Evaluating Top-k Meta Path Queries on Large Heterogeneous Information Networks

Zichen Zhu, Reynold Cheng, Loc Do, Zhipeng Huang
The University of Hong Kong
Department of Computer Science
Hong Kong
{zczhu2, ckcheng, zphuang}@cs.hku.hk, dohaloc@gmail.com

Haoci Zhang
Columbia University
Department of Computer Science
New York, U.S.
zhanghaoci@gmail.com

***Abstract*—Heterogeneous information networks (HINs), which are typed graphs with labeled nodes and edges, have attracted tremendous interest from academia and industry. Given two HIN nodes s and t, and a natural number k, we study the discovery of the k most important paths in real time. The paths found can be used to support friend search, product recommendation, anomaly detection, and graph clustering. Although related algorithms have been proposed before, they were primarily designed to return the k shortest paths from unlabeled graphs. This leads to two problems: (1) there are often many shortest paths between s and t, and so it is not easy to choose the k best ones; and (2) it is arguable whether a shorter path implies a more crucial one. To address these issues, we study the *top-k meta path query* for a HIN. A meta path abstracts multiple path instances into a high-level path pattern, thereby giving more insight between two nodes. We further study several ranking functions that evaluate the *importance* of meta paths based on *frequency* and *rarity*, rather than on path length. We propose a solution that seamlessly integrates these functions into an A* search framework. The connectivity experiment on ACM dataset shows that our proposed method outperforms state-of-the-art algorithms.**

***Index Terms*—Heterogeneous Information Networks, top-k, meta path**

I. Introduction

Given a graph G and two nodes u and v, a *top-k shortest path query* retrieves the k shortest paths between u and v [5]. This query, which enables the retrieval of relationship information between two graph nodes, has been studied extensively in various applications, including bibliographical, social, and road networks. The continuous growth of the sizes of these graphs and the need of online performance drives the development of efficient algorithms and data structures (e.g., [2], [9], [10], [13], [20]). The query is natively supported in graph database engines (e.g., Neo4j [3] and Pregel [11]).

In this paper, we ask the question: is top-k shortest path query the best way for retrieving relationship from a *heterogeneous information network* (or *HIN*)? An HIN is essentially a *typed graph*, whose nodes and edges are tagged with "type labels" to indicate their meanings. Due to its huge amount of information, HINs have recently raised a lot of interest [7], [15], [16], [18], [19]. Fig. 1 illustrates an ACM bibliographical network [15], where each node and edge has a type (e.g., Jiawei Han is an *author*; IEEE is an *affiliation*; Jiawei Han *belong to* IEEE). Suppose that we want to know how these two authors are related in this HIN. A simple way is to evaluate a top-1 shortest path query on it. However, there are plenty of (13) shortest paths between them. Should all these paths be returned to the query user (who may then get a query result with numerous paths)? Is it better to *summarize* the common properties of these paths (e.g., from these shortest paths, we learnt that the two researchers are close collaborators as they have co-authored many papers)?

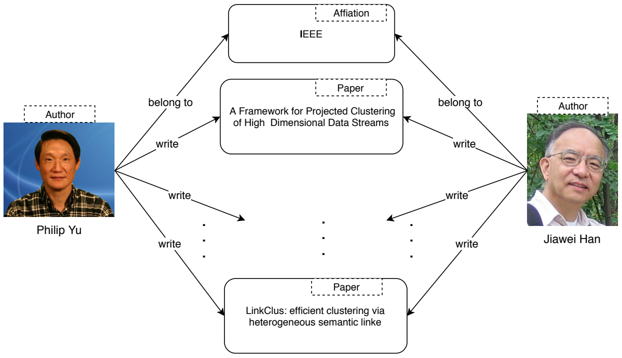

Affiation
IEEE
Paper
A Framework for Projected Clustering of High Dimensional Data Streams
Author
Philip Yu
belong to
write
write
write
Author
Jiawei Han
belong to
write
write
write
Paper
LinkClus: efficient clustering via heterogeneous semantic linke

Fig. 1. Shortest paths between *Philip Yu* and *Jiawei Han* in the ACM network.

