - P - A$	Two authors co-author one paper	2
$A - P - V - P - A$	Two authors publish papers in the same venue	4
$A - P - A - P \rightarrow P$	An author's co-author cite a paper	4

and hence s never have any relationships with v. Moreover, although we only consider the situation of a single source node, our framework can be easily generalized for the prediction tasks that contain a group of source nodes.

We address the relationship prediction problem in a ranking manner, i.e., the proposed algorithm will assign higher score to the nodes that have more relationship instances with s. The $score(v)$ is computed differently in the two phases of S-Rank. In Supervised PageRank (SPR), $score(v)$ is computed through modified PageRank model, while in Meta Path-based Ranking (MPR), it is computed by meta path-based measures. Actually, $score(v)$ indicates the probability that v will create target relation with the source node s in the future.

Table 3 List of notations

Notation	Description
G	HINs
$V, \mathcal{A}$	objects in HINs, object types
$E, \mathcal{R}$	edges in HINs, edge types
$F, \mathcal{T}$	feature set, feature types
ϕ, φ, δ	the mapping functions of types
$\mathcal{P}$	a meta path
$dom(\mathcal{P})$	the start node type of a meta path
$range(\mathcal{P})$	the end node type of a meta path
$\mathcal{M}_{\mathcal{P}}$	a measure based on $\mathcal{P}$
p	the stationary distribution of PageRank
Q	the probability transition matrix in PageRank
W	weight vector set associated with feature types
g	weight vector associated with $\mathcal{M}_{\mathcal{P}}$
c_{uv}	the edge capacity
$\pi_{w_i}(f)$	calculation function for c_{uv}
$\mathbf{C}$	candidate node set
$\mathbf{T}$	training set of node pairs
$\mathbf{P}$	selected meta paths set
$Rank_1$	result of the first phase of S-Rank
$Rank_2$	result of the second phase of S-Rank
$SRank$	result of S-Rank

$\mathbf{L}_n$ is defined to represent the set of labeled nodes, to which s has ever created relation $\mathcal{P}$ for n times ($n \geq 0$). It is obvious that $\mathbf{L}_0 = \mathbf{C}$, and $\forall v \in \mathbf{L}_n$, $\phi(v) = range(\mathcal{P})$. Then, a training set of pairs is generated as: $\mathbf{T} = \{\langle u, v\rangle | u \in \mathbf{L}_i, v \in \mathbf{L}_j \ and \ i > j\}$. Hence, the potential meaning for each pair $\langle u, v\rangle \in \mathbf{T}$ is that s is more likely to have relationships with node u than v. This methodology of generating training set can effectively avoid the imbalance problem. As mentioned in [19], relationship prediction dataset is extremely *imbalance*. The number of relationships known to be present is often significantly less than the number of relationships known to be absent. The fact that the training set has fewer examples of one class poses a difficulty for traditional classifiers which aim to infer reliable patterns in a supervised way. However, the size of training set is guaranteed in S-Rank, since $\mathbf{T}$ is the combination of nodes, to which s has ever created relationships for many times.

Take simplified DBLP network illustrated in Fig. 3 as an example, node s is an author as the source node, and other nodes are papers. The green part is a specific period of the network, the blue part indicates new links generate in future network, and the links are citation relationships denoted by meta path $A - P \rightarrow P$. Then $\mathbf{C} = \{1, 4, 5, 6, 7\}$, since these nodes have never been cited by s before. Instead of simply setting $\{1, 2, 3, 4\}$ as training set, a set of pairs of nodes is formed as $\mathbf{T} = \{\langle 2, 3\rangle, \langle 2, 1\rangle, \langle 2, 4\rangle, \langle 3, 1\rangle, \langle 3, 4\rangle\}$, which can increase the number of training samples and avoid the imbalance problem. Now, the goal is to predict which nodes in $\mathbf{C}$ will be cited by s. In this article, S-rank will score these 5 papers to assess the probability they may create links to s, then rank them in descending order. The performance of S-rank can be evaluated by comparing the result with real future network (the blue part).

3 S-rank framework

In this section, a three-phase framework, S-Rank, is proposed. We first introduce the overview of the supervised ranking model. Then two supervised ranking phases are proposed and illustrated in detail. Finally, the results of the two phases are integrated into the final ranking result.

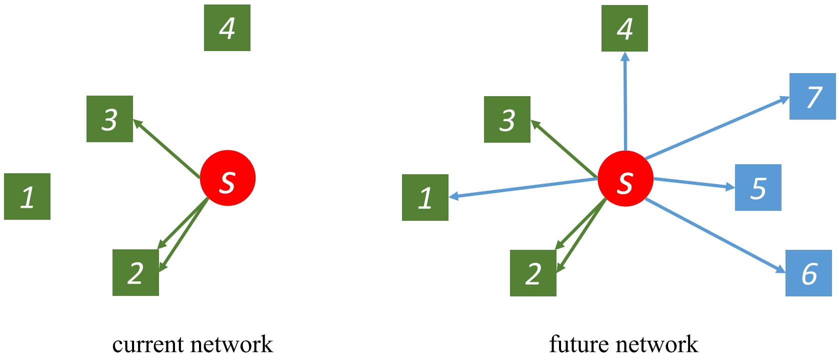


Fig. 3 A simple example for relationship prediction

3.1 Framework overview

Following the similar work [25], we divide the supervised framework into two stages, i.e., **Training Stage** and **Testing Stage**. To simulate a dynamic network, we will partition two time intervals as current network and future network for both stages. In the training stage, the nodes that never have relationships with the source node s in time interval $T_0 = [t_0, t_1]$ are selected as candidate node set $\mathbf{C}$ and the features are also extracted; then label information from time interval $T_1 = [t_1, t_2]$ are extracted to form the supervised node pair set $\mathbf{T}$. In the testing stage, we extract features in the time interval $T_0{}' = [t_0{}', t_1{}']$, and apply the learned knowledge in the training stage to predict new relationships in time interval $T_1{}' = [t_1{}', t_2{}']$. Finally, we can evaluate the prediction according to the ground truth in $T_1{}'$. We take the DBLP network as an example to illustrate our S-Rank framework in Fig. 4. Actually, T_1 and $T_1{}'$ are the future status for the network in T_0 and $T_0{}'$. On the other hand, T_0 is the history for $T_0{}'$ from which we can learn the patterns.

3.2 Supervised ranking model

Inspired by the outstanding work [1], a supervised ranking technology is proposed to learn the hidden patterns (a weight vector θ) from the historical data $\mathbf{T}$. More specifically, θ in Supervised PageRank (SPR) will guide the walker to visit those nodes to which s will create relationships in the future; θ in Meta Path-based Ranking Method (MPR) indicates the importance of different meta paths in the process of creating relationships. The objective of the supervised ranking model is to minimize the following function.

$$\min_{\theta} F(\theta) = ||\theta||^2 + \lambda \sum_{\langle u,v\rangle \in \mathbf{T}} l(score(v) - score(u)), \quad (1)$$

where $||\theta||^2$ is the regular term that prevents overfitting. Considering the underlying meaning of pair $\langle u, v\rangle$, namely s is more likely to have relationships with node u than v, $l(.)$ will assign a non-negative penalty based on $score(v) - score(u)$. If $score(v) - score(u) >= 0$, $l(.) > 0$; otherwise $l(.) = 0$. Parameter λ controls how the fitness of the model

