MORGAN & CLAYPOOL PUBLISHERS

Mining Heterogeneous Information Networks

Principles and Methodologies

Yizhou Sun
Jiawei Han

SYNTHESIS LECTURES ON DATA MINING AND KNOWLEDGE DISCOVERY

Jiawei Han, Lise Getoor, Wei Wang, Johannes Gehrke, Robert Grossman, *Series Editors*

ABSTRACT

Real-world physical and abstract data objects are interconnected, forming gigantic, interconnected networks. By structuring these data objects and interactions between these objects into multiple types, such networks become *semi-structured heterogeneous information networks*. Most real-world applications that handle big data, including interconnected social media and social networks, scientific, engineering, or medical information systems, online e-commerce systems, and most database systems, can be structured into heterogeneous information networks. Therefore, effective analysis of large-scale heterogeneous information networks poses an interesting but critical challenge.

In this book, we investigate the principles and methodologies of mining heterogeneous information networks. Departing from many existing network models that view interconnected data as homogeneous graphs or networks, our semi-structured heterogeneous information network model leverages the rich semantics of typed nodes and links in a network and uncovers surprisingly rich knowledge from the network. This semi-structured heterogeneous network modeling leads to a series of new principles and powerful methodologies for mining interconnected data, including: (1) rank-based clustering and classification; (2) meta-path-based similarity search and mining; (3) relation strength-aware mining, and many other potential developments. This book introduces this new research frontier and points out some promising research directions.

KEYWORDS

information network mining, heterogeneous information networks, link analysis, clustering, classification, ranking, similarity search, relationship prediction, user-guided clustering, probabilistic models, real-world applications, efficient and scalable algorithms

Contents

CHAPTER 4

Meta-Path-Based Similarity Search

We now introduce a systematic approach for dealing with general heterogeneous information networks with a specified but arbitrary network schema, using a meta-path-based methodology. Under this framework, similarity search (Chapter 4) and other mining tasks such as relationship prediction (Chapter 5) can be addressed by systematic exploration of the network meta structure.

4.1 OVERVIEW

Similarity search, which aims at locating the most relevant information for a query in a large collection of datasets, has been widely studied in many applications. For example, in spatial database, people are interested in finding the k nearest neighbors for a given spatial object [35]; in information retrieval, it is useful to find similar documents for a given document or a given list of keywords. Object similarity is also one of the most primitive concepts for object clustering and many other data mining functions.

In a similar context, it is critical to provide effective similarity search functions in information networks, to find similar entities for a given entity. In a bibliographic network, a user may be interested in the top-k most similar authors for a given author, or the most similar venues for a given venue. In a network of tagged images such as Flickr, a user may be interested in search for the most similar pictures for a given picture. In an e-commerce system, a user would be interested in search for the most similar products for a given product. Different from the attribute-based similarity search, links play an essential role for similarity search in information networks, especially when the full information about attributes for objects is difficult to obtain.

There are a few studies leveraging link information in networks for similarity search, but most of these studies are focused on homogeneous networks or bipartite networks, such as personalized PageRank (P-PageRank) [29] and SimRank [28]. These similarity measures disregard the subtlety of different types among objects and links. Adoption of such measures to heterogeneous networks has significant drawbacks: even if we just want to compare objects of the same type, going through link paths of different types leads to rather different semantic meanings, and it makes little sense to mix them up and measure the similarity without distinguishing their semantics. For example, Table 4.1 shows the top-4 most similar venues for a given venue, DASFAA, based on (a) the common authors shared by two venues, or (b) the common topics (i.e., terms) shared by two venues. These two scenarios are represented by two distinct meta-paths: (a) $VPAPV$, denoting that the similarity is defined by the connection path "venue-paper-author-paper-venue," whereas (b) $VPTPV$, by

the connection path "venue-paper-topic-paper-venue." A user can choose either (a) or (b) or their combination based on the preferred similarity semantics. According to Path (a), DASFAA is closer to DEXA, WAIM, and APWeb, that is, those that share many common authors, whereas according to Path (b), it is closer to Data Knowl. Eng., ACM Trans. DB Syst., and Inf. Syst., that is, those that address many common topics. Obviously, different connection paths lead to different semantics of similarity definitions, and produce rather different ranking lists even for the same query object.

Table 4.1: Top-4 most similar venues to "DASFAA" with two meta-paths

Rank	path: $VPAPV$	path: $VPTPV$
1	DASFAA	DASFAA
2	DEXA	Data Knowl. Eng.
3	WAIM	ACM Trans. DB Syst.
4	APWeb	Inf. Syst.

