Ranking-based Clustering on General Heterogeneous Information Networks by Network Projection

Chuan Shi
Beijing University of Posts and Telecommunications
Beijing China
shichuan@bupt.edu.cn

Ran Wang
Beijing University of Posts and Telecommunications
Beijing China
wangran51@126.com

Yitong Li
Beijing University of Posts and Telecommunications
Beijing China
yitongglee@gmail.com

Philip S. Yu
University of Illinois at Chicago
IL USA
psyu@uic.edu

Bin Wu
Beijing University of Posts and Telecommunications
Beijing China
wubin@bupt.edu.cn

ABSTRACT

Recently there is an increasing attention in heterogeneous information network analysis, which models networked data as networks including different types of objects and relations. Many data mining tasks have been exploited in heterogeneous networks, among which clustering and ranking are two basic tasks. These two tasks are usually done separately, whereas recent researches show that they can mutually enhance each other. Unfortunately, these works are limited to heterogeneous networks with special structures (e.g. *bipartite or star-schema network*). However, real data are more complex and irregular, so it is desirable to design a general method to manage objects and relations in heterogeneous networks with arbitrary schema. In this paper, we study the ranking-based clustering problem in a general heterogeneous information network and propose a novel solution HeProjI. HeProjI projects a general heterogeneous network into a sequence of sub-networks and an information transfer mechanism is designed to keep the consistency among sub-networks. For each sub-network, a path-based random walk model is built to estimate the reachable probability of objects which can be used for clustering and ranking analysis. Iteratively analyzing each sub-network leads to effective ranking-based clustering. Extensive experiments on three real datasets illustrate that HeProjI can achieve better clustering and ranking performances compared to other well-established algorithms.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications-Data Mining

CIKM 2014 Shanghai China

General Terms

Theory

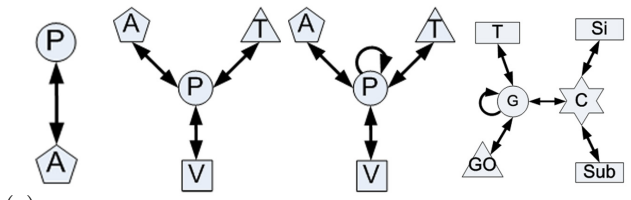
Keywords

heterogeneous information network, clustering, ranking

1. INTRODUCTION

Recently there is a surge of research on Heterogeneous Information Network (HIN) in which objects are of different types and links among objects represent different relations. It is clear that this kind of networks is ubiquitous and forms a critical component of modern information infrastructure [5]. For example, in the case of bibliographic network (e.g., network schema of DBLP dataset shown in Fig. 1(c)), the object types include authors, papers, venues; and links between objects correspond to different relations, such as write relation between authors and papers.

Many data mining tasks have been exploited in HIN, such as clustering [18], classification [7], and ranking [16]. The link-based clustering attracts more and more attention, which usually groups objects that are densely interconnected but sparely connected with the rest of the network [10]. Also with the booming of search engine, object ranking [2, 6] becomes an important data mining task, which evaluates the importance of objects. Conventionally, clustering and ranking are two independent tasks and they are usually used separately. However, recent researches show that clustering and ranking can mutually promote each other and their combination makes more sense in many applications [17, 19]. If we know the important objects in a cluster, we can understand this cluster better; and the ranking in a cluster provides more subtle and meaningful information for clustering. Although it is a promising way to do clustering and ranking together, previous approaches are confined to a simple HIN with special structure. For example, Sun et al. validated the mutual improvement of clustering and ranking in bipartite network [17] (an example shown in Fig. 1(a)) and star-schema network [19] (an example shown in Fig. 1(b)). Shi et al. [20] integrated clustering and ranking in the hybrid network including heterogeneous and homogeneous relations. However, the data in real applications are



(a) bipartite (b) star-schema (c) DBLP (d) SLAP

Figure 1: Examples of heterogeneous information networks. The letters are the abbreviation of different types of objects (e.g., P: paper, A: author). The details can be seen in Sec. 5.1.

usually more complex and irregular, which are beyond the widely used bipartite or star-schema network. For example, the bibliographic data (see an example in Fig. 1(c)) includes not only heterogeneous relations but also homogeneous relations (e.g., self loop on P); the bioinformatics data [3] (see an example in Fig. 1(d)) have more complex structure, which includes multiple hub objects (e.g., C and G). So it is desirable to design effective ranking based clustering algorithm for these complex and irregular HIN data. Broadly speaking, for HIN with arbitrary schema, we need to design a general solution to manage the objects and their relations, which is the basic for mining useful patterns on it.

Obviously, it is more practical and useful to determine the underlying clusters and ranks on a general heterogeneous information network, but they are seldom exploited until now. When we integrate ranking and clustering on a HIN with arbitrary schema, it faces the following challenges. 1) A general HIN has more complex structure. For a simple HIN with a bipartite or star-schema structure, it is relatively easy to manage heterogeneous objects and build models. However, a general HIN may have arbitrary schema, beyond the bipartite or star-schema structure. Although an intuitive way is to decompose it into multiple simpler sub-networks, the issue is how we decompose the HIN without structural information loss and maintain the consistency among the decomposed sub-networks. 2) It is challenging to integrate the clustering and ranking in a complex heterogeneous network. We know that it is still a daunting task to separately do clustering and ranking on a general HIN. Therefore, it is more difficult to design an effective mechanism to combine these two tasks on the HIN.

In this paper, we study the ranking-based clustering problem on a general HIN and propose a novel algorithm **HeProjI** to solve the **He**terogeneous network **Proj**ection and **I**ntegration of clustering and ranking tasks. In order to conveniently manage objects and relations in a HIN with arbitrary schema, we design a network projection method to project the HIN into a sequence of sub-networks without structural information loss, where the sub-network may be a relatively simple bipartite or star-schema network. Moreover, an information transfer mechanism is developed to maintain the consistency across sub-networks. For each sub-network, a path-based random walk method is proposed to generate the reachable probability of objects, which can be effectively used to estimate the cluster membership probability and the importance of objects. Through iteratively analyzing each sub-network, HeProjI can obtain the steady and consistent clustering and ranking results. We perform a number of experiments on three real datasets to validate the effectiveness of HeProjI. The results show that HeProjI not only achieves better clustering and ranking accuracy compared to well-established algorithms, but also effectively handles complex HIN which cannot be handled by previous methods.

2. RELATED WORK

Many data mining tasks have been exploited in heterogeneous information networks. According to the organization methods of objects, contemporary work can be roughly classified the following three types. 1) HIN is decomposed to multiple homogeneous networks. Most network analysis focus on homogeneous networks [10, 12]. However, the information loss from the decomposition operation may induce the inconsistency and unbalance among networks. 2) Bipartite graph is widely used to organize two types of objects and the relations among them, such as conference-author [17] and author-document [21]. As an extended version, the K-partite graphs [8] are able to represent the multiple types of objects. However, they both ignore the homogeneous relation among objects of same type. 3) HIN is usually organized as star-schema network [13, 19, 18] where a target type is central node and connected by several attribute types. Many data with the target-attribute relations can be represented with this schema, such as bibliographic data [19] and movie data [13]. However, more real networked data may have multiple hub types and homogeneous relations, which cannot be represented with the star-schema network.