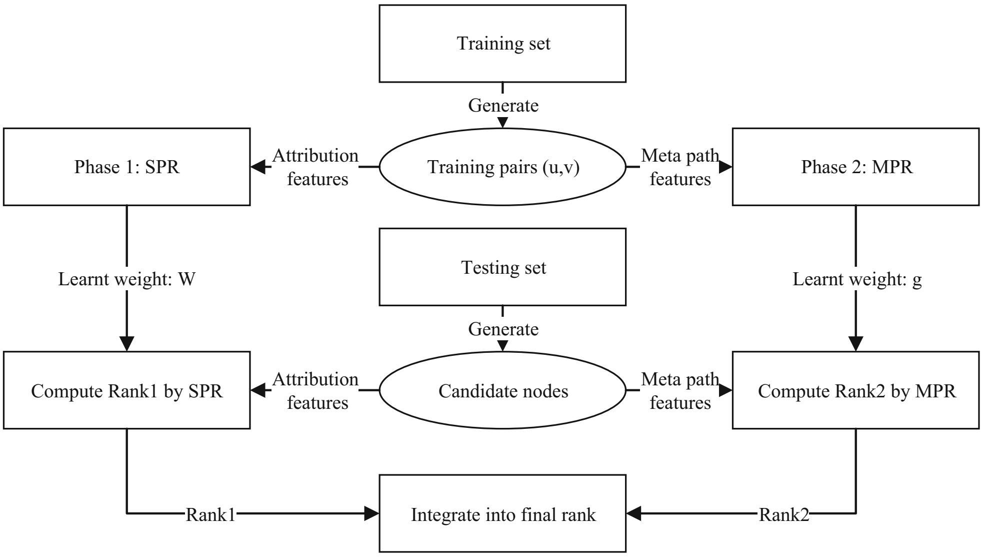


Fig. 4 S-Rank framework

affects the optimal value. This model is applied in both SPR and MPR.

3.3 First phase: supervised pagerank (SPR)

PageRank, as a powerful method that can capture the Global-info in graph ranking, can be expressed by Eq. 2. The vector p is the stationary distribution of the PageRank. Q is the probability transition matrix. In this subsection, we aim at presenting a mixture model that combines both the Global-info and Attr-info.

$$p^T = p^T Q \tag{2}$$

Given a HIN $G = (V, E, F)$ contains rich attribute information ($f \in F$), the challenge is to make a connection between the weight vector θ and the feature vector F in HINs in order to apply (1) to learn θ.

To achieve this goal, the weight vector θ is extended to a weight vector set $W = \{w_1, w_2, ..., w_n\}$, where $n = |\mathcal{R}|$. w_i is utilized to weight F, namely for each edge $(u, v) \in G$, we can calculate the capacity $c_{uv} = \pi_{w_i}(f)$ by combining w_i and $f \in F$. Function $\pi_{w_i}(f)$ takes the product of w_i and F as input. c_{uv} indicates the ability that edge (u, v) can guide s to visit the target nodes. Then the conductance matrix Q' is computed as:

$$Q'_{uv} = \begin{cases} \frac{c_{uv}}{\sum_w c_{uw}} & if \;\; (u, v) \in E \\ 0 & otherwise. \end{cases} \tag{3}$$

Further, Q can be obtained by the general calculation in (4), where $0 < \alpha < 1$ is the probability of jumping (as against walking) in each step. $\alpha(v = s)$ means α is calculated only when $v = s$ and thus each row of Q sums to 1.

$$Q_{uv} = (1 - \alpha)Q'_{uv} + \alpha(v = s) \tag{4}$$

Thus, by modifying the objective function in (1), we can obtain (5), where p denotes the stationary distribution of the PageRank. Notice that there is a difference of objective function with previous work [1]. Considering the candidate set $\mathbf{C}$ and different meta path-based features of source authors, instead of minimizing the value of objective function for all the source authors in one equation, we train the weight W for each author multiple times and calculate the mean value as the final value of W.

$$\min_W F(W) = \sum_{w_i \in W} ||w_i||^2 + \lambda \sum_{\langle u,v \rangle \in \mathbf{T}} l(p_v - p_u) \tag{5}$$

To solve this optimal problem, a gradient descent-based method is applied. For each w_i, we can write the derivative as follows.

$$\begin{aligned} \frac{\partial F(W)}{\partial w_i} &= 2w_i + \lambda \textstyle\sum_{\langle u,v \rangle \in \mathbf{T}} \frac{\partial l(p_v - p_u)}{\partial w_i} \\ &= 2w_i + \lambda \textstyle\sum_{\langle u,v \rangle \in \mathbf{T}} \frac{\partial l(p_v - p_u)}{\partial (p_v - p_u)} (\frac{\partial p_v}{\partial w_i} - \frac{\partial p_u}{\partial w_i}). \end{aligned} \tag{6}$$

Given a loss function $l(.)$, it is simple to compute the derivative $\frac{\partial l(p_v - p_u)}{\partial (p_v - p_u)}$. However, it is not an easy task to compute $\frac{\partial p_u}{\partial w_i}$, since there exists recursive relation in (2). Notice that (2) can be rewritten as $p_u = \sum_j p_j Q_{ju}$ and the recursive equation is

$$\frac{\partial p_u}{\partial w_i} = \sum_j Q_{ju} \frac{\partial p_j}{\partial w_i} + p_j \frac{\partial Q_{ju}}{\partial w_i}, \langle j, u \rangle \in E. \tag{7}$$

L. Backstrom and J. Leskovec [1] adopts a power-method like algorithm to compute $\frac{\partial p_u}{\partial w_i}$. It recursively applies the chain rule to (7). We extend the algorithm to fit our problem. The extended algorithm is described in Algorithm 1. For each node u, it repeatedly computes the derivative $\frac{\partial p_u}{\partial w_i}$ for every kind of weight $w_i \in W$ based on the result obtained in previous iteration. When the algorithm ends, all $\frac{\partial p_u}{\partial w_i}$ are computed and can be directly applied to (6). As W is updated in each iteration in gradient descent, the probability transition matrix Q also needs to be updated. We preprocess it before Algorithm 1 starts and take Q as the input of Algorithm 1. Despite the fact that different nodes are distinct from node type, we initialize the value p with $\frac{1}{|V|}$ since the result of PageRank has nothing to do with the initial value.

Algorithm 1 Iterative computation of p and all $\frac{\partial p_u}{\partial w_i}$

```
for each $u \in V$ do
    ${p_u}^{(0)} = \frac{1}{|V|}$
end for
$t = 1$
while $not\ converged$ do
    for each $u \in V$ do
        $p_u = \sum_j {p_j}^{(t-1)} Q_{ju}$
    end for
    $t = t + 1$
end while
for $i = 1, ..., |\mathcal{R}|$ do
    Initialize $w_i$ with zero vector
    $t = 1$
    for $k = 1, ..., |w_i|$ do
        while $not\ converged$ do
            for each $u \in V$ do
                ${\frac{\partial p_u}{\partial w_{ik}}}^{(t)} = \sum_j Q_{ju} {\frac{\partial p_j}{\partial w_{ik}}}^{(t-1)} + p_j^{(t-1)} \frac{\partial Q_{ju}}{\partial w_{ik}}$
            end for
            $t = t + 1$
        end while
    end for
end for
```

To compute $\frac{\partial p_u}{\partial w_i}$, we further need to compute $\frac{\partial Q_{ju}}{\partial w_i}$. The following Equation

$$\frac{\partial Q_{ju}}{\partial w_i} = (1-\alpha)\frac{\frac{\partial c_{ju}}{\partial w_i}(\sum_k c_{jk}) - c_{ju}(\sum_k \frac{\partial c_{jk}}{\partial w_i})}{(\sum_k c_{jk})^2} \tag{8}$$

can be easily obtained by deriving (2) and (7). Remind that when $(j,u) \notin E$, its value is 0.