To address the above questions, we study how to leverage the type information of a HIN to return the k best paths to a query user. Particularly, we propose the *top-k meta path query*, which computes the k best *meta paths* between nodes u and v. A *meta path*, first proposed in [18], is essentially a sequence of node types and edge types. Two possible meta paths, derived from the shortest paths between Philip and Jiawei in the HIN of Fig. 1, are:

$$M_1: \quad \textit{Author} \xrightarrow{\textit{belong to}} \textit{Affiliation} \xrightarrow{\textit{belong to}^{-1}} \textit{Author}$$
$$M_2: \quad \textit{Author} \xrightarrow{\textit{write}} \textit{Paper} \xrightarrow{\textit{write}^{-1}} \textit{Author}$$

where $X \xrightarrow{R^{-1}} Y$ means the reverse direction of edge R (i.e., $Y \xleftarrow{R} X$). The above meta paths summaries the underlying paths between Philip and Jiawei. For example, M_1 states that the two authors belong to the same affiliation (i.e., IEEE), while M_2 is based on the fact they have co-authored a number of papers. A meta path abstracts the paths and provides important insights about paths. This allows a query user to focus on the high-level relationship patterns, rather than on the detailed path instances. Meta paths have also been shown to be useful in node relevance computation [7], [15], [18], graph clustering [19], and recommendation [16].

To evaluate a top-k meta path query, we need to decide the k *best* meta paths. This is not trivial. Let us again consider the top-1 meta path query for (Philip Yu, Jiawei Han) in Fig. 1. Should we choose M_1 or M_2? Should we only consider the k shortest meta paths? To answer the first question, notice that although M_1 and M_2 have the same length, M_2 is in fact *more important* than M_1. This is because Philip and Jiawei have co-authored 12 papers, and so M_2 summarizes 12 distinct paths between them. On the other hand, M_1 only represents one path (i.e., *Philip Yu – IEEE – Jiawei Han*). Moreover, we found that most authors in the HIN are registered IEEE members, so M_1 is not a very special relationship between these two researchers. On the other hand, it is relatively uncommon that two authors have co-authored in more than ten papers. Hence, in this scenario, M_2 is arguably a better candidate M_1 for the result of the top-1 meta path query.

For the second question (i.e., whether a shorter meta path is better), let us consider two other meta paths between Philip and Jiawei.

$$M_3: \quad Author \xrightarrow{write} Paper \xrightarrow{cite^{-1}} Paper \xrightarrow{write^{-1}} Author$$

$$M_4: \quad Author \xrightarrow{write} Paper \xrightarrow{cite} Paper \xrightarrow{write^{-1}} Author$$

which depict "citation" relationship (i.e., an author's paper cites another author's work). We found that M_3 and M_4 both represent more than 13 paths, and are much more than the single path that instantiates M_1. Also, M_3 and M_4 are less common among other authors than M_1. Hence, although M_1 has a shorter length than these two meta paths, it may not necessarily be ranked higher than M_3 and M_4.

Our contributions. From the above examples, we observe that in answering a top-k meta path query, we should not only focus on the shortest meta paths, but should also consider other factors (e.g., number of paths pertaining to the meta paths, and the "uniqueness", or *rarity*, of a meta path). We present an A* search algorithm framework, and propose an *importance function*, which captures several properties of a meta path (e.g., its number of instances (or "support"), degree of commonality, and meta path lengths) for determining its ranking with respect to the query. Because the importance function incorporates existing relevance measures for meta paths, our proposed solution is generic and supports these measures. We also study a new importance, which effectively captures the importance of a meta path. And we have tested our solutions and performed case studies on ACM. The experimental results show that our best solution is effective in answering top-k meta path queries.

Organization. The rest of this paper is as follows. Section II discusses related work. Section III formally defines the top-k meta path query. Section IV introduces some basedlines and our solution framework. Section V gives more details of how to compute the importance function. Section Section VI presents our experimental results. We conclude in Section VII.

II. Related Work

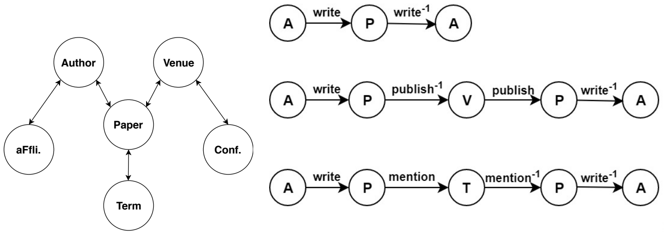
Top-k shortest path query has attracted a lot of interest in the graph database community. These existing research works often design fast algorithms. For example, [5] improves the algorithm performance and reduces the space cost based on the seminal Yen's algorithm [20]. Besides the classical top k-shortest path search, there are also other variants. In [9], an *A* Prune* algorithm is developed to tackle the K *Multiple-Constrained-Shortest-Path problem* (KMCSP). The authors in [2] design an iterative bounding approach and two index structures, namely *Partial Shortest Tree* and *Incremental Shortest Path Tree*, to reduce the search space for the K *Shortest Paths Join* (KSPJ). The work of [10] formalizes the K *Shortest Paths with Diversity* (KSPD), and proposes a general framework to identify k-shortest paths, such that the paths are dissimilar with each other, and the total length of the paths is minimized. In addition, [13] proposes an approach to synthesize an approximate algorithm, which can be applied to the *Single Source Shortest Path* (SSSP) query.