Recently, the clustering on heterogeneous network attracts much attentions. Some of spectral clustering-based methods confine to bi-type relational data [17]. The spectral clustering methods are also developed for a general relational data which are modeled as K-partite graphs [8]. Sun et al. [18] presented a semi-supervised clustering algorithm to generate different cluster results with path selection according to user guidance. The ranking problem is also an important task in data mining, which evaluates the importance of objects based on some ranking functions. Conventional ranking tasks are set in homogeneous networks, such as PageRank [2] and SimRank [6]. Recently, more and more researches began to pay attention to the rank problem in heterogeneous networks. For example, Sun et al. [16] proposed PathSim to evaluate the similarity of same-typed object pairs in HIN. Contemporary clustering and ranking are usually done independently.

In recent years, ranking-based clustering algorithms illustrate that ranking and clustering can mutually promote each other. RankClus [17] is proposed to generate clusters integrated with ranking, and theoretical and experimental analysis show that the quality of clustering and ranking are mutually enhanced. Furthermore, Sun et al. [19] studied the clustering of multi-typed heterogeneous networks with a star network schema and proposed NetClus to generate high-quality net-clusters. However, these two algorithms are confined to the specified network schema, i.e., RankClus and NetClus only for the bipartite and star-schema networks, respectively. Recently, Shi et al. [20] proposed the ComClus to promote clustering and ranking performance on a kind of hybrid network including the heterogeneous and homogeneous relations. In fact, ComClus is also confined to star schema network with self loop. So these methods cannot be directly applied to a general HIN with arbitrary schema.

3. PROBLEM FORMULATION

In this section, we give the problem definition and some important concepts used in this paper.

DEFINITION 1. ***General heterogeneous information network.*** *Given a schema $\mathcal{A} = (\mathcal{T}, \mathcal{R})$ which consists of a set of entities type $\mathcal{T} = \{T\}$ and a set of relations $\mathcal{R} = \{R\}$, a general information network is defined as a graph $\mathcal{G} = (X,E)$ with an object type mapping function $\tau : X \to \mathcal{T}$ and link type mapping function $\psi : E \to \mathcal{R}$. Each object $|\mathcal{T}| > 1$ or the types of relations $|\mathcal{R}| > 1$, the network is called **heterogeneous information network**; otherwise, it is a **homogeneous information network**.*

Fig. 1 shows the schema of several HIN examples. The bipartite network in Fig. 1(a) only includes two types of objects, and the widely used star-schema network [13, 19, 18] in Fig. 1(b) organizes objects in HIN with one target type and several attribute types. However, a general heterogeneous information network may be more complex and irregular. It may not only include homogeneous or heterogeneous relations, but also include multiple hub objects. Fig. 1(d) shows such a general HIN example. The object G has heterogeneous relations (e.g., $G \to GO$ and $G \to C$) as well as homogeneous relations (e.g., $G \to G$). Moreover, the network is beyond the star-schema because of multiple hub objects (e.g., G and C). It is clear that bipartite graph and star-schema network is the special case of a general HIN.

For a general HIN, it is difficult to manage objects and relations in the network. Although we can project it into several homogeneous networks through assigning meta paths as reference [4] did, it will loss much information among different-typed objects. We know that, as the special case of HIN, the bipartite and star-schema networks are relatively easy to manage objects and relations in the network. So a basic idea of handling a general HIN is to decompose it into simpler networks. Following this idea, we design a novel HIN projection method. Specifically, we can select one type (called pivotal type) and its connected other types (called supportive type). These types and their relations constitute the schema of a projected sub-network of original HIN. Formally, it can be defined as follows:

DEFINITION 2. ***Projected sub-network.*** *For a HIN with schema $\mathcal{A} = (\mathcal{T}, \mathcal{R})$, its projected sub-network has the schema $\mathcal{A}' = (\mathcal{T}', \mathcal{R}')$ where $\mathcal{T}' \subset \mathcal{T}$,$\mathcal{R}' \subset \mathcal{R}$, $\mathcal{T}'$ includes one **pivotal type** (denoted as P) and other types connected with P (called **supportive type**, denoted as $\mathcal{S} = \{S\}$). $\mathcal{R}'$ includes the heterogeneous relations between P and S and homogeneous relations among P (if existing).*

A projected sub-network can be denoted as $P - \mathcal{S}$. The $X^{(P)}$ is the object set of pivotal type, and $X^{(S)}$ represents the object set of supportive type S. For convenience, the projected sub-network is also called sub-network which can be represented with its pivotal type P. For example, Fig. 2(c) shows the projected sub-network $G - \{C, T, GO\}$ with type G object (the one in red) as the pivotal type, while types C, T and GO are the supportive types as they are object types connected to object type G. Similarly, Fig. 2(b) and (d) show the projected sub-networks with pivotal type objects GO and C, respectively.

It is clear that a HIN can be projected into a sequence of sub-networks through selecting different pivotal types. So we define the HIN projection concept as follows.

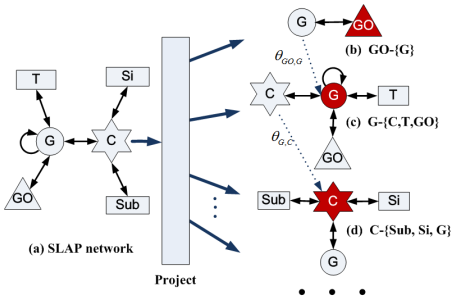


Figure 2: An example of HIN projection. The pivotal type is marked with red color. The dot line represents the information transfer among sub-networks.

DEFINITION 3. ***HIN projection.*** *A HIN with t types of objects can be projected into an ordered set of t projected sub-networks by successively selecting one of the t types as pivotal type.*

Fig. 2 shows a projection example of SLAP network, a bioinformatics dataset (details in Section 5.1). Through successively selecting the 6 object types (GO, G, C and so on) as pivotal type, the SLAP network is projected into a sequence of 6 sub-networks. It is clear that the HIN projection has the following properties.

Property 1. HIN projection is a structure-information lossless network decomposition.

According to Def. 3, all objects and relations in original HIN are in the projected sub-networks. That is to say, the HIN can be reconstructed from the set of projected sub-networks.

Property 2. Each projected sub-network in HIN projection should be a bipartite graph or a star-schema network (with self loop).

According to Def. 2, if there are two types of objects in the sub-network, it is a bipartite graph; otherwise it is a star-schema network. Note that, different from the conventional bipartite and star-schema network, the pivotal type in sub-networks may include the homogenous relation (i.e., self loop).

Property 3. HIN projection is not unique for a general HIN.

A HIN has different projection sequences through selecting different orders of pivotal types. For example, the SLAP network in Fig. 2 has the projection sequences: $GO - G - C - Si - Sub - T$, $T - G - GO - C - Si - Sub$ and so on. In fact, a HIN with t types of objects has the $t!$ projection sequences in all.