After finishing the computation of $\frac{\partial F(W)}{\partial w_i}$, a gradient descent method can be applied to minimize $F(W)$ directly. Gradient descent may not converge to a global minimum since the optimal problem is not convex. Other methods such as Genetic Algorithm (GA) can be also considered. But it is difficult to encode the answer since we have no idea about the value domain. In practice, we resolve this problem by randomly initializing W at several different starting points, and take the best answer as the result.

Although all nodes in HINs are ranked, we only concentrate on the nodes in candidate node set $\mathbf{C}$. Thus, when we conduct the evaluation, only value of p_v $(v \in \mathbf{C})$ is extracted from the stationary distribution of PageRank to calculate the ranking result.

3.4 Second phase: meta path-based ranking method (MPR)

Compared with PageRank which takes use of the Global-info, topologies represented by meta path can be referred as the Local-info in HINs, which has been proven to be an excellent local structure feature [25–27].

In MPR, for a candidate node $v \in \mathbf{C}$ and a source node s, we define the $score(v)$ in (1) as k_v by utilizing a linear regression model to distinguish different preferences on different meta paths.

$$k_v = \textstyle\sum_{\mathcal{P}\in\mathbf{P}} g_i \times \mathcal{M}_{\mathcal{P}}, v \in \mathbf{C}, \tag{9}$$

where $\mathcal{M}_{\mathcal{P}}$ is a meta path-based measure.

A meta path set $\mathbf{P}$ is carefully selected under specific semantics and the length of each $\mathcal{P} \in \mathbf{P}$ is limited (e.g., 5) since a meta path will be invalid when it has a large length. Sun et al. [26] presented the result that longer paths bring in farther neighbours, which actually do not have much relation with source nodes. Relatively short paths is good enough for measurements, while long ones may lose particularity of meta paths.

Different k_v can be obtained when different meta path measures $(\mathcal{M}_{\mathcal{P}})$ are chosen. g_i indicates the i-th weight associated with the i-th meta path. The goal in this phase is to learn the weight vector g, which represents the importance of each meta path in determining whether two nodes will have relationships in the future. Therefore, (1) is rewritten as follows.

$$\min_g F(g) = ||g||^2 + \lambda \sum_{\langle u,v\rangle\in\mathbf{T}} l(k_v - k_u). \tag{10}$$

It is straightforward to derive (10) to obtain $\frac{\partial F(g)}{\partial(g)}$. Here we also employ a gradient descent method to solve the optimal problem. According to the learned g, we can calculate k_v for each $v \in \mathbf{C}$ in testing stage and obtain the ranking result.

3.5 Third phase: integrating results

With the trained weights, candidate nodes are scored and ranked by p_v and k_v in SPR and MPR. We record the ranking result as $Rank_1$ and $Rank_2$, respectively.

$Rank_1$ has absorbed both the Global-info and the Attr-info, and meanwhile $Rank_2$ can represent the perspective of the Local-info. $Rank_1$ and $Rank_2$ are obtained from the same dataset and the same model (1) but represent two different insights independently. Under this circumstance, it is intuitive for us to integrate these two ranking results together. This situation is similar to Search Engine which combines PageRank score and document correlation score to give a good search result. The PageRank score contains the structural information and the document correlation score denotes the relevance between query and document from the aspect of textual information. We denote the integration procedure as:

$$SRank = \Re(Rank_1, Rank_2), \tag{11}$$

where $SRank$ is the final result of our framework. Technology about $\Re$ involves a classical issue, namely *Rank Aggregation*, which has been extensively studied [21]. In this article we mainly focus on how to integrate three kinds of information together. Therefore, we conduct experiments to investigate two widely adopted implementations of $SRank$, denoted as (12) and (13).

$$SRank(u) = Rank_1(u)^{\beta} \times Rank_2(u)^{1-\beta} \tag{12}$$

$$SRank(u) = Rank_1(u) \times \beta + Rank_2(u) \times (1-\beta), \tag{13}$$

where $\beta \in (0, 1)$ is a parameter that controls the trade-off between $Rank_1$ and $Rank_2$, which can be manually adjusted according to the performance of experiments.

3.6 Complexity analysis

It is obvious that the main expensive part of S-Rank exists in Algorithm 1. Thus, here we only focus on analyzing its complexity. Algorithm 1 is time-consuming since it has 4 loops from line 11 to line 22. In the most inner loop, each access to the element of Q can be done in $O(1)$, but it also requires to access the neighbour nodes of one node. This operation can

Table 4 Dataset of DBLP

Datasets	$\#A$	$\#V$	$\#P$	$\#A-P$	$\#P\rightarrow P$	$\#P-V$
Training set	882	631	2832	3597	1003	2832
Group1	1008	717	3540	4576	1495	3540
Group2	1235	852	4212	5890	2382	4212
Group3	1461	938	5611	8046	3871	5611
Total subset	2505	3373	50910	68922	60749	50910

reach $O(|V|)$ in the worst case. Therefore, the complexity of Algorithm 1 is $O(t \times |\mathcal{R}| \times |w| \times |V|)$. Although S-Rank has a relative high training complexity, it is fast in predicting phase. In practice, the training can be done off-line using parallelization technologies to improve the training performance.

4 Experiments

In this section, we conduct experiments to evaluate the effectiveness of S-Rank framework comparing with the state-of-art methods. We also make additional discussions about the impact of training time and restart parameters.

4.1 Experiment description

A real-world DBLP bibliographic dataset is used to create a HIN [29]. DBLP bibliographic dataset is a popular dataset to build a heterogeneous information network for relationship prediction. It records abundant publication data of papers, such as the authors, published year, published venue, citations, etc. Because citation information is required, we refer to the DBLP dataset provided by *Aminer.org*.[2] This dataset collects papers published from 1936 to 2010, which contains 2,084 k papers and 2,244 k citation relationships in total. Since the original network is too large, we generate a subset of DBLP by sampling the authors that published papers in the World Wide Web (WWW) from 2001 to 2008, then extract the publication histories from 1991 to 2007 of these authors. Previous works [13, 33] have shown that the subset generated by this strategy is an effective sample.

All the baseline methods and S-rank are trained and tested in following way. In the training stage, we set $T_0 =$ [1991, 1998] and $T_1 =$ [1999, 2000]. This time interval for training works the best for both performance and efficiency. The impact of different training intervals is analyzed in later sections. We select training nodes that have citation relationships with source node in T_1, but never had in T_0. For S-rank, these nodes are further formed into training pairs $\langle u, v\rangle$. According to the definition of training set (Section 2.2), u is more cited by source node than v. The parameters (W and g in this article) are learnt in training stage. To verify the effectiveness of our framework, we extract three groups (group1 to group3) of data for the testing stage, in which T_1' is set to [2000, 2001], [2001, 2002] and [2002,2003] respectively. Similar to training stage, we select testing nodes that have citation relationship with source node in T_1', but never had in T_0'. Based on their features extracted from T_0', these nodes are scored by the learnt model. Then we compare the score with real network. The higher score means more citation relationships with source node in T_1'. According to the network schema in Fig. 2, there are 3 types of nodes and 3 types of edges in DBLP. The number of different nodes and links in subsets is presented in Table 4.