HIN and meta paths. A solution [8] that generates meta paths is parametrized by the maximum length of a meta path l, which is hard to set, as it varies among different data sets. We propose a general framework to explore top-k important meta paths between two given nodes in a HIN without specifying the maximum length. To avoid setting l, [12] proposes a *Forward Stage-wise Path Generation* (FSPG) method which adopts the *Least-Angle Regression* (LARS) [4] to discover meta paths according to the some example node pairs. However, this machine-learning-based solution involves lots of computation and examples, and so it is not suitable for top-k meta paths queries. Apart from these examples-based methods, there is another approach [6] that explores top-k sub-graphs for the whole graph based on a given pattern. However, few papers mentioned how to select the top-k most representative meta paths for a given pair of nodes. [14] mentions a method to mine interesting meta paths in a complex HIN, but this method does not support the top-k meta path query. There is a weight function for meta paths in [19], but it aims to train the weight according to several given meta paths, but it is also not clear how it can be applied in top-k meta path query. There are other weight functions used in [16] and [21] for a meta path, which can then be applied in exploring top-k meta paths, and we will compare these two methods inthe paper.

III. Preliminaries

At first we introduce some terminologies and give a formal definition of the top-k meta paths exploration problem.

Definition 1. ***Heterogeneous Information Network [18]*** *A* heterogeneous information network(HIN) *is a directed graph $G = (V, E)$ with an object type mapping $\phi : V \to \mathcal{A}$ and a link type mapping $\psi : E \to \mathcal{R}$ subject to $|\mathcal{A}| > 1$ and $|\mathcal{R}| > 1$, where V denotes the object set and $E \subseteq V \times V$ denotes the link set, and $\mathcal{A}$ denotes the object type and $\mathcal{R}$ denotes the link type.*


Author
Venue
aFfli.
Paper
Conf.
Term
A write P write-1 A
A write P publish-1 V publish P write-1 A
A write P mention T mention-1 P write-1 A

(a) Network Schema (b) Meta Paths

Fig. 2. ACM dataset

Network schema is proposed to better understand meta-level description of a given heterogeneous information network, which can greatly simplify many complicated HINs.

Definition 2. ***Network Schema [17]*** *The* network schema *is a meta template for a heterogeneous network $G = (V, E)$ with the object type mapping $\phi : V \rightarrow \mathcal{A}$ and the link type mapping $\psi : E \rightarrow \mathcal{R}$, which is a directed graph defined over objects types $\mathcal{A}$, with edges as relations from $\mathcal{R}$, noted as $T_G = (\mathcal{A}, \mathcal{R})$.*

Definition 3. ***Meta Path [18]*** *A* meta path $\mathcal{P}$ *is path defined on the graph of network schema $T_G = (\mathcal{A}, \mathcal{R})$, and is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$. If there is no ambiguity, we can short it as $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between $A_1, A_2, \cdots, A_{l+1}$, where $\circ$ denotes the concatenation operator on relations.*

Example 1. *A typical HIN is the bibliographic network of ACM and the corresponding network schema is shown in Fig. 2a, where there are six types of entities: paper, venue, author, term, affliatio and conference. Links exist between authors and papers denoting the writing or written-by relations, between papers and topics denoting mentioning or mentioned-by relations, and between papers and venues denoting the publishing or published-in relations, between papers and papers denoting citing or cited-by relations. Fig. 2b lists 3 very common meta paths between 2 authors in ACM data set: two authors writes at least one paper together, two authors published some paper in the same venue, two authors published some paper mentioning the same term.*

Here we use lower-case letters(e.g., a_i) to denote objects in HIN while upper-case letters(e.g., A) to denote object types in T_G. We say a concrete path $p = (a_1 a_2 \cdots a_{l+1})$ between objects a_1 and a_{l+1} in network G follows the meta path $\mathcal{P}$, or say it is a **path instance** of the relevance path $\mathcal{P}$, which can be noted as $p \in \mathcal{P}$, if $\forall a_i \in V, \varphi(a_i) = A_i$ and $\forall e_i = \langle a_i, a_{i+1} \rangle \in E, \psi(e_i) = R_i$ in $\mathcal{P}$. We also note $\mathcal{P}^{-1}$ as the **reverse meta path** of $\mathcal{P}$, which defines an inverse relation of $\mathcal{P}$ in T_G. If a meta path $\mathcal{P}$ is equal to $\mathcal{P}^{-1}$ such as APA, this meta path $\mathcal{P}$ can be called a **symmetric meta path**. Two meta paths $\mathcal{P}_1 = (A_1 A_2 \cdots A_l)$ and $\mathcal{P}_2 = (B_1 B_2 \cdots B_k)$ are **concatenable** if and only if A_l is equal to B_1, and the concatenated path is written as $\mathcal{P} = \mathcal{P}_1 \circ \mathcal{P}_2$, which equals to $(A_1 A_2 \cdots A_l B_2 \cdots B_k)$.