Assume that J represents a type in type set $\{T\}$. The object set can be denoted as $X = \{X^{(J)}\}$, and $X^{(J)} = \{X_p^{(J)}\}$ where $X_p^{(J)}$ is the object $p \in X^{(J)}$ (i.e., $\tau(p) = J$). The relations among objects include two types (homogeneous and heterogeneous relations), which can be represented by the two types of matrices ***homogeneous*** and ***heterogeneous relation matrices***, respectively. If type J has homogeneous relation (e.g., the self loop on P in Fig. 1(c)), the homogenous relation matrices can be written as $H^{(J)}$, where $H_{pq}^{(J)}$ denotes the relation between $X_p^{(J)}$ and $X_q^{(J)}$. If two types (I and J) have heterogeneous relation (e.g., $P - A$ in Fig. 1(c)), the heterogeneous relation matrices can be written as $H^{(I,J)}$,

where $H_{pq}^{(I,J)}$ denotes the relation between $X_p^{(I)}$ and $X_q^{(J)}$. Correspondingly, we have **homogeneous transition matrix** $M^{(J)}$ and **heterogeneous transition matrix** $M^{(I,J)}$. It is clear that the transition matrix $M^{(I,J)}$ can be derived from the relation matrix $H^{(I,J)}$ by $M^{(I,J)}{=}D^{(I,J)^{-1}}H^{(I,J)}$, where $D^{(I,J)}$ is the diagonal matrix with the diagonal value equaling to the corresponding row sum of $H^{(I,J)}$. Similarly, $M^{(J)}{=}D^{(J)^{-1}}H^{(J)}$. Taking Fig. 1(c) as example, $M^{(P)}$ is the transition probability matrix of the citation relation $H^{(P)}$, and $M^{(A,P)}$ is the transition probability matrix of the $A-P$ relation $H^{(A,P)}$. For given network structure, we can derive the homogeneous and heterogeneous transition matrix. In the following section, we consider that the transition matrix are known.

Different from conventional clustering in homogeneous networks, cluster in HIN should include different types of objects, where these objects share the same semantic meaning. For example, in bibliographic data, a cluster about data mining area includes venues, authors, and papers in this field. For each type objects $X^{(J)}$, we define the **membership matrix** $B^{(J|C_k)} \in [0,1]^{|X^{(J)}|\times|X^{(J)}|}$, which is a diagonal matrix whose diagonal value represent the membership probability of $X_p^{(J)}$ belonging to the cluster C_k. Note that the sum of membership probability of $X_p^{(J)}$ in K clusters is 1 (i.e., $\sum_{k=1}^{K} B_{pp}^{(J|C_k)}{=}1$). Now, we can formulate the problem of clustering on a general HIN as follows. Given a heterogeneous network $G{=}(X,E)$ and the semantic cluster number K, our goal is to find a clusters set $\{C_k\}_{k=1}^{K}$, where C_k is defined as $C_k = \{\{B^{(J|C_k)}\}_{J\in\{T\}}\}$. In this way, it is a soft clustering. That is, an object p in $X^{(J)}$ can belong to several clusters, and it is in a cluster C_k with the probability $B_{pp}^{(J|C_k)}$. Moreover, a cluster C_k can contain all kinds of objects.

4. THE HEPROJI ALGORITHM

Through the HIN projection, it will become much easier to analyze the HIN through handling a set of simple projected sub-networks, since these sub-networks are bipartite or star schema networks. However, it may result in a troublesome business: how to maintain the consistency among different sub-networks. To solve it, we design an information transfer mechanism which inherits a portion of information from other sub-networks to current one. In order to integrate the clustering and ranking in a uniform framework, a model is required to flexibly support these two tasks. Following this idea, we build a probabilistic model to estimate the probability of supportive and pivotal objects in each sub-network. Moreover, the probability of objects can effectively infer the clustering information and represent the importance of objects.

4.1 Framework of HeProjI Algorithm

Specifically, we first project the original HIN into a sequence of sub-networks, and then randomly assign the pivotal objects of the first sub-network into K clusters (i.e., initialize $\{C_k\}_{k=1}^{K}$). For each sub-network, a path-based random walk method is proposed to estimate the reachable probability of supportive objects in each cluster C_k and then a generative model is used to obtain the probability of pivotal objects. After that, an EM algorithm is employed to estimate the posterior probability of objects (i.e., the clustering information $\{C_k\}_{k=1}^{K}$). According to probability of objects, we can also calculate their ranking in each cluster. The above step is repeated until convergence. In the iterative process, the clustering and ranking can mutually promote each other until they reach a steady result. The basic framework of HeProjI is shown in Algorithm 1. In the following sections, we will present these operations in detail.

Algorithm 1 HeProjI: Detecting K clusters on HIN

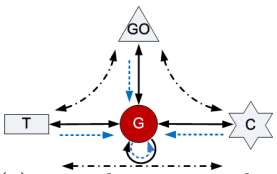
Input:
Cluster number K and transition probability matrix M.
Output:
Membership probability $B^{(J|C_k)}$ of objects on each cluster$\{C_k\}_{k=1}^{K}$
Project the HIN into a sequence of sub-networks
Randomly initialize the membership probability $B^{(J|C_k)}$
repeat
  Select the projected sub-network $(P-\mathcal{S})$ in order
  for cluster $C_k \in C$ **do**
    Establish the probability of supportive objects: $Pr(X^{(S)}|C_k)$
    Generate the probability of pivotal objects: $P(X^{(P)}|C_k)$
    Estimate the posterior probability of objects: $P(C_k|X^{(P)})$, $P(C_k|X^{(S)})$
  end for
  Rank the objects: $Rank(X^{(P)}|C_k)$, $Rank(X^{(S)}|C_k)$
until the membership probability obtains convergence

4.2 Reachable Probability Estimation of Objects

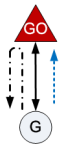
4.2.1 Basic idea

As we have noted that the built probabilistic model can not only support the clustering and ranking tasks but also maintains the consistency among sub-networks. So the design of the model should obey the following two rules. 1) PageRank principle. In order to support the ranking task, the probability of objects should be able to reflect their ranks. In other words, the probability of objects should be positively correlated to the node degree. 2) Consistency principle. In order to maintain the consistency among sub-networks, an effective mechanism should be designed to transfer appropriate information among sub-networks.

For the first rule (i.e., PageRank principle), the random walk is an apparent solution. However, it is traditionally used in homogeneous networks [6, 2]. Although it is also used in bipartite graph [21], it is seldom applied in HIN. Sun et al. [19] employed it to estimate the probability of attribute objects in a star-schema network, while it is confined to two types of objects. Heterogeneous objects and link semantics make it difficult to directly employ random walk in HIN. In a projected sub-network, there are different types of supportive objects and they are connected through pivotal objects. So the random walk among objects should follow the specified paths. That is, the random walkers among supportive objects would need to pass through the pivotal objects. As a consequence, we need to estimate the probability of supportive and pivotal objects separately. The reachable probability of a supportive object can be calculated as the sum of the probability of walkers from other supportive objects walking to it through the pivotal type. The probability of pivotal objects can be generated through its reachable supportive objects. Because the bipartite network only contains one supportive type, the probability of supportive object can be calculated by the sum of probability of walkers from the same type of objects walking to



(a) star-schema network



(b) bipartite network

Figure 3: Illustration of the probability estimation process for supportive and pivotal objects. The black dash-dot line represents the random-walk process among supportive objects and the blue dotted line represents the generative process of pivotal objects.

it through the pivotal type. Fig. 3 shows the probability estimation process. The reachable probability of type C can be calculated by random-walkers wandering from type GO and T to type C through type G in Fig. 3(a).