4.2 Experimental setup

Meta path selection This article focus on predicting the *Citation Relation* ($\mathcal{P} = A - P \rightarrow P$) for three compelling reasons: (1) citation prediction is one of the most typical problem in DBLP and has been rarely studied in the literature; (2) citation prediction is challenging both from being a long term prediction and its weak propagation property [34]; (3) the meta path $\mathcal{P}$ with $dom(\mathcal{P}) \neq range(\mathcal{P})$ has seldom been chosen in previous studies. We select the meta path according to two rules: (1) such meta path $\mathcal{P}$ that $dom(\mathcal{P}) = A$ and $range(\mathcal{P}) = P$; (2) as mentioned in Section 3.4, the length of $\mathcal{P}$ is limited to 5. Table 5 summarizes the meta paths $\mathbf{P}$ we selected in experiments.

Measures on meta path Intuitively, different meta paths can capture different semantics in HINs. These rich semantics make meta path a powerful topology in HINs, and numerical measures can be developed on it to capture the semantics. Once a meta path $\mathcal{P}$ is given, some of the meta path-based measures (denoted as $\mathcal{M}_{\mathcal{P}}$) can be summarized as follows.

- **Path Count** [25]. Path Count, denoted as PC, simply counts the number of path instances following the given pattern $\mathcal{P}$ between two objects.
- **Random Walk score** [25]. Random Walk, denoted as RW, describes the ability one object has in visiting another object along a given meta path.

[2] Available at https://aminer.org/dblp_citation.

Table 5 Selected meta paths

Number	Meta path	Length
1	$A - P$	1
2	$A - P \rightarrow P \rightarrow P$	3
3	$A - P - V - P$	3
4	$A - P - A - P$	3
5	$A - P - V - P \rightarrow P$	4
6	$A - P - A - P \rightarrow P$	4
7	$A - P - V - P - A - P$	5
8	$A - P - A - P - V - P$	5
9	$A - P - A - P - A - P$	5

- **Symmetric Random Walk score** [25]. Compared with *RW*, symmetric random walk, denoted as *SRW*,[3] considers the action started from two endpoints of a meta path.

We give an example in Fig. 5 to explain how these scores are calculated. Let $\mathcal{P} = A-P-V-P-A$ represent the relation between two authors, then $PC(A,B) = 4$, $RW(A,B) = \frac{PC(A,B)}{PC(A,.)} = \frac{2}{5}$ and $SRW(A,B) = \frac{2}{5} + \frac{4}{9} = \frac{38}{45}$, where A and B denote $Alice$ and Bob, respectively.

The definition of meta path is based on the local paths between two nodes (u, v) in HINs. It is easy for us to calculate the numeric local structural features between u and v according to the measures proposed above, and these features are able to be used in MPR.

Baseline methods We compare our algorithm with two baseline algorithms: Personalized PageRank (*PPR*) and *PathPredict* (mentioned in Section 1). *PPR*, as an unsupervised method, estimates reachability by considering both the information passing along links between objects and the probability the random walker jumps back to source node s. *PathPredict* is a binary classification model to solve relationship prediction. In our experiments, we adopt the LIBLINEAR [6] to implement *PathPredict*. LIBLINEAR is an open source library for large-scale linear classification, which provides easy-to-use command-line tools and library calls for developers.

Evaluation metrics Although S-Rank is a rank-based model and Ma et al. [18] proposed novel evaluation measures, the metrics are not adopted in this article since we do not use ensemble pruning. The proposed method is evaluated by two performance metrics: the Area under the ROC Curve (AUC) [9] and the Precision at Top k ($prec@k$), i.e., how many of top k nodes suggested by our algorithm actually receive relationships from s. AUC can be calculated by (14):

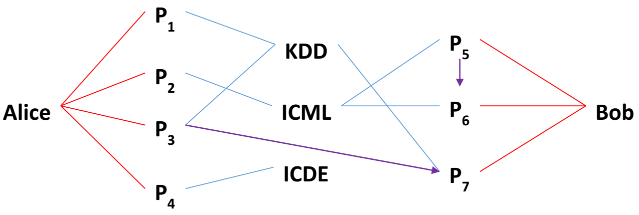


Fig. 5 An example of DBLP for score calculation

$$AUC = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1}, \tag{14}$$

where n_0 and n_1 are the numbers of positive and negative examples. Particularly, in this article, positive examples are the papers cited by source authors in T_1', while negative ones are the papers that have never been cited. $S_0 = \sum r_i$, where r_i is the rank of i-th positive example that ranked by score reversely. It is appropriate to choose these two measures since our dataset is *class imbalanced* and our experiments aim at recommending the highly related papers to an author.

Feature extraction There are three kinds of edges in the dataset: $A - P$, $V - P$ and $P \rightarrow P$. For each edge in the network, we extract features according to its edge type. *year* is extracted as a feature for each type of edge. We also calculate the *cosine similarity* and *jaccard similarity* of the titles of two cited papers for edge $P \rightarrow P$. The *author order* is taken as a feature for edge $A - P$. The extracted features for each type of edge are presented as follows.

- $V - P$: $\frac{1}{|y_p - Y| + 1}$, where y_p denotes the publication year and Y denotes the starting year of dataset.
- $A - P$: $\frac{1}{order}$ and $\frac{1}{|y_p - y_a| + 1}$, where $order$ denotes the author order and y_a represents the debut year of one author.
- $P \rightarrow P$: cosine similarity and jaccard similarity between titles; $\frac{1}{|y_1 - y_2| + 1}$ where y_1 and y_2 denote the publication year of the two papers.

Choice of function and parameters setting There are many implementations for functions mentioned in Section 3.3. We complete our experiments by choosing the following functions according to the empirical performance.

- Exponential capacity for c_{uv}:

  $\pi_w(f) = \exp(f \cdot w)$.
- Wilcoxon-Mann-Whitney (WMW) loss [32]:

  $$h(x) = \frac{1}{1 + \exp(-x/b)}.$$

To determine the function of $SRank$ and the value of β, we conduct experiments and the results are shown in Fig. 6.

[3] Remind that SRW here is distinct from SRW method mentioned in Table 1.

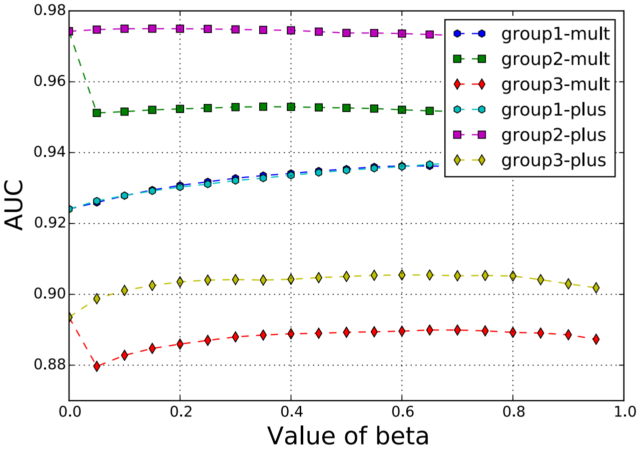


Fig. 6 Performance of *SRank* using different β on three groups of data

mult and *plus* represent (12) and (13), respectively. By analysing the results, we can learn that *plus* performs better than *mult*, and different β indeed affects the results. To achieve stable and overall better performance, we set β as a random float near 0.6.