IV. Querying Top-K Important Meta-Paths

In this section, we start with some baselines and then introduce a general framework to evaluate the importance of a meta path between two objects $s, t \in V$ in a given heterogeneous information network $G = (V, E)$. After that we discuss how we apply this model in A* searching.

A. baselines

SMP. k-Shortest Meta Paths (noted as **SMP**) is the simplest solution, since the length is one intrinsic characteristic of a meta path $\mathcal{P}$. There is also a proof, derived from [18], showing that too long meta paths are not necessarily useful in determining similarity between two objects in a HIN. A naiive solution to **SMP** is to perform breadth-first search (BFS) from s to find all shortest paths connecting to t, which are then used to construct k-shortest meta paths.

SLV1 and **SLV2**. A strength concept of a meta path $\mathcal{P}$ is proposed to evaluate its importance. For $\mathcal{P}$ of more than one relations, i.e., $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, the corresponding strength can be calculated as follows.

$$Strength(\mathcal{P}) = \prod_{i=1}^{l} \frac{1}{\sqrt{O(A_i \xrightarrow{R_i} A_{i+1}) * I(A_i \xrightarrow{R_i} A_{i+1})}} \tag{1}$$

where $O(A_i \xrightarrow{R_i} A_{i+1})$ is the average out-degree of A_i-type nodes, and $I(A_i \xrightarrow{R_i} A_{i+1})$ is the average in-degree of A_{i+1}-type nodes on the relation type R_i from A_i to A_{i+1}. Considering strength and length, [16], [21] propose Strength-and-Length-based Versioned (noted as **SLV**) importance function as follows.

$$SLV1 : \mathcal{I}(\mathcal{P}) = e^{Strength(\mathcal{P})} * e^{-|\mathcal{P}|} \tag{2}$$

$$SLV2 : \mathcal{I}(\mathcal{P}) = Strength(\mathcal{P}) * |\mathcal{P}|^{-1} \tag{3}$$

B. Importance Function of A Meta Path

We posit that length, path instances, global and local relationships between the given node pairs are fundamental to evaluate importance of a meta path. Therefore, the meta path importance function can be generalized as follows:

$$\mathcal{I}_{s,t}(\mathcal{P}) = \mathcal{S}_{s,t}(\mathcal{P}) * \mathcal{R}_{s,t}(\mathcal{P}) * Penalty(|\mathcal{P}|) \; (\mathcal{P} \in \mathbb{P}_{s \rightarrow t}) \tag{4}$$

where $\mathbb{P}_{s \rightarrow t}$ refers to the meta path set in which each one can connect the two nodes, s and t. And in (4), for a meta path $\mathcal{P}$, $\mathcal{S}_{s,t}(\mathcal{P})$ refers to the support of this meta path between the two nodes s and t, $\mathcal{R}_{s,t}(\mathcal{P})$ refers to the rarity between them. The last part in (4), $Penalty(|\mathcal{P}|)$, is the penalty function of its length, $|\mathcal{P}|$. More details of the importance function will be discussed in the following sections.

C. How to Make it Support A Searching*

The proposed top-k important meta paths searching is performed by an A* manner, as shown in Algorithm 1. $MetaNode$ is a class to represent current expanded meta path and store relevant infomation such as $\mathcal{I}(\mathcal{P})$. The whole algorithm is based on a priority queue(noted as Q) of many

$MetaNode$ instances, which is decreasingly sorted by the upper bound of its importance(noted as $\overline{\mathcal{I}(\mathcal{P})}$) and the algorithm will end only if the minimum $\mathcal{I}(\mathcal{P}$ of explored top-k meta paths(noted as $mPaths$) is greater than $\overline{\mathcal{I}(\mathcal{P})}$ of the first node in Q. To guarantee the A* algorithm work, the importance function $\mathcal{I}(\mathcal{P})$ should have the **monotonic-decreasing-maximum** property.