For the second rule (consistency principle), it is an intuitive idea to transfer information among sub-networks. However, what and how do we transfer? It is clear that the sub-networks are overlapped. If we transfer the information of any overlapping types, the model may be hard to control, since two sub-networks may have many overlapping types and one type may appear in many sub-networks. If we do clustering on each sub-network individually, it is difficult to map clusters among sub-networks. We know that the random walk among supportive objects all pass through pivotal objects. So we only need to transfer the information of pivotal type, and then the information can be propagated to other supportive objects by random walkers. In order to maintain the clustering consistency during the iteration, we let the pivotal objects in the current sub-network inherit a portion of clustering information from previous sub-networks with a controlling parameter. The dot line in Fig. 2 shows two information inheritance examples. Specifically, the information on object G calculated in Fig. 2(b) is passed on to the calculation of the pivotal object G in Fig. 2(c) which affects the calculation of object C, while the information on object C is then passed on to the calculation of pivotal object C in Fig. 2(d).

4.2.2 *Reachable Probability for Supportive Objects*

First, we estimate the probability of supportive objects. The path-based random walk process is formulated with matrix representation. We use $M^{(S^I,S^J|P,C)}$ to represent the probability transition matrix from supportive type S^I to type S^J passing pivotal type P in the sub-network C. $M^{(S^I,S^J|P,C)}$ can be calculated as follow:

$$M^{(S^I,S^J|P,C)} = M^{(S^I,P|C)} \times M^{(P,S^J|C)} \quad (1)$$

where $M^{(S^I,P|C)}$ is the transition matrix from S^I to P (i.e., $M^{(S^I,P)}$). Compared to conditional transition matrix $M^{(S^I,S^J|P,C_k)}$ defined below, $M^{(S^I,S^J|P,C)}$ is also called the global transition matrix, which is fixed for the sub-network C. For example, in Fig. 3(a), the global transition matrix $M^{(T,GO|G,C)}$ means the transition probability from type T to GO through G on the sub-network $G - \{T, C, GO\}$. In the proposed model, the global probability of objects is important information to smooth the probability of pivotal objects (see Eq. 8 for more details).

When considering the clustering information, the transition matrices among supportive objects should be adjusted according to clusters. The clustering information can be represented by the membership matrix of pivotal objects, so the conditional transition matrix from S^I to S^J through P in the cluster C_k (i.e., $M^{(S^I,S^J|P,C_k)}$) can be defined as follows:

$$M^{(S^I,S^J|P,C_k)} = M^{(S^I,P|C)} \times B^{(P|C_k)} \times M^{(P,S^J|C)} \quad (2)$$

where $B^{(P|C_k)}$ is the membership of pivotal objects on cluster C_k.

The above transition matrices only consider the clustering information in the current sub-network, which may cause the inconsistency among different sub-networks. For example, in the bibliographical data shown in Fig. 1(c), clustering on the sub-network $P - \{A, V, T\}$ may focus on research areas, while clustering on the sub-network $A - \{P\}$ may more concern about co-author relations. In order to keep the clustering consistency among sub-networks, we can inherit a portion of cluster information from previous sub-networks. Only the clustering information of pivotal type is inherited from previous networks and it is integrated with current clustering information of pivotal type. The reason why the simple mechanism work is that the pivotal objects, as hub node, can propagate the clustering information to all supportive objects. The transition matrices can be redefined as:

$$B''^{(P|C_k)} = \theta_{S,P} \times B'^{(P|C_k)} + (1 - \theta_{S,P}) \times B^{(P|C_k)} \quad (3)$$

$$M^{(S^I,S^J|P,C_k)} = M^{(S^I,P|C)} \times B''^{(P|C_k)} \times M^{(P,S^J|C)} \quad (4)$$

where $B'^{(P|C_k)}$ is the inherited membership matrix when the type P serves as a supportive type in the sub-network whose pivotal type is S; and the $\theta_{S,P}$ is a learning rate parameter that controls the ratio of information inheritance from previous sub-network (pivotal type is S) to current one (pivotal type is P). The dot line in Fig. 2 illustrates the two examples of information inheritance. The new transition matrix has the following advantages. 1) It transfers the clustering information among sub-networks, which keeps the consistency of sub-networks. 2) It helps to speed up the convergence, since the priori clustering information is adopted. For a bipartite network, the transition probability matrix can be denoted as $M^{(S^I,S^I|P,C_k)}$, which has the same calculation mechanism.

The conditional probability of supportive type S^J on sub-network C and cluster C_k are denoted as $Pr(X^{(S^J)}|C) \in [0,1]^{1\times|X^{(S^J)}|}$ and $Pr(X^{(S^J)}|C_k) \in [0,1]^{1\times|X^{(S^J)}|}$. Inspired the PageRank [2], the probability of one type of objects is decided by the reachable probability from other types of objects through pivotal objects. So the conditional probability of supportive type S^J can be defined as follows.

$$Pr(X^{(S^J)}|C) = \sum_{S^I\in\mathcal{S},S^I\neq S^J} Pr(X^{(S^I)}|C) \times M^{(S^I,S^J|P,C)} \quad (5)$$

$$Pr(X^{(S^J)}|C_k) = \sum_{S^I\in\mathcal{S},S^I\neq S^J} Pr(X^{(S^I)}|C_k) \times M^{(S^I,S^J|P,C_k)} \quad (6)$$

The calculation is an iterative process and $Pr(X^{(S^J)}|C_k)$ is initialized as the even value at the first iteration. For a

bipartite network, random walkers start from type S^J and end up with the same type through the pivotal type P. The probability of supportive type S^J, $Pr(X^{(S^J)}|C_k)$ can be defined as $Pr(X^{(S^J)}|C_k) = Pr(X^{(S^J)}|C_k) \times M^{(S^J,S^J|C_k)}$.