Other parameters for our algorithm are set as follows: (1) λ is set to 1 since overfitting is not an issue in the experiments; (2) restart parameter α is set to 0.1 for S-Rank (We will discuss in Section 4.4.5). To ensure the fairness of the experiments, restart parameter in *PPR* is set to the same value; (3) step of gradient descent is initially set to 0.02 and decreases by a damping factor of 0.96; (4) convergence condition ϵ is setted to 10^{-9}.

4.3 Experimental results and analysis

The experimental results on three groups of data are shown in Table 6, where the bold numbers indicate the best performance. Different $\mathcal{M}_{\mathcal{P}}$ is utilized to implement *PathPredict*, which are denoted as *PP-PC*,*PP-RW*,*PP-SRW* and *PP-Hybrid* respectively. Note that *Hybrid* in *PP-Hybrid* is the sum of the former three ones. To evaluate the performance, we adopted three kinds of metrics ($prec@10$, $prec@40$ and AUC). The results reported in Table 6 are the average of multiple experiments which select different author nodes as the source node.

First, S-Rank and *PPR* have the absolute advantage over *PathPredict* on AUC. The reason is that *PathPredict*, as a classification method, focuses on finding similarity or relevancy between two authors, which may lose superiority when $dom(\mathcal{P})$ differs from $range(\mathcal{P})$. This also indicates that the ranking manner represented by PageRank works better than the classification manner in citation prediction tasks. Both *PPR* and S-Rank can achieve high AUC because they both use PageRank strategy. However, S-Rank receives even higher performance than *PPR* since it combine Attr-info and Local-info (namely $\mathcal{M}_{\mathcal{P}}$) together.

Next, S-Rank is also more effective from the perspective of $prec@k$. $prec@10$ is meaningful in the situation of recommendation, which mainly concentrates on predicting the front dozens of results. In general, although *PP-Hybrid* outperforms *PPR* on $prec@10$, it performs slightly worse on $prec@40$ and much worse on AUC. This means *PathPredict* has more positive examples in the top of results but cannot find out enough positive examples from overall candidate nodes set. The reason is that *PathPredict* only makes use of Local-info, and hence can only find out relatively closer neighbours from the candidate nodes. On the other hand, *PPR* only utilizes Global-info, which makes it good at overall performance but cannot make prediction precisely on top k. It can also be observed that S-Rank dominates the results of $prec@40$ (improving at most 44.52% against *PPR* and 74.64% against *PathPredict*) and is competitive on $prec@10$ with *PathPredict*. Thus, there is reason to believe that this improvement comes from the combination of three kinds of information in HINs.

Experimental results indicate that the three kinds of information have their own advantages over the relationship prediction task. Therefore, it is essential to combine them together. S-Rank can effectively complete the mission and significantly outperform the baseline methods on three metrics. Besides, traditional relationship prediction only focuses on evaluating the similarity of the the same type of node, such as predicting friendship between two people in a social network or predicting co-author relationships in

Table 6 Performance comparison between s-rank and baseline methods

Methods	group1			group2			group3		
	$prec@10$	$prec@40$	AUC	$prec@10$	$prec@40$	AUC	$prec@10$	$prec@40$	AUC
PPR	4.49	9.74	0.92	7.47	15.61	0.97	2.27	13.40	0.89
PP-PC	3.94	4.62	0.20	8.10	21.34	0.53	6.34	9.81	0.34
PP-RW	0.29	1.20	0.47	2.47	3.67	0.39	3.91	4.72	0.32
PP-SRW	0.83	1.14	0.13	3.80	4.96	0.22	4.36	6.27	0.34
PP-Hybrid	**5.31**	7.13	0.25	8.35	20.41	0.51	**8.41**	10.51	0.37
S-Rank	5.28	**12.43**	**0.93**	**8.91**	**22.56**	**0.97**	5.84	**17.91**	**0.90**

heterogeneous bibliographic networks. In this article, the proposed S-Rank framework studies the citation relationship as an example of relationships between different types of node. Therefore, S-Rank framework is suitable for more various applications in social networks, such as predicting which club will a person join in, which kind of movies will he be interested in and what kind of activities will he attend.

4.4 Discussions

4.4.1 Impact of different meta path-based measures

In the third phase of S-Rank, we combine two ranking results together (namely $Rank_1$ and $Rank_2$). Since $Rank_2$ is related to $\mathcal{M}_\mathcal{P}$, we compare implementations by using three kinds of $\mathcal{M}_\mathcal{P}$ (mentioned in Section 4.2) as well as their hybrid version. They are denoted as *S-PC*, *S-RW*, *S-SRW* and *S-Hybrid* respectively. We conduct experiments to investigate which implementation of $Rank_2$ can achieve the best performance when cooperating with $Rank_1$. From the results shown in Fig. 7, we can learn that there is little difference among the performances of *S-PC*, *S-RW* and *S-Hybrid*, and they overall outperform *S-SRW*. In our experiments, we chose *S-RW* to implement S-Rank. Through this study as well as the experiments later, it is clear that different $\mathcal{M}_\mathcal{P}$ has different expressiveness under different situations. This issue remains to be explored for future work.

4.4.2 Weights of different meta paths

As we stated in Section 2.1, different meta paths can capture different semantics in HINs. Intuitively, the weight associated with each meta path is also different. S-Rank can learn the weights through MPR. From the results shown in Fig. 8, we can observe that the weights for three kinds of measures are unified to some extent. They all assign higher value to meta path $A-P$, $A-P-A-P$ and $A-P-A-P-V-P$ but lower value to $A-P-V-P-A-P$. Considering the semantics associated with these meta paths, the results are consistent with our knowledge in real life. $A-P$ and $A-P-A-P$ represent an author citing papers that he or his coauthors wrote before, and this phenomenon is of common occurrence in academic world; $A-P-A-P-V-P$ represents an author citing papers that have high relevance with his research interests; while the paper reached through $A-P-V-P-A-P$ is too far from the source author since the number of papers in one venue is too large and the other authors have generally extensive research interests.

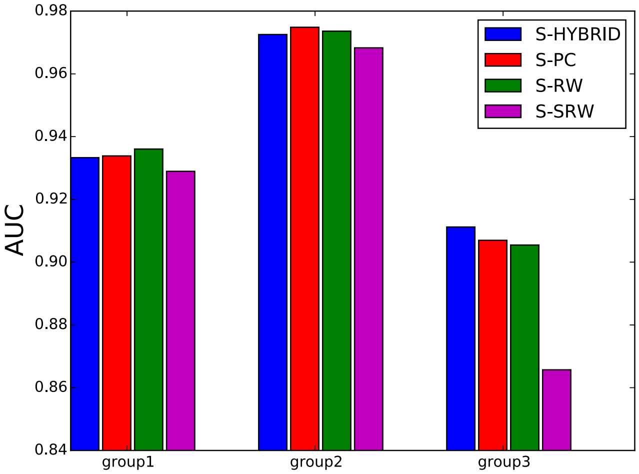


Fig. 7 Performace of S-Rank applying different $\mathcal{M}_\mathcal{P}$

4.4.3 Accuracy for imbalanced data

Continuing discussing the poor performance of *PathPredict* on AUC (Table 6), the essence to this phenomenon is that the dataset for citation prediction is extremely *class imbalanced*. In the experiments, the examples with positive label (papers that the source authors will cite in the future) only account for 1–2% of the training examples (all candidate papers). Thus, the traditional classifier works poorly because it is more difficult to infer reliable patterns with fewer examples of one class [19]. From the results shown in Fig. 9, we can see that all the methods obtain a high accuracy (around 98%), but S-Rank achieves even higher recall than *PathPredict*. This is because S-Rank utilizes training pairs combining both positive examples and negative examples, and hence S-Rank can effectively avoid imbalanced problem.