Definition 4. Monotonic-Decreasing-Maximum Property *The* monotonic-decreasing-maximum property *of the importance function $\mathcal{I}_{s,t}(\mathcal{P})$ is defined as follows: given a graph $G = (V, E)$, its corresponding schema $T_G = (\mathcal{A}, \mathcal{R})$, and a pair of (s,t) where $s, t \in V$, $\forall \mathcal{P} \in T_G$, $\forall R_i \in \mathcal{R}$, $\overline{\mathcal{I}_{s,t}(\mathcal{P})} \geq \overline{\mathcal{I}_{s,t}(\mathcal{P} \circ R_i)}$.*

Algorithm 1 Top-K Important Meta Paths Discovery

```
Input: Network $G$, two nodes $s, t \in G$ and the number $k$
Output: Top-$k$ Important Meta Paths between $s$ and $t$
if not $isConnected(s, t, G)$ then
    $output("No\ Meta\ Paths\ Found")$
    return {}
end if
$D_{s,t} \leftarrow GetSimilarPairs(s, t, G)$
$root \leftarrow initMetaNode(s, G)$
$mPaths \leftarrow \{\}$
$Q \leftarrow initPriorityQueue(MetaNode.class)$
$Q.push(root)$
while not $Q.empty()$ do
    $N \leftarrow Q.pop()$
    if $len(mPaths) == k$ then
        if $\overline{\mathcal{I}(N.mPath)} < \mathcal{I}(mPaths[k-1])$ then
            $break$
        end if
    end if
    $new\_mPaths \leftarrow expand(N, G)$
    for each $mPath \in new\_mPaths$ do
        if $t$ and $s$ is connected through $mPath$ then
            $Ipt \leftarrow \mathcal{I}(s, t, mPath)$
            $updateMetaPaths(mPaths, Ipt, mPath)$
        end if
        $Q.push($new $MetaNode(mPath))$
    end for
end while
return $mPaths$
```

V. Computing Importance

In this section, we will discuss how to compute each part of $\mathcal{I}_{s,t}(\mathcal{P})$ in detail.

A. Length Penalty Function

For a meta path $\mathcal{P}$, a length penalty function aims to diminish its importance when $\mathcal{P}$ is longer. There have been soome functions to penalize the meta path's length such as $e^{-|\mathcal{P}|}$ in [16] or $\frac{1}{|\mathcal{P}|}$ in [21]. [12] also used $\beta^{|\mathcal{P}|}$ as a penalty function, where β is a decay factor ranging in the open interval $(0, 1)$, and this can be seen as a generalized version of $e^{-|\mathcal{P}|}$. We employ the $\beta^{|\mathcal{P}|}$ in this paper, as we can tune the β to adjust how much we penalize the length.

B. Rarity Function

The rarity function is designed to evaluate how rare the meta path P is among similar pairs to (s,t) in a given HIN $G = (V, E)$. And the similar pairs(noted by $D_{s,t}$) is defined as follows:

$$D_{s,t} = D_t \cup D_s \tag{5}$$

where

$$\begin{aligned} D_t &= \{(s, v) | v \in V, \phi(v) \cap \phi(t) \neq \emptyset\} \\ D_s &= \{(v, t) | v \in V, \phi(v) \cap \phi(s) \neq \emptyset\} \end{aligned} \tag{6}$$

inspired by the inverse document frequency(**IDF**) in TF-IDF, we evaluate the rarity $\mathcal{P}$ in a similar way:

$$\mathcal{R}_{s,t}(\mathcal{P}) = log \frac{|D_{s,t}|}{|\{(u, v) \in D_{s,t}, \mathcal{P} \in \mathbb{P}_{u \to v}\}|} \tag{7}$$

Because we do not exclude s in D_s or t in D_t, it is easy to prove that $\forall \mathcal{P} \in \mathbb{P}_{s \to t}, \mathcal{R}_{s,t}(\mathcal{P}) \leq log|D_{s,t}|$.

C. Support

To guarantee the monotonic-decreasing-maximum property of $\mathcal{I}_{s,t}(\mathcal{P})$, we should pay special attention to the support function. According to the definition of $\mathcal{I}_{s,t}(\mathcal{P})$ and the maximum rarity, we can have the following equation:

$$\overline{\mathcal{I}_{s,t}(\mathcal{P})} = log|D_{s,t}| * \overline{\mathcal{S}_{s,t}(\mathcal{P})} * Penalty(\mathcal{P}) \tag{8}$$

Since the penalty function is strict decreasing, we thus require $\mathcal{S}_{s,t}(\mathcal{P})$ have the monotonic-decreasing-maximum property to guarantee the same property of $\overline{\mathcal{I}_{s,t}(\mathcal{P})}$.