4.2.3 Reachable Probability for Pivotal Objects

Then we estimate the probability of pivotal objects. We can consider the pivotal objects are generated by adjacent supportive objects, so a generative model can be adopted here. The probability of pivotal objects comes from two parts: heterogeneous and homogeneous relations (if the pivotal type has self loop). For heterogeneous relations, the heterogeneous probability of pivotal object p in the sub-network C (i.e., $Pr(X_p^{(P)}|C)$) can be calculated as follows:

$$Pr(X_p^{(P)}|C) = \prod_{S^J \in \mathcal{S}q \in N(p)} \prod Pr(X_q^{(S^J)}|C) \tag{7}$$

where $N(p)$ is the set of neighbors of object p in the sub-network. It means the pivotal object p is generated by the different types of adjacent supportive objects. Then, we consider the probability of pivotal object p in a cluster C_k (i.e., $Pr(X_p^{(P)}|C_k)$). Similarly, the probability is also generated from the adjacent supportive objects in the cluster C_k. In addition, we add the global probability of pivotal object $X_p^{(P)}$ to smooth the probability:

$$Pr(X_p^{(P)}|C_k) = \lambda \prod_{S^J \in \mathcal{S}q \in N(p)} \prod Pr(X_q^{(S^J)}|C_k) + (1-\lambda) Pr(X_p^{(P)}|C) \tag{8}$$

where the smooth parameter λ represents the portion of global probability. The smooth operation is an important component due to following reasons. 1) It prevents pivotal objects from accumulating into minority clusters, which helps to improve the clustering accuracy. 2) It makes the probability change of pivotal objects more steady, which can improve the stability of HeProjI. The experiments in Sec. 5.7 also validate the importance of smooth operation.

For homogeneous relations (i.e., the pivotal object has self loop), we can calculate the cluster based homogeneous transition probability for pivotal type as follows:

$$M^{(P|C_k)} = M^{(P|C)} \times B^{(P|C_k)} \tag{9}$$

$M_{.p}^{(P|C_k)}$ denotes the sum of transition probability of other pivotal objects reaching p in cluster C_k, which represents the importance of object p to some extent.

When considering the homogeneous relations (if existing), the probability of pivotal object p is generated by the heterogeneous and homogeneous relations, so it can be calculated as follows:

$$P(X_p^{(P)}|C_k) = Pr(X_p^{(P)}|C_k) \times M_{.p}^{(P|C_k)}. \tag{10}$$

4.3 Posterior Probability for Objects

In order to determine the membership of objects, we need to estimate posterior probability of objects. In each sub-network, there are two kinds of objects (i.e., pivotal and supportive objects). Because pivotal objects are the hub of sub-network that integrate supportive objects and contain complete semantic information, we first estimate the posterior probability of pivotal objects, and then the posterior probability of supportive objects is decided by that of pivotal objects.

Now we consider how to estimate the posterior probability of pivotal objects $P(C_k|X^{(P)})$. According to the Bayesian rule, $P(C_k|X^{(P)}) \propto P(X^{(P)}|C_k) \times P(C_k)$. Since the cluster size $P(C_k)$ is unknown, we need to estimate an appropriate $P(C_k)$ to balance the cluster size. We use the $P(C_k)$ that maximizes the likelihood of generating pivotal objects in different clusters. The likelihood of pivotal objects is defined as:

$$logL = \sum_{p \in X^{(P)}} log[\sum_{k=1}^{K} P(X_p^{(P)}|C_k) \times P(C_k)]. \tag{11}$$

An EM algorithm can be utilized for the latent $P(C_k)$ by maximizing the $logL$. We can derive the Eq. 12 and 13. Initially, we set the $P(C_k)$ with even values and then repeat the E step (i.e., Eq. 12) and M step (i.e., Eq. 13) to iteratively update the latent cluster probability until the $P(C_k)$ obtains convergence.

$$P^t(C_k|X^{(P)}) \propto P(X^{(P)}|C_k) \times P(C_k) \tag{12}$$

$$P^{t+1}(C_k) = \sum_{p \in X^{(P)}} P^t(C_k|X_p^{(P)}) \times \frac{1}{|X^{(P)}|} \tag{13}$$

Next we estimate the posterior of supportive objects. The basic idea is that the posterior probability of supportive objects comes from its pivotal neighborhoods. We define it as follow:

$$P(C_k|X_q^{(S^J)}) = \sum_{p \in N(q)} P(C_k|X_p^{(P)}) \times \frac{1}{|N(q)|} \tag{14}$$

where $P(C_k|X_q^{(S^J)})$ is the probabilities of supportive object $X_q^{(S^J)}$ belonging to cluster C_k; $N(q)$ is the neighbor set of supportive object q. It means that the posterior probability of supportive object $X_q^{(S^J)}$ is the average value of its pivotal neighborhoods.

4.4 Ranking for Objects

Since the probability model obeys the PageRank principle, we can regard the conditional probability of objects as their ranks.

$$Rank(X^{(J)}) \approx P(X^{(J)}|C_k) \tag{15}$$

Because the conditional probability $P(X^{(J)}|C_k)$ in HeProjI is estimated by the random walk process, it may prefer to assign a higher probability to an object with a higher degree. However, in some applications, the link-number based measure is not proper. For example, advertisement webpage may have many poor value links (i.e., high degree but low rank).

If we know the additional information of objects, which can be used to measure the importance of objects, we can integrate the information into the proposed method and then get the more reasonable rank. Based on the conditional probability of objects, we propose a general ranking method for objects as follows:

$$Rank(X^{(J)}) = AI(X^{(J)}) \times P(X^{(J)}|C_k) \tag{16}$$

where the $AI(X^{(J)})$ is the Additional Importance measure (AI) of objects $X^{(J)}$. For example, in bibliographic network, the importance of a paper is decided by its citations to a

large extent, and the AI can be a measure that is proportion to citations. We can also propagate the AI information to adjacent objects by transition probability matrix. It is denoted as follows:

$$Rank(X^{(I)}|C_k) = Rank(X^{(J)}|C_k) \times M^{(J,I)}. \quad (17)$$

4.5 Time Complexity Analysis

Time complexity of HeProjI is composed of two main parts: 1) analyzing each sub-network; 2) handling the projection sequence. In each sub-network, the complexity of estimating the distribution of supportive objects is $O(t_1K|E||S|)$ where $|E|$ is the number of edges in this sub-network, $|S|$ is the number of supportive nodes, and t_1 is the iteration number and K is the cluster number. The complexity of estimating the distribution of pivotal objects is $O(K|E_p|)$ where $|E_p|$ is the number of edges of pivotal objects. Then the time complexity of calculating posterior probability for pivotal objects is $O(t_2K|P|)$ where t_2 is the iteration times, $|P|$ is the number of pivotal objects. Similarly, the posterior probability for supportive objects has the complexity $O(K|E|)$. So the complexity for each sub-network is $O(t_3K(t_1|E||S| + |E_p| + t_2|P| + |E|))$ where t_3 is the iteration number for clustering adjustment in this sub-network. Besides, HeProjI has a projection sequence which selecting different object as the pivotal type. And thus the whole time complexity is $O(t_4|T|t_3K(t_1|E|S+|E_p|+t_2|P|+|E|))$, where $|T|$ is the number of type and t_4 is the iteration of clustering. Omitting tiny and constant items, the time complexity of HeProjI can be summarized by $O(c_1|E| + c_2|P|)$.

5. EXPERIMENTS

In this section, we evaluate the effectiveness of HeProjI, and compare it with several state-of-art methods on three real datasets.

5.1 Datasets

In this paper, we use two real information networks: DBLP and SLAP. These two networks are summarized as follows and their schemas are shown in Fig. 1(c) and (d).