To systematically distinguish the semantics among paths connecting two objects, we introduce a meta-path-based similarity framework for objects of the same type in a heterogeneous network. A meta-path is a sequence of relations between object types, which defines a new composite relation between its starting type and ending type. The meta-path framework provides a powerful mechanism for a user to select an appropriate similarity semantics, by choosing a proper meta-path, or learn it from a set of training examples of similar objects.

In this chapter, we introduce the meta-path-based similarity framework, and relate it to two well-known existing link-based similarity functions for homogeneous information networks. Especially, we define a novel similarity measure, PathSim, that is able to find peer objects that are not only strongly connected with each other but also share similar visibility in the network. Moreover, we propose an efficient algorithm to support online top-k queries for such similarity search.

4.2 PATHSIM: A META-PATH-BASED SIMILARITY MEASURE

The similarity between two objects in a link-based similarity function is determined by how the objects are connected in a network, which can be described using paths. For example, in a co-author network, two authors can be connected either directly or via common co-authors, which are length-1 and length-2 paths, respectively. In a heterogeneous information network, however, due to the heterogeneity of the types of links, the way to connect two objects can be much more diverse. For example, in Table 4.2, Column I gives several path instances between authors in a bibliographic network, indicating whether the two authors have co-written a paper, whereas Column II gives several path instances between authors following a different connection path, indicating whether the two authors have ever published papers in the same venue. These two types of connections represent different relationships between authors, each having some different semantic meaning.

Table 4.2: Path instance vs. meta-path in heterogeneous information networks

	Column I: Connection Type I	Column II: Connection Type II
Path instance	Jim-P_1-Ann Mike-P_2-Ann Mike-P_3-Bob	Jim-P_1-SIGMOD-P_2-Ann Mike-P_3-SIGMOD-P_2-Ann Mike-P_4-KDD-P_5-Bob
Meta-path	Author-Paper-Author	Author-Paper-Venue-Paper-Author

Now the questions is, given an arbitrary heterogeneous information network, is there any way to systematically identify all the possible connection types (i.e., relations) between two object types? In order to do so, we propose two important concepts in the following.

4.2.1 NETWORK SCHEMA AND META-PATH

First, given a complex heterogeneous information network, it is necessary to provide its meta level (i.e., schema-level) description for better understanding the network. Therefore, we propose the concept of **network schema** to describe the meta structure of a network. The formal definition of network schema has been given in Definition 1.2 in Chapter 1.

The concept of network schema is similar to that of the ER (Entity-Relationship) model in database systems, but only captures the entity type and their binary relations, without considering the attributes for each entity type. Network schema serves as a template for a network, and tells how many types of objects there are in the network and where the possible links exist. Note that although a relational database can often be transformed into an information network, the latter is more general and can handle more unstructured and non-normalized data and links, and is also easier to deal with graph operations such as calculating the number of paths between two objects.

As we illustrated previously, two objects can be connected via different paths in a heterogeneous information network. For example, two authors can be connected via "author-paper-author" path, "author-paper-venue-paper-author" path, and so on. Formally, these paths are called *meta-paths*, defined as follows.

Definition 4.1 **(Meta-path)** A *meta-path* $\mathcal{P}$ is a path defined on the graph of network schema $T_G = (\mathcal{A}, \mathcal{R})$, and is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$, which defines a composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_l$ between types A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations.

For the bibliographic network schema shown in Figure 4.1 (a), we list two examples of meta-paths in Figure 4.1 (b) and (c), where an arrow explicitly shows the direction of a relation. We say a path $p = (a_1 a_2 \ldots a_{l+1})$ between a_1 and a_{l+1} in network G follows the meta-path $\mathcal{P}$, if $\forall i, a_i \in A_i$ and each link $e_i = \langle a_i a_{i+1} \rangle$ belongs to each relation R_i in $\mathcal{P}$. We call these paths as *path instances* of $\mathcal{P}$, denoted as $p \in \mathcal{P}$. The examples of path instances have been shown in Table 4.2.

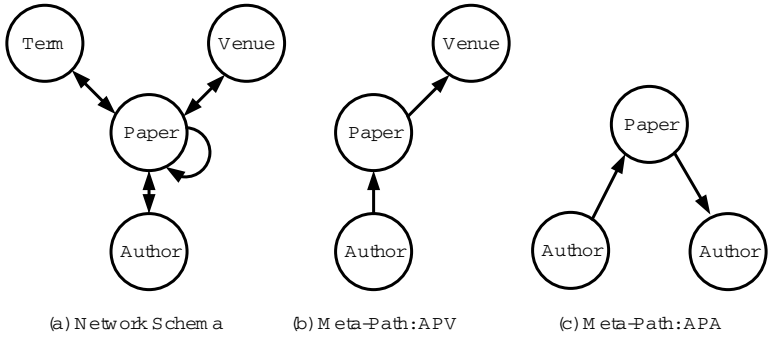


Figure 4.1: Bibliographic network schema and meta-paths.