D. Binary Support (BS)

A naive idea to design the support is a binary function, which returns a positive constant if $\mathcal{P} \in \mathbb{P}_{s \to t}$, and returns 0 if $\mathcal{P} \notin \mathbb{P}_{s \to t}$. The constant function always has the monotonic-decreasing-maximum property because it is bounded by a constant. For example, we set the constant as 1, and the corresponding binary support function is as shown as follows:

$$\mathcal{S}_{s,t}(\mathcal{P}) = \begin{cases} 1, & \mathcal{P} \in \mathbb{P}_{s \to t} \\ 0, & \mathcal{P} \notin \mathbb{P}_{s \to t} \end{cases} \tag{9}$$

$\overline{\mathcal{S}_{s,t}(\mathcal{P})}$ can be seen as the constant value, 1, which means that $\mathcal{S}_{s,t}(\mathcal{P})$ can be applied into our A* algorithm. Therefore, we now have a complete importance function(*Binary-Supported*, noted as **BS**). This importance function can also be seen as the combination of the rarity and the penalty function, because the support remains the same for $\mathcal{P} \in \mathbb{P}_{s \to t}$.

E. MNI-based Support (MNIS)

Inspired from the Minimum Image [1], we design another support function to reflect the frequency without violating the monotonic-decreasing-maximum property. The definition of *Minimum Instances*(short as **MNI**) is given as follows.

Definition 5. Minimum Instances *The* minimum instances *of a meta path $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$ between a given pair (s,t) refers to the minimum instance number of each node type in that meta path $\mathcal{P}$(not including two ends). If we note as p_i the node v in the i-th node type($1 < i < l+1$) in* *a path instance p of $\mathcal{P}$, we then can get the following equation to calculate the* ***MNI*** *for a meta path $\mathcal{P}$:*

$$MNI(\mathcal{P}) = \min_{1<i<l+1} |\{p_i|p \in \mathcal{P}\}| \qquad (10)$$

The $|\{p_i|p \in \mathcal{P}\}|$ in (10) *refers to the instance number of the i-th node type in the meta path $\mathcal{P}$. And obviously,* ***MNI*** *of a meta path $\mathcal{P}$ has the monotonic-decreasing-maximum property because $\min_{1<i<l+1} |\{p_i|p \in \mathcal{P}\}| \geq \min_{1<i<l+2} |\{p_i|p \in \mathcal{P} \circ R_i\}|$.*

However, compared with one-to-many relation, such as $Paper \xrightarrow{publish^{-1}} Venue$, one-to-one relation such as $Paper \xrightarrow{mention} Topic$ could always have smaller **MNI** because one paper can mention several topics but it can be only published in one venue. To eliminate the unbalance, we multiply **MNI** by another strength function :

$$Strength(A \xrightarrow{R} B) = \frac{1}{\min(O(A \xrightarrow{R} B), I(A \xrightarrow{R} B))} \qquad (11)$$

Note that we only consider those A-type nodes and B-type nodes that have at least one relation $A \xrightarrow{R} B$ when we compute the average out-degree and in-degree. Therefore, multiplying **MNI** by this function will keep the monotonic-decreasing-maximum property of the $\mathcal{I}(\mathcal{P})$. And we can offline compute $O(A \xrightarrow{R} B)$ and $I(A \xrightarrow{R} B)$ of each relation for every strength function and reuse them to accelerate top-k meta paths query.

Synthesizing the above discussion, we can combine $MNI_{s,t}(\mathcal{P})$ with $Strength(\mathcal{P})$ and then construct the **MNI** support function in our framework:

$$\mathcal{S}_{s,t}(\mathcal{P}) = Strength(\mathcal{P}) * MNI_{s,t}(\mathcal{P}) \qquad (12)$$

We thus have another version of importance function (*strengthened-MNI Supported* noted as **MNIS**) by using the strengthened **MNI** support and keeping other parts in $\mathcal{I}(\mathcal{P})$ unchanged. By incorporating the strength, $\mathcal{S}_{s,t}(\mathcal{P} \circ R_i)$ of the next $MetaNode$ can be calculated by multiplying the current $\mathcal{S}_{s,t}(\mathcal{P})$ and the strength of the expanding relation R_i, which will reduce $\mathcal{S}_{s,t}(\mathcal{P})$ significantly and thus A* searching will end much faster.

VI. Results

We now discuss a case study and a connectivity experiment by comparing the performance of MNIS, BS with SMP, SLV1 and SLV2. These results can help us validate the effectiveness of MNIS. All the experiment programs are complied with 4.6.3-version gcc and executed in Ubuntu 12.04.01 with 8 Intel(R) Core(TM) i7-3770 processors.

A. Data set

Our experiments employ the HIN of **ACM data set** [15]. The ACM data set contains 14 representative computer science conferences: KDD, SIGMOD, WWW, SIGIR, CIKM, SODA, STOC, SOSP, SPAA, SIGCOMM, MobiCOM, ICML, COLT and VLDB. These conferences contain 196 venues proceedings. This dataset includes 12K papers, 17K authors, 1.8K affiliations and 1.5K frequent terms. Each paper is also labeled by a subject and we extract subjects for further use. After that, there are 1096K links remaining in this data set. The corresponding network schema is shown in Fig 2a.