1. DBLP dataset. The dataset is about bibliographic information in computer science domain, which constructs a HIN with four types of objects (paper (P), author (A), venue (V), and term (T)) and their relations. To evaluate the clustering accuracy, we randomly label 1031 papers and 1295 authors with their research areas. In experiments, we extract two different-scaled subsets of the DBLP which are called DBLP-S and DBLP-L, respectively.

DBLP-S: It is a small size dataset which includes three research areas: database (DB), data mining (DM), and information retrieval (IR). There are 21 venues, 25020 papers, 10907 authors and 14940 terms extracted from paper title.

DBLP-L: It is a large dataset which includes 8 areas: computer network, information security, computer architecture, theory, software engineering & programming language, artificial intelligence & pattern recognition, computer graphics, data mining & information retrieval & database. It has 280 venues (35 venues for each area), 275,649 papers, 238,673 authors and 295,123 terms.

2. SLAP dataset [3]. This dataset integrates several well-known bioinformatics datasets (e.g., PubChem, DrugBank, PPI) into a single framework using semantic web technologies for drug discovery. Here SLPA is a simple version which includes 6 types of objects (i.e., gene (G), gene-ontology (GO), chemical compound (C), tissue (T), side effect (Si), substructure (Sub)) and their relations. There are 323 genes, 38,116 compounds, 672 kinds of side effect, 212 kinds of substructure, 170 tissues, 948 gene ontologies and 105,387 links among these objects. We have known a priori that these genes are affiliated to 5 gene families, which are considered as the labels of genes.

5.2 Clustering Effectiveness Study

In this section, we study the clustering effectiveness of HeProjI through comparing it with other well-established algorithms.

The first experiment is done on DBLP dataset, since this dataset has a relatively simple structure and is suitable for comparison with previous algorithms. The representative algorithms are included in experiments, which are summarized as follows.

- HeProjI. It is the proposed algorithm.
- HeProjI$_{\setminus S}$. It is HeProjI without considering the smooth information from general network (i.e., λ is 1 in Eq. 8).
- HeProjI$_{\setminus I}$. It is HeProjI without considering inheriting information from other sub-networks (i.e., Θ is 0 in Eq. 3).
- ComClus [20]. It is a ranking-based clustering method designed for the star-schema network with self loop.
- NetClus [19]. It is a ranking-based clustering method designed for the star-schema network without self loop.
- iTopicModel [15]. It integrates topic model and heterogeneous link information, so it can be used to do clustering in HIN.
- NetPLSA [9]. It regularizes a statistical topic model with a harmonic regularizer based on a graph structure.

The clustering quality is measured by the fraction of vertices identified correctly, FVIC [10, 12], which evaluates the average matching degree by comparing each predicting cluster with the most matching real cluster. The larger the FVIC is the better the partition is. HeProjI, ComClus and NetClus can be applied to DBLP dataset directly. For NetClus, we do not consider the self loop of type P, since NetClus cannot solve it. Note that RankClus [17] is not included here, because it only solves the bipartite network. Moreover, for iTopicModel and NetPLSA, we make a homogeneity assumption of links so that it can be applied to this dataset. The smoothing parameter λ in HeProjI is fixed at 0.9. All learning rate Θ are fixed at 0.3. In HeProjI, the projection sequence of is $P - A - C - T$. The parameters in other algorithms are set with the suggested values in their literals.

Table 1: Clustering accuracy for DBLP dataset

Accuracy		Paper (DBLP-S)	Venue (DBLP-S)	Author (DBLP-S)	Paper (DBLP-L)
HeProjI	**Mean**	**0.857**	**0.823**	**0.725**	**0.603**
	Dev.	0.043	0.047	0.034	0.071
HeProjI$_{\setminus S}$	**Mean**	0.781	0.753	0.698	0.566
	Dev.	0.077	0.069	0.057	0.113
HeProjI$_{\setminus I}$	**Mean**	0.703	0.681	0.605	0.507
	Dev.	0.053	0.045	0.039	0.083
ComClus	**Mean**	0.764	0.775	0.690	0.576
	Dev.	0.020	0.027	0.015	0.024
NetClus	**Mean**	0.742	0.718	0.689	0.566
	Dev.	0.063	0.065	0.051	0.104
iTopicModel	**Mean**	0.512	0.762	0.587	0.361
	Dev.	0.072	0.094	0.073	0.167
NetPLSA	**Mean**	0.466	0.565	0.316	0.338
	Dev.	0.047	0.081	0.023	0.092

From the results shown in Table 1, we can observe that HeProjI achieves the best accuracy and lower standard deviation on all objects. HeProjI$_{\setminus S}$ also has good performances. However, due to omitting the smoothing operation, it has worse performances and stability when compared to HeProjI. The performances of HeProjI$_{\setminus I}$ degrade greatly, since

Table 2: Clustering accuracy for SLAP dataset

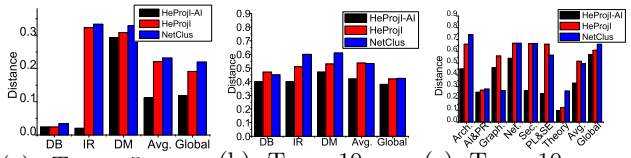
Accuracy	HeProjI		NCut	
	Mean	Dev.	Mean	Dev.
Gene	**0.68**	0.057	0.355	0.165
Chemical Compound	**0.437**	0.031	0.307	0.091
Gene Ontology	**0.557**	0.026	0.261	0.088
Tissue	**0.407**	0.066	0.293	0.09
Side Effect	**0.548**	0.098	0.25	0.056
Substructure	**0.481**	0.053	0.314	0.102

it does not inherit clustering information from other sub-networks. In this condition, HeProjI$_{\setminus I}$ analyzes these sub-networks independently, so the inconsistency among sub-networks causes its bad performances. NetClus and ComClus both have respectable results. However, the absence of citation information among papers may lead to NetClus's worse performances when it is compared with ComClus. The iTopicModel and NetPLSA methods ignore the heterogeneity of objects and relations, so their performances are bad.

For SLAP network, contemporary methods cannot solve it directly. In order to compare with other algorithms, we convert the SLAP network into a homogeneous network through ignoring the heterogeneity of objects. As a comparison algorithm, the classical spectral clustering algorithm, NCut [14], is run on the homogeneous network. The projection sequence is $GO-G-C-T-Sub-Si$. HeProjI uses the same parameters with the above experiments, except the learning rate $\Theta[\theta_{G,GO}, \theta_{GO,G}, \theta_{G,C}, \theta_{G,T}, \theta_{C,Sub}, \theta_{C,Si}] = [0.3, 0.5, 0.7, 0.7, 0.7, 0.7]$. The results are shown in Table 2. It is clear that HeProjI performs much better than NCut. We know that there are distinct differences on different types of objects and relations, e.g., $70,672$ links in $G-C$ relation and 2222 links in $G-GO$ relation. If we do not consider object types, as NCut does, the clusters may be serious unbalanced, which results in the bad performances of NCut.