4.4.4 Impact of training intervals

Since the training costs of S-Rank framework grow rapidly with the increasing of the length of time intervals, we need to consider both performance and efficiency. The training time interval that has a good trade-off between

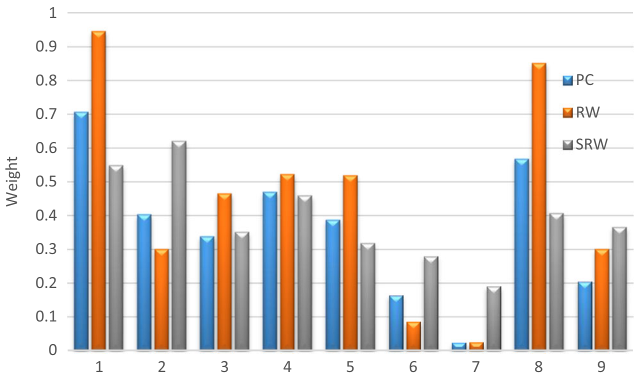


Fig. 8 Weights of different meta paths for PC, RW and SRW. The number in the *horizontal axis* represents the meta paths in Table 5

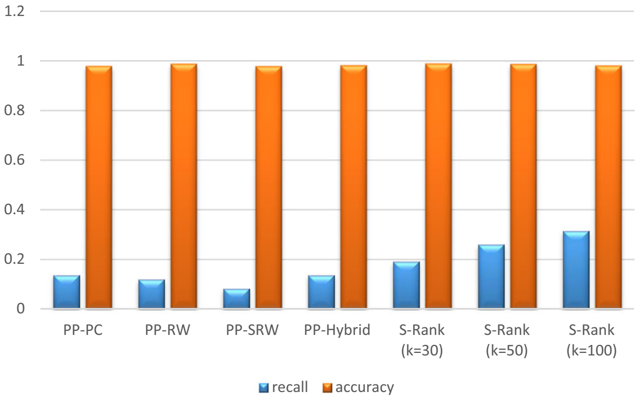


Fig. 9 Recall and accuracy of *PathPredict* and S-Rank

performance and length is required for this framework. To obtain the suitable interval, we apply different training time intervals in the training phase and test the performance on future time intervals. Notice that we set T_1' much longer than we did in Section 4.2, because in this section we mainly focus on how the training time influences the performance of prediction. Three test groups, denoted by $Group1'$, $Group2'$ and $Group3'$, are generated for testing in future time intervals of [2000, 2001], [2003, 2004] and [2005, 2006], respectively. From the results shown in Table 7, we can draw a conclusions that S-Rank performs better with the growth of training time and becomes stabled in training intervals [1992, 1998] and [1991, 1998]. In this article, [1991, 1998] is chosen as T_0, because it achieves slightly better average $prec@10$ and AUC than [1992, 1998].

Figure 10 gives an understandable illustration how the AUC changes in different training time intervals. The blue bar denotes the *best group* that achieves the highest AUC, and the gray bars denote the *second best groups* that the AUC differences are less than 0.1 comparing to the best group. When the training time is more than 7 years ([1992, 1998] and [1991, 1998]), we can learn that S-Rank achieves more balanced performance on all the three groups.

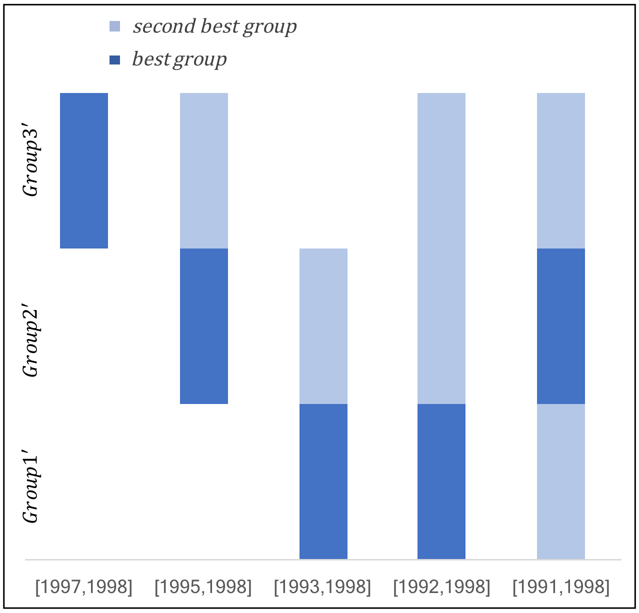


Fig. 10 AUC changes in different training time intervals. *Horizontal axis* represents different training time intervals

4.4.5 Impact of restart parameter

The impact of restart parameter α in the training stage of S-Rank is illustrated in Fig. 11, from which we can see that S-Rank converges in 22 iterations and obtains overall fewer iterations and minimum objective function value when $\alpha = 0.1$. In spite of relatively fewer iterations, unfortunately, the computational expense of each iteration is high (this situation only happens to SPR, while the iteration speed of MPR is fairly fast). In our experiments, each iteration costs nearly 20 minutes. Another limitation of S-Rank is that the optimization for both phases are non-convergence, which means that the convergency value is not a globally optimal solution. Therefore, we resolve this problem by using different starting points to find an optimal solution (we run the first and the second phase 30 times and 110 times, respectively). Since the prediction of S-Rank is as fast as *PPR* because it only needs to weight each edge in HINs, the testing phase can be processed online after the off-line training

Table 7 Performance of s-rank utilizing different training intervals

T_0	$Group1'$		$Group2'$		$Group3'$		Average	
	$prec@10$	AUC	$prec@10$	AUC	$prec@10$	AUC	$prec@10$	AUC
[1997,1998]	5.50	0.5372	5.43	0.6122	6.67	0.7222	5.87	0.6239
[1995,1998]	6.24	0.7396	8.01	0.8347	7.50	0.8437	7.25	0.8060
[1993,1998]	8.00	0.9231	8.58	0.8641	7.93	0.7851	8.17	0.8574
[1992,1998]	9.21	0.9290	8.00	0.8958	9.00	0.9195	8.74	0.9148
[1991,1998]	9.50	0.9159	9.27	0.9378	9.15	0.9279	9.30	0.9272

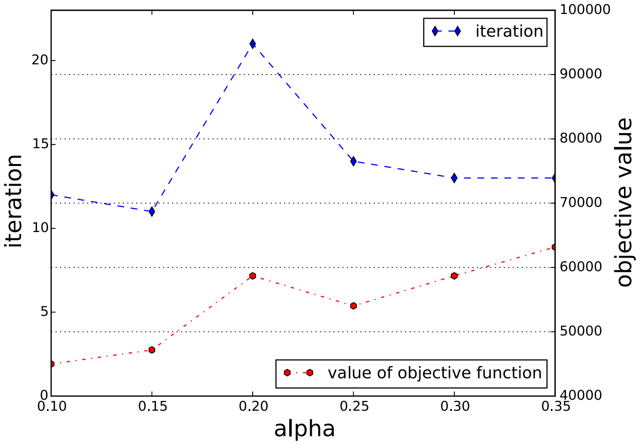


Fig. 11 Impact of the restart parameter α for iteration and the objective function value

has been finished. Due to the excellent prediction speed and performance, we can conclude that S-Rank is a valuable and practical framework for relationship prediction over HINs.