B. Case Study

In this section, we first study the effectiveness of our approach on ACM data set through exploring the top-k relationship between *Jiawei Han* and *Philip Yu* with different methods. We will follow the definition of M_1, M_2, M_3, M_4 in Section I to simplify our discussion. We use k-shortest path to explore top 4 path instances between them and we found 13 path instances that have the same length. Among these 13 paths, 12 paths have the same meaning in meta level: they are coauthors for 12 papers, which can be expressed by M_2 (i.e., $A \rightarrow P \leftarrow A$), while the left instance can be captured by M_1.

TABLE I
Top 4 Meta Paths of < *J. Han, P. Yu* > by MNIS ($\beta = 0.3$)

Rank	Meta Path $\mathcal{P}$	$\mathcal{I}(\mathcal{P})$	MNI	Rarity
1	$A \rightarrow P \leftarrow A$	0.154	12	5.8844
2	$A \rightarrow P \leftarrow P \leftarrow A$	$\mathbf{2.4 * 10^{-3}}$	**14**	**4.3884**
3	$A \rightarrow P \rightarrow P \leftarrow A$	$\mathbf{2.1 * 10^{-3}}$	**13**	**4.1258**
4	$A \rightarrow P \leftarrow A \rightarrow P \leftarrow A$	$3.65 * 10^{-4}$	21	3.6735

TABLE II
Top 4 Meta Paths of < *J. Han, P. Yu* > by SLV1 and SLV2

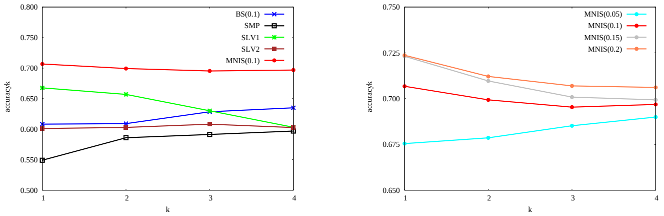
Rank	Meta Path $\mathcal{P}$	$\mathcal{I}(\mathcal{P})$ (SLV1)	$\mathcal{I}(\mathcal{P})$ (SLV2)
1	$A \rightarrow P \leftarrow A$	$1.2 * 10^{-2}$	0.139
2	$A \rightarrow F \leftarrow A$	$4.86 * 10^{-4}$	0.135
3	$A \rightarrow P \rightarrow P \leftarrow A$	$\mathbf{4.82 * 10^{-4}}$	$\mathbf{4.99 * 10^{-2}}$
4	$A \rightarrow P \leftarrow P \leftarrow A$	$\mathbf{4.82 * 10^{-4}}$	$\mathbf{4.99 * 10^{-2}}$

The top 4 relationships between *Jiawei Han* and *Philip Yu* (abbreviated as < *J. Han, P. Yu* >) by MNIS are shown in Table I. Because the top-4 rankings of SLV1 and SVL2 are the same, they are both put in Table II.

According to the result, all methods rank M_2 the first place but when it comes to the top 4 meta paths, much more difference emerged. In the result of MNIS, M_1 (short as $A \rightarrow F \leftarrow A$) can not even rank into the top 4 important relationships. In addition to the ranking of M_1, from Table II we can also see that SLV1 and SLV2, these two methods do not consider the direction of the edge and thus the weights of M_3 (i.e., $A \rightarrow P \leftarrow P \leftarrow A$) and M_4 (i.e., $A \rightarrow P \rightarrow P \leftarrow A$) are the same, which means that SLV1 and SLV2 can not differentiate two meta paths if they share the same edge types but some of them have different directions. Compared with SLV1 and SLV2, MNIS can differentiate M_3 and M_4. In a conclusion of this case study, only MNIS can rank M_2 as the top one and differentiate M_3 and M_4 at the same time.

C. Label-based Connectivity Analysis

In the original ACM data set, each paper is related to just one subject, and thus we extract these subjects as labels for those papers to evaluate the performance of the label-based connectivity. We first sample 100 positive pairs and 100 negative pairs from all papers where a positive pair means two papers are related to the same subject, and a negative pair