5.3 Ranking Effectiveness Study

To evalute the ranking effectiveness of HeProjI, we make a ranking accuracy comparison between HeProjI and NetClus. We utilize the venues rank recommended by Microsoft Academic Search [1] as the ground truth. In order to measure the quality of the ranking result, we employ the *Distance* criterion proposed in [11], which computes the differences between two ranking lists of the same set of objects. The criterion not only measures the number of mismatches between two lists but also gives a big penalty term to top mismatch objects in the lists. The smaller *Distance* means the better performance.



(a) Top 5 on DBLP-S (b) Top 10 on AI&PR of DBLP-L (c) Top 10 on DBLP-L

Figure 4: Ranking accuracy comparison on top venues (the smaller *Distance*, the better performance).

Three algorithms are tested on the DBLP dataset. In addition to NetClus, there are two versions of HeProjI (HeProjI with/without AI). The citations of paper are used as the AI measure. We extract the top 5 and 10 venues in different research areas and then calculate the *Distance* measure for them. Additionally, we also compare the accuracy of the global rank on both HeProjI and NetClus. The comparison results are shown in Fig. 4. We can find that two versions of HeProjI achieve better rank performances compared with NetClus in the most cases, since their *Distance* get lower values. Moreover, the HeProjI-AI performs better than HeProjI. In DBLP dataset, the citation information of papers (i.e., AI) reflects the quality of the papers to a large extent. So integrating the AI in HeProjI helps to improve the rank accuracy of papers. Moreover, the citation information can also promote the ranking accuracy of venues through the $P-V$ relation (see Eq. 17). So HeProjI-AI achieves the best ranking performances.

5.4 Case Study

We compare the ranking effectiveness of HeProjI and NetClus with a case study on DBLP dataset. We use the global rank to prove the ranking effectiveness of the HeProjI method. Table 3 shows the top 15 venues ranked by HeProjI and NetClus on DBLP-S. From these results, the ranks of venues generated by HeProjI-AI more conform to the intuition. Although it is hard to rank conferences across different areas, the order within each area is more or less established and the HeProjI-AI confirms with that order. For example, in the DB area, it is SIGMOD, VLDB and ICDE, while in the data mining area, it is KDD, ICDM, and PKDD. However, there are some out of order venues generated by NetClus. For example, among the database conferences, SIGMOD is ranked after VLDB and ICDE. Because NetClus cannot combine additional AI information (i.e., the citations of papers) and tends to get the rank which is propotion to its link number, it has the tendency to rank a good venue publishing a smaller number of papers with a lower rank (e.g., PODS) and a venue publishing a larger number of papers with higher rank (e.g., DEXA). Besides, for HeProjI which does not consider AI information, the rank of venues is basically proportional to their links, since the probability of objects are generated by a random-walk based method. The experiments reflect that the HeProjI method can flexibly and effectively integrate heterogeneous informations and achieve more reasonable ranks.

5.5 Convergence and Stability Study

Now, we study the convergence and stability of HeProjI on the DBLP dataset. The entropy is able to measure the unpredictability of a cluster as well as the convergence of algorithm. We can define the following entropy:

$$AvgEntropy(X^{(J)}) = -\frac{1}{K}\sum_{k=1}^{K}\sum_{p=1}^{|X^{(J)}|} P(C_k|X_p^{(J)}) logP(C_k|X_p^{(J)}). \tag{18}$$

Fig. 5 shows the comparison of AvgEntropy of HeProjI and NetClus on different types of objects of DBLP-S. We can observe that the HeProjI achieves lower *AvgEntropy* on all conditions. We think the reason is that the HeProjI method rationally combines more information from all types of objects. It helps HeProjI to achieve steady solution.

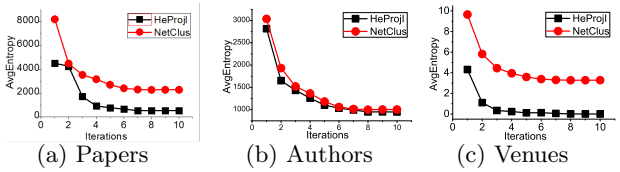
5.6 Time Complexity Study

We recorded the running time of each sub-network in HeProjI along with the iterations on DBLP-S and SLAP in the

Table 3: Top 15 venues in 3 clusters on DBLP-S

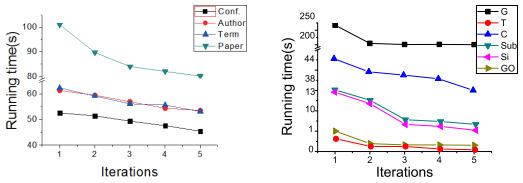
Rank		1	2	3	4	5	6	7	8
HeProjI-AI	Venue	SIGMOD	VLDB	SIGIR	ICDE	KDD	PODS	WWW	CIKM
	#Papers	2428	2444	2509	2832	1531	940	1501	2204
HeProjI	Venue	ICDE	SIGIR	VLDB	SIGMOD	CIKM	DEXA	KDD	WWW
	#Papers	2832	2509	2444	2428	2204	1731	1531	1501
NetClus	Venue	VLDB	ICDE	SIGMOD	SIGIR	KDD	WWW	CIKM	ICDM
	#Papers	2444	2832	2428	2509	1531	1510	2204	1436

Rank		9	10	11	12	13	14	15
HeProjI-AI	Venue	ICDM	EDBT	PKDD	WSDM	PAKDD	DEXA	WebDB
	#Papers	1436	747	680	198	1030	1731	972
HeProjI	Venue	ICDM	PAKDD	PODS	EDBT	PKDD	ECIR	WSDM
	#Papers	1436	1030	1436	747	680	575	198
NetClus	Venue	PODS	DEXA	PAKDD	EDBT	PKDD	WSDM	ECIR
	#Papers	940	1731	1030	747	680	198	575



(a) Papers (b) Authors (c) Venues

Figure 5: The change of AvgEntropy with iterations.



(a) DBLP-S (b) SLAP

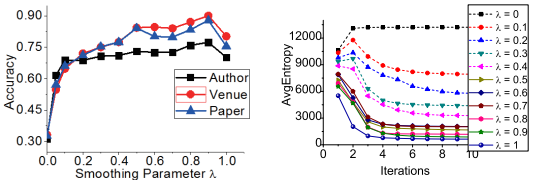
Figure 6: The time of analyzing sub-networks in HeProjI along with the iterations. The pivotal type represents the corresponding sub-networks.

experiments of Sec. 5.2. The results are shown in Fig. 6. We can observe that the complex sub-networks (including more object types and more links) cost much more running time, such as the sub-network $P - \{A, T, C\}$ in Fig. 6(a) and $G - \{T, GO, C\}$ in Fig. 6(b). It is reasonable, since more links and nodes need to be handled in this condition. Moreover, the analysis time of each sub-network decreases along the iteration. We think the prior knowledge inherited from previous iterations on the sub-networks helps to fasten convergence. Although the iteration process in HeProjI results in its higher time complexity, the time used in each iteration drops down quickly in most cases.