5 Related work

Heterogeneous information network(HIN) is a novel network type, which aims to integrate valuable semantic and structural information into it. HINs have many advantages than traditional homogeneous networks, since HINs are more capable to represent real-word networks. Strategies on HINs can be also applied to many areas.

HIN shows its capabilities in the area of text processing. Shen et al. [22] propose a probabilistic model to solve linking named entities in HINs. Wang et al. [31] study the traditional text classification problem with HINs. They represent the texts in a HIN manner and utilize a meta path-based approach to link texts. They successfully develop some HIN-kernels guiding the classifier to utilize the hidden semantics in the heterogeneous representation of texts.

HIN is also powerful with regard to recommendation. Yu et al. [35] aim to accomplish the personalized recommendation by incorporating the implicit feedback, which is denoted by different relationships in HINs. Shi et al. [24] notice that previous studies about HINs did not utilize the attributes existent in links. They propose a Weighted Heterogeneous Information Networks upon the original definition to conduct the personalized recommendation. This part of their work is very similar to our work. However, the attributes in our work are organized as an attribute vector, which means the attribute in node can also be incorporated into the vector and thus has different meanings in different type of links.

The most traditional task in HINs is link prediction. Link prediction in HINs has been extensively studied these years. Since meta path has been proposed by Sun et al. [26], several meta path-based approaches and their modified versions are applied to link predictions. PathPredict [25] is a significant model that utilizes meta path, which tries to model the relationship prediction problem as a binary classification. It proposes some measures as features based on meta path and chooses the Logistic Regression as the classifier. Although PathPredict effectively take advantage of local topological features, namely Local-info, it ignores the rich Attr-info and global topological features encoded in the networks.

Some approaches can capture Global-info. RW_ALL aims at tackling link prediction through modifying HIN into a new network that given meta paths are represented by direct links between start node and end node [13]. Then random walk is applied to capture the Global-info in the modified network. By modifying an existing HIN into another network, the important relations (meta paths) in HINs can be highlighted. Thus, RW_ALL is also capable of capturing Local-info. However, the modification process is extremely difficult since different HINs have different meta structures, which limits the applicability of RW_ALL.

Various approaches have verified that meta path is powerful for mining structural information, but it is defective for mining Attr-info. Therefore, other methods which can capture Attr-info should be involved. Bucket [34] has studied the citation relationship prediction problem. It proposes the Discriminative Term Bucketing (DTM) to capture document and topic similarities that maintain possible citation relations, and then combines the meta path-based features to predict the citation probability. Bucket takes advantage of both the Local-info and Attr-info, but it is not a universal approach since the data structure of DTM is strongly limited to the citation prediction problem.

Except for the above problem of meta path, the selection of meta paths may be ambiguous. Cao et al. [4] state that contemporary link prediction treats the schema of HINs too simple, which is bipartite or star-schema. Naturally, meta paths in these HINs are predefined. They proposed a novel Link Prediction with automatic meta Paths method (LiPaP). LiPaP designs an algorithm called Automatic Meta Path Generate (AMPG), which is a greedy algorithm to select meta paths according to their priorities. The automatic meta path generation is novel, however, the pattern trained by automatically selected meta paths is not specified, which means that it may not be useful to predict some relationships whose semantics are denoted by other meta paths.

Some approaches do not utilize meta path. Deng et al. [5] proposed PAV (Paper, Author, Venue) model to rank the objects in HINs, which assigns weights to the edges and then applies a random walk model to rank the objects. The weight is computed according to the latent attribute information of each edge. PAV model is similar to the SPR model of our proposed S-Rank framework. Both of them can capture Global-info and Attr-info, but SPR is a Page Rank-based strategy. Similar to PAV, SRW (Supervised Random Walk) [1] involves a supervised ranking method for link prediction on social networks, which can effectively combine the Global-info and Attr-info together. Gao et al. proposed a semi-supervised learning method called SSP (Semi-Supervised PageRank) [7] to learn Global-info and Attr-info by applying a Markov random walk on the graph. However, SRW and SSP are studied under the context of homogeneous networks, and thus they cannot be directly applied to HINs.

Cao et al. [3] study the collective prediction problem in HINs. This work is strongly related to HINs since it raised two meaningful problems: (1) the existence possibility of links between two different types of objects should not be measured by traditional proximity measures that are defined on one single type of nodes; (2) different types of links are not independent but related with complex dependencies among them. To address these two problems, [3] proposes a relatedness measure and an iterative framework inspired by co-training. Another collective classification problem in HINs can refer to [12], in which Kong et al. proposed a novel method that can exploit a large number of different types of dependencies among objects simultaneously.

Recently, Link prediction has been extended to other researches. Baoxu Shi and Tim Weninger [23] introduce the link prediction in HINs to a traditional problem, namely Fact Checking. They propose a new model of the top discriminative meta paths, which is able to understand the meaning of some statement and accurately determine its veracity. Ma and Dai [17] study time series prediction on financial datasets and propose PS-ELMs (Pruned Stacking Extreme Learning Machine) algorithm to tackle this issue.

A similar problem to link prediction, namely *Inferring Social Ties*, is also studied on HINs. Tang et al. [28] study the problem of inferring social ties over multiple heterogeneous networks. The idea of applying social psychological theories to mine the similar patterns existed in different networks is novel. Wenbin Tang el al. [30] aim to tackle this problem in the situation of large scale network. Moreover, He et al. [10] extend the similarity measurement by absorbing transitive similarity and temporal dynamics. Ma et al. [16] developed a new prediction task in HINs, namely Neighbor Distribution Prediction (NDP).

6 Conclusions and future work

In this article, we studied the relationship prediction problem in HINs. We first analyzed the impact of three categories of information, namely Local-info, Global-info and Attr-info, behind the creation of relationships. Then, we proposed a novel supervised three-phase framework, called S-Rank, to utilize all the useful information and predict the emergence of relationships in the future. To the best of our knowledge, our work is the first to completely combine Global-info, Local-info and Attr-info together. Experimental results indicate that the combination of three kinds of information can significantly improve the performance compared to with the baseline methods.

For future work, we are interested in two main aspects. Firstly, we aim to figure out a solution with good convergence property. Secondly, the potential advantages of combining three kinds of information remain to be explored by other technologies. For example, more complex Rank Aggregation methods and co-training [2] techniques can be used in our framework.

Acknowledgments This work was partially supported by National High Technology Research and Development Program (863 Program) of China (No. 2015AA015403) and National Science Foundation of China (No. 61632019).