(a) MNIS and Baselines Comparison (b) Beta Comparison in MNIS

Fig. 3. Accuracy@k of Author Connectivity Analysis

means two papers are related to different subjects. Then, for the j-th pair($j = 1 \leq j \leq 100$) of two papers in positive pair, we compute the top 4 meta paths(noted by $\{\mathcal{P}_{i,j} | 1 \leq i \leq 4\}$) between them according using different algorithms. For the i-th meta path $\mathcal{P}_{i,j}$ obtained by the j-th pair, we now can count how many positive pairs can be connected by it(the number of connected positive pairs is noted as $p_{i,j}$) and how many negative pairs can be connected by it(the number of connected negative pairs is noted as $n_{i,j}$). The accuracy of the i-th meta path $\mathcal{P}_i$ for the j-th pair can be calculated by $\dfrac{p_{i,j}}{p_{i,j} + n_{i,j}}$. Finally we evaluate the performance by the average accuracy of the top-k meta paths, noted by $\overline{Accuracy@k}$:

$$\overline{Accuracy@k} = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{100} \sum_{j=1}^{100} \frac{p_{i,j}}{p_{i,j} + n_{i,j}} \quad (k = 1..4) \qquad (13)$$

The comparison result between MNIS ($\beta = 0.1$) and other baselines are shown in Fig. 3a, which reveals that MNIS outperforms any other baselines in this task. Furthermore, from Fig. 3b, we can also see how different β ($\beta = 0.05, 0.1, 0.15, 0.2$) influences on the $\overline{Accuracy@k}$. The result shows that a larger β generates a better result, and this is because a larger β indicates a larger searching space which makes those important but longer meta paths can be explored.

VII. Conclusion

In this paper, we study the problem of discovering top-k important meta paths between two nodes in a HIN. We propose a ranking function for meta paths based on frequency and rarity, and design an A* search algorithm for efficient top-k search. The connectivity experiment on ACM shows that our proposed method outperforms state-of-the-art methods.

Acknowledgment

Reynold Cheng, Zichen Zhu, Loc Do, and Zhipeng Huang were supported by the Research Grants Council of Hong Kong (RGC Projects HKU 106150091, 17229116, 17205115) and the University of Hong Kong (Projects 104004572, 102009508 ,104004129). We would like to thank the reviewers for their insightful comments.

References

[1] B. Bringmann and S. Nijssen, "What is frequent in a single graph?" in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2008, pp. 858–863.

[2] L. Chang, X. Lin, L. Qin, J. X. Yu, and J. Pei, "Efficiently computing top-k shortest path join," in *Extending Database Technology*, 2015.

[3] N. Developers, "Neo4j," *Graph NoSQL Database [online]*, 2012.

[4] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[5] J. Gao, H. Qiu, X. Jiang, T. Wang, and D. Yang, "Fast top-k simple shortest paths discovery in graphs," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 509–518.

[6] M. Gupta, J. Gao, X. Yan, H. Cam, and J. Han, "Top-k interesting subgraph discovery in information networks," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 820–831.

[7] Z. Huang, Y. Zheng, R. Cheng, Y. Sun, N. Mamoulis, and X. Li, "Meta structure: Computing relevance in large heterogeneous information networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1595–1604.

[8] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Machine learning*, vol. 81, no. 1, pp. 53–67, 2010.

[9] G. Liu and K. Ramakrishnan, "A* prune: an algorithm for finding k shortest paths subject to multiple constraints," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2001, pp. 743–749.

[10] H. Liu, C. Jin, B. Yang *et al.*, "Finding top-k shortest paths with diversity," *IEEE Transactions on Knowledge and Data Engineering*, 2017.

[11] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: a system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 135–146.

[12] C. Meng, R. Cheng, S. Maniu, P. Senellart, and W. Zhang, "Discovering meta-paths in large heterogeneous information networks," in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 754–764.

[13] Z. Shang and J. X. Yu, "Auto-approximation of graph computing," *Proceedings of the VLDB Endowment*, vol. 7, no. 14, pp. 1833–1844, 2014.

[14] B. Shi and T. Weninger, "Mining interesting meta-paths from complex heterogeneous information networks," in *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*. IEEE, 2014, pp. 488–495.

[15] C. Shi, X. Kong, Y. Huang, S. Y. Philip, and B. Wu, "Hetesim: A general framework for relevance measure in heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2479–2492, 2014.

[16] C. Shi, C. Zhou, X. Kong, P. S. Yu, G. Liu, and B. Wang, "Heterecom: a semantic-based recommendation system in heterogeneous networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1552–1555.

[17] Y. Sun and J. Han, "Mining heterogeneous information networks: a structural analysis approach," *Acm Sigkdd Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

[18] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[19] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, "Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 7, no. 3, p. 11, 2013.

[20] J. Y. Yen, "Finding the k shortest loopless paths in a network," *management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[21] T. Zhu, Z. Peng, S. Wang, S. Y. Philip, and X. Hong, "Measuring the relevance of different-typed objects in weighted signed heterogeneous information networks," in *Computer Supported Cooperative Work in Design (CSCWD), 2017 IEEE 21st International Conference on*. IEEE, 2017, pp. 556–561.