5.7 Parameter Study

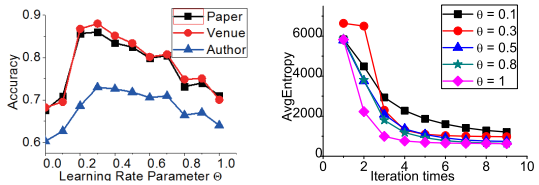
There is a set of parameters in HeProjI: the learning rate vector (i.e., Θ) and the smoothing parameter λ. With the *AvgEntropy* and clustering accuracy for different types of objects, we discuss the effect of different parameter settings on HeProjI.

The smoothing parameter λ is used to control the portion of global probability utilized by each cluster (see Eq. 8). We run HeProjI on DBLP-S with different λ. The results are shown in Fig. 7. Fig. 7(a) shows that HeProjI achieves better performances when λ is from 0.5 to 0.9. It implies that the appropriate global information is helpful for clustering. Too much (λ is small) or no global information (λ is 1) both will degrade the performances of HeProjI. Fig. 7(b) also illustrates that the appropriate global information (i.e., $\lambda \in [0.5, 0.9]$) will benefit for the stability and convergence of algorithms.



(a) Average accuracy (b) Paper *AvgEntropy*

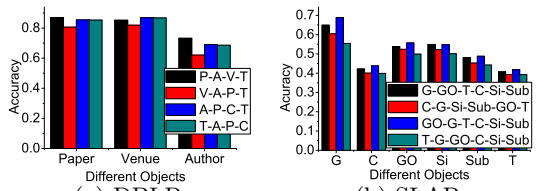
Figure 7: Accuracy and AvgEntropy with different λ.



(a) Average accuracy (b) Paper *AvgEntropy*

Figure 8: Accuracy and AvgEntropy of HeProjI with different Θ.

The learning rates (i.e., Θ) are important parameters which control how much information learned from other sub-networks. We run HeProjI on DBLP-S to observe the effect of Θ on clustering accuracy and convergence. In this experiment, we fixed the smooth parameter λ with 0.9. For convenience, we set the elements of vector Θ with a unified value. From Fig. 8, we can observe that the algorithm accuracy first increases and then decreases with the increment of Θ. It illustrates that either excessive or little information from other sub-networks degrades the algorithm performances. We think it is proper to set the learning rate vector Θ in the range of [0.3, 0.5]. Note that, we set the learning rate with a uniform value for all parameters. However, the learning rate can be different for different sub-networks in real applications. For exmaple, HeProjI achieves good performances on SLAP dataset when setting different learning rates for sub-networks (see Sec. 5.2).



(a) DBLP (b) SLAP

Figure 9: Clustering accuracy of objects with different projection sequences.

5.8 Projection Order Study

In this experiment, we discuss the impact of different projection sequences (i.e., the order of analyzing sub-networks). The experiments are done on DBLP-S and SLAP datasets and the same parameters of HeProjI are set with that of Sec. 5.2. Under different projection sequences, the clustering accuracy of objects are shown in Fig. 9.

On one hand, at first glance, the differences of clustering accuracy are small under different projection sequences, which illustrates HeProjI is not very sensitive to the sequence of analyzing sub-networks. On the other hand, we can also observe that HeProjI consistently has bad performances on all objects under some sequences, such as the sequence $T-G-GO-C-Si-Sub$ in Fig. 9(b). We think the reason lies in the type T objects has a small number of links to other types of objects, so the successive sub-networks can inherit little useful information from it. Although the order of analyzing sub-networks does not have large impact on the performance of HeProjI, we still suggest that HeProjI selects the sub-network whose pivotal type with rich information (e.g., P type in DBLP) or clear semantic meanings (e.g., GO type in SLAP) first.

6. CONCLUSIONS

This paper studied the ranking-based clustering problem in a general heterogeneous information network and proposed a novel algorithm HeProjI. For a general HIN with arbitrary schema, HeProjI projects it into a sequence of projected sub-networks and iteratively analyzes each sub-network. For each sub-network, a path-based random walk model is built to estimate the reachable probability of objects which can effectively be used for clustering and ranking analysis. The experiments show that HeProjI achieves better clustering and ranking result than other representative algorithms.

7. ACKNOWLEDGMENTS

This work is supported by the National Basic Research Program of China (2013CB329603), the National Science Foundation of China (Nos. 61375058, and 71231002), the Ministry of Education of China and China Mobile Research Fund (MCM20130351) and the Beijing Higher Education Young Elite Teacher Project.

8. REFERENCES

[1] `http://academic.research.microsoft.com`.

[2] S. Brin and L. Page. The anatomy of a large-scale hyper textual web search engine. *Comput. Netw. ISDN Syst*, 30(1-7):1757-1771, 1998.

[3] B. Chen, Y. Ding, and D. Wild. Assessing drug target association using semantic linked data. *PLoS Comput. Biol.*, 8(7): 1757-1771, 2012.

[4] M. Grcar and N. Lavrac. A methodology for mining document-enriched heterogeneous information networks. In *Discovery Science*, pages 107-121, 2011.

[5] J. Han. Mining heterogeneous information networks: the next frontier. In *KDD*, page Keynote speech, 2012.

[6] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD*, pages 538-543, 2002.

[7] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *KDD*, pages 1298-1306, 2011.

[8] B. Long, X. Wu, Z. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *KDD*, pages 317-326, 2006.

[9] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *WWW*, pages 101-110, 2008.

[10] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physics Review E*, 69(026113):1757-1771, 2004.

[11] Z. Nie, Y. Zhang, J. R. Wen, and W. Y. Ma. Object-level ranking: bringing order to web objects. In *WWW*, pages 567-574, 2005.

[12] C. Shi, Z. Yan, Y. Cai, and B. Wu. Multi-objective community detection in complex networks. *Applied Soft Computing*, 12(2):850-859, 2012.

[13] C. Shi, C. Zhou, X. Kong, P. S. Yu, G. Liu and W. Bai. Heterecom: a semantic-based recommendation system in heterogeneous networks. In *KDD*, pages 1552-1555, 2012.

[14] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR*, pages 731-737, 1997.

[15] Y. Sun, J. Han, J. Gao, and Y. Yu. Itopicmodel: information network-integrated topic modeling. In *ICDM*, pages 493-502, 2009.

[16] Y. Sun, J. Han, X. F. Yan, P. S. Yu, and T. Wu. PathSim: meta path-based top-K similarity search in heterogeneous information networks. In *VLDB*, pages 992-1003, 2011.

[17] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *EDBT*, pages 565-576, 2009.

[18] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu. Integrating meta-path selection with user guided object clustering in heterogeneous information networks. In *KDD*, pages 1348-1356, 2012.

[19] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, pages 797-806, 2009.

[20] R. Wang, C. Shi, P. S. Yu, and B. Wu. Integrating clustering and ranking on hybrid heterogeneous information network. In *PAKDD*, pages 583-594, 2013.

[21] D. Zhou, S. Orshanskiy, H. Zha, and C. Giles. Co-ranking authors and documents in a heterogeneous network. In *ICDM*, pages 739-744, 2007.