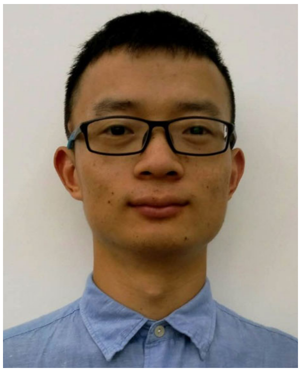
References

1. Backstrom L, Leskovec J (2011) Supervised random walks: predicting and recommending links in social networks. In: The fourth ACM international conference on Web search and data mining. ACM, pp 635–644
2. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: The eleventh annual conference on computational learning theory. ACM, pp 92–100
3. Cao B, Kong X, Yu PS (2014) Collective prediction of multiple types of links in heterogeneous information networks. In: ICDM, pp 50–59
4. Cao X, Zheng Y, Shi C, Li J, Wu B (2016) Link prediction in schema-rich heterogeneous information network. In: Advances in knowledge discovery and data mining - 20th Pacific-Asia conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part I, pp 449–460
5. Deng ZH, Lai BY, Wang ZH, Fang GD (2012) Pav: a novel model for ranking heterogeneous objects in bibliographic information networks. Expert Syst Appl 39(10):9788–9796
6. Fan R, Chang K, Hsieh C, Wang X, Lin C (2008) LIBLINEAR: a library for large linear classification. J Mach Learn Res 9:1871–1874
7. Gao B, Liu T, Wei W, Wang T, Li H (2011) Semi-supervised ranking on very large graphs with rich metadata. In: SIGKDD, pp 96–104
8. Han J (2012) Mining heterogeneous information networks: the next frontier. In: SIGKDD. ACM, pp 2–3
9. Hand DJ, Till RJ (2001) A simple generalisation of the area under the roc curve for multiple class classification problems, pp 171–186
10. He J, Bailey J, Zhang R (2014) Exploiting transitive similarity and temporal dynamics for similarity search in heterogeneous information networks. In: DASFAA, pp 141–155
11. Kautz H, Selman B, Shah M (1997) Referral web: combining social networks and collaborative filtering. Commun ACM 40(3):63–65
12. Kong X, Yu PS, Ding Y, Wild DJ (2012) Meta path-based collective classification in heterogeneous information networks. In: The

21st ACM international conference on information and knowledge management. ACM, pp 1567–1571
13. Lee JB, Adorna H (2012) Link prediction in a modified heterogeneous bibliographic network. In: ASONAM. IEEE, pp 442–449
14. Liang W, He X, Tang D, Zhang X (2016) S-rank: a supervised ranking framework for relationship prediction in heterogeneous information networks. Lecture notes in computer science, vol 9799. Springer, pp 305–319
15. Liben-Nowell D, Kleinberg JM (2003) The link prediction problem for social networks. In: Proceedings of the 2003 ACM CIKM international conference on information and knowledge management. New Orleans, pp 556–559
16. Ma Y, Yang N, Li C, Zhang L, Yu PS (2015) Predicting neighbor distribution in heterogeneous information networks. In: Proceedings of the 2015 SIAM international conference on data mining. Vancouver, pp 784–791
17. Ma Z, Dai Q (2016) Selected an stacking elms for time series prediction. Neural Process Lett 44:831–856
18. Ma Z, Dai Q, Liu N (2015) Several novel evaluation measures for rank-based ensemble pruning with applications to time series prediction. Expert Syst Appl 42:280–292
19. Menon AK, Elkan C (2011) Link prediction via matrix factorization. In: ECML/PKDD (2), pp 437–452
20. Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web
21. Rajkumar A, Agarwal S (2014) A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In: ICML, pp 118–126
22. Shen W, Han J, Wang J (2014) A probabilistic model for linking named entities in web text with heterogeneous information networks. In: SIGMOD, pp 1199–1210
23. Shi B, Weninger T (2016) Fact checking in heterogeneous information networks. In: Proceedings of the 25th international conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume, pp 101–102
24. Shi C, Zhang Z, Luo P, Yu PS, Yue Y, Wu B (2015) Semantic path based personalized recommendation on weighted heterogeneous information networks. In: Proceedings of the 24th ACM international on conference on information and knowledge management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015, pp 453–462
25. Sun Y, Barber R, Gupta M, Aggarwal CC, Han J (2011) Co-author relationship prediction in heterogeneous bibliographic networks. In: ASONAM. IEEE, pp 121–128
26. Sun Y, Han J, Yan X, Yu PS, Wu T (2011) Pathsim: meta path-based top-k similarity search in heterogeneous information networks. PVLDB 4(11):992–1003
27. Sun Y, Han J, Aggarwal CC, Chawla NV (2012) When will it happen? Relationship prediction in heterogeneous information networks. In: WSDM, pp. 663–672
28. Tang J, Lou T, Kleinberg J (2012) Inferring social ties across heterogenous networks. In: The fifth ACM international conference on Web search and data mining. ACM, pp 743–752
29. Tang J, Zhang J, Yao L, Li J, Zhang L, Su Z (2008) Arnetminer: extraction and mining of academic social networks. In: SIGKDD, pp 990–998
30. Tang W, Zhuang H, Tang J (2011) Learning to infer social ties in large networks. In: Machine learning and knowledge discovery in databases - European conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III, pp 381–397
31. Wang C, Song Y, Li H, Zhang M, Han J (2016) Text classification with heterogeneous information network kernels. In: Proceedings of the thirtieth AAAI conference on artificial intelligence. Phoenix, pp 2130–2136
32. Yan L, Dodier RH, Mozer M, Wolniewicz RH (2003) Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In: ICML, pp 848–855
33. Yin Z, Gupta M, Weninger T, Han J (2010) A unified framework for link recommendation using random walks. In: ASONAM. IEEE, pp 152–159
34. Yu X, Gu Q, Zhou M, Han J (2012) Citation prediction in heterogeneous bibliographic networks. In: SDM. SIAM, pp 1119–1130
35. Yu X, Ren X, Sun Y, Gu Q, Sturt B, Khandelwal U, Norick B, Han J (2014) Personalized entity recommendation: a heterogeneous information network approach. In: Seventh ACM international conference on web search and data mining, WSDM 2014. New York, pp 283–292

Wenxin Liang received his B.E. and M.E. degrees from Xi'an Jiaotong University, China in 1998 and 2001, respectively. He received his Ph.D. degree in Computer Science from Tokyo Institute of Technology, Japan in 2006. He was a Postdoc Research Fellow, CREST of Japan Science and Technology Agency (JST) and a Guest Research Associate, GSIC of Tokyo Institute of Technology from Oct. 2006 to Mar. 2009. His main research interests include Data Engineering, Artificial Intelligence, Social Networks, etc. He is currently an associate professor at the School of Software, Dalian University of Technology, China. He is a senior member of China Computer Federation (CCF), and a member of IEEE, ACM, ACM SIGMOD Japan Chapter and Database Society of Japan (DBSJ).

Xiao Li is a Master's student in Computer Science at the School of Software, Dalian University of Technology, China. He received his B.S. degree in Software Engineering from Dalian University of Technology, China in 2016 and started his Master's program since the same year. His research interests include Machine Learning and Data Mining.



Xiaosong He received his B.S. and M.S. degrees in Software Engineering from Dalian University of Technology, China in 2013 and 2016, respectively. He worked at Southwest China Research Institute of Electronic Equipment from 2016 to 2017. He is currently working at A Bit AI Co., Ltd, Beijing, China. His research interests include Machine Learning, Data Mining and Text Matching and Classification.

Xinyue Liu is an associate professor at the School of Software, Dalian University of Technology, China. She got her B.S and M.S degrees from Northeast Normal University, China, in 2003 and 2006, respectively. She received her Ph.D. degree in Computer Science from Dalian University of Technology, China in 2012. Her research interests include Data mining, Machine Learning and Information Retrieval.

Xianchao Zhang is a full professor at Dalian University of Technology, China. He got his Bachelor's and Master's degrees in mathematics from National University of Defense Technology, China, in 1994 and 1998, respectively. He received his Ph.D. degree in Computer Science from the University of Science and Technology of China in 2000. From 2000 to 2003, he worked as a research and development manager in some international companies. He joined Dalian University of Technology in 2003. His research interests include Design and Analysis of Algorithms, Machine Learning, Data Mining, and Information Retrieval.