In addition to pointing out the meta-path we are interested in, we also need to consider how to quantify the connection between two objects following a given meta-path. Analogously, a meta-path-based measure in an information network corresponds to a feature in a traditional data set, which can be used in many mining tasks.

4.2.2 META-PATH-BASED SIMILARITY FRAMEWORK

Given a user-specified meta-path, say $\mathcal{P} = (A_1 A_2 \ldots A_l)$, several similarity measures can be defined for a pair of objects $x \in A_1$ and $y \in A_l$, according to the path instances between them following the meta-path. We use $s(x, y)$ to denote the similarity between x and y, and list several straightforward measures in the following.

- Path count: the number of path instances p between x and y following $\mathcal{P}$: $s(x, y) = |\{p : p \in \mathcal{P}\}|$.
- Random walk: $s(x, y)$ is the probability of the random walk that starts form x and ends with y following meta-path $\mathcal{P}$, which is the sum of the probabilities of all the path instances $p \in \mathcal{P}$ starting with x and ending with y, denoted as $Prob(p)$: $s(x, y) = \sum_{p \in \mathcal{P}} Prob(p)$.
- Pairwise random walk: for a meta-path $\mathcal{P}$ that can be decomposed into two shorter meta-paths with the same length $\mathcal{P} = (\mathcal{P}_1 \mathcal{P}_2)$, $s(x, y)$ is then the pairwise random walk probability starting from objects x and y and reaching the same middle object: $s(x, y) = \sum_{(p_1 p_2) \in (\mathcal{P}_1 \mathcal{P}_2)} Prob(p_1) Prob(p_2^{-1})$, where $Prob(p_1)$ and $Prob(p_2^{-1})$ are random walk probabilities of the two path instances.

In general, we can define a meta-path-based similarity framework for two objects x and y as: $s(x, y) = \sum_{p \in \mathcal{P}} f(p)$, where $f(p)$ is a measure defined on the path instance p between x and y. Note that P-PageRank and SimRank, two well-known network similarity functions, are weighted combinations of random walk measure or pairwise random walk measure, respectively, over meta-paths with different lengths in homogeneous networks. In order to use P-PageRank and SimRank

in heterogeneous information networks, we need to specify the meta-path(s) we are interested in and restrict the random walk on the given meta-path(s).

4.2.3 PATHSIM: A NOVEL SIMILARITY MEASURE

Although there have been several similarity measures as presented above, they are biased to either highly visible objects or highly concentrated objects but cannot capture the semantics of peer similarity. For example, the path count and random walk-based similarity always favor objects with large degrees, and the pairwise random walk-based similarity favors concentrated objects where the majority of the links goes to a small portion of objects. However, in many scenarios, finding similar objects in networks is to *find similar peers*, such as finding similar authors based on their fields and reputation, finding similar actors based on their movie styles and productivity, and finding similar products based on their functions and popularity.

This motivated us to propose a new, meta-path-based similarity measure, called *PathSim*, that captures the subtlety of peer similarity. The intuition behind it is that two similar peer objects should not only be strongly connected, but also share comparable visibility. As the relation of peer should be symmetric, we confine PathSim to symmetric meta-paths. It is easy to see that, *round trip meta-paths* in the form of $\mathcal{P} = (\mathcal{P}_l \mathcal{P}_l^{-1})$ are always symmetric.

Definition 4.2 (PathSim: A meta-path-based similarity measure) Given a symmetric meta-path $\mathcal{P}$, PathSim between two objects x and y of the same type is:

$$s(x, y) = \frac{2 \times |\{p_{x \rightsquigarrow y} : p_{x \rightsquigarrow y} \in \mathcal{P}\}|}{|\{p_{x \rightsquigarrow x} : p_{x \rightsquigarrow x} \in \mathcal{P}\}| + |\{p_{y \rightsquigarrow y} : p_{y \rightsquigarrow y} \in \mathcal{P}\}|},$$

where $p_{x \rightsquigarrow y}$ is a path instance between x and y, $p_{x \rightsquigarrow x}$ is that between x and x, and $p_{y \rightsquigarrow y}$ is that between y and y.

This definition shows that given a meta-path $\mathcal{P}$, $s(x, y)$ is defined in terms of two parts: (1) their connectivity defined by the number of paths between them following $\mathcal{P}$; and (2) the balance of their visibility, where the visibility of an object according $\mathcal{P}$ is defined as the number of path instances between the object itself following $\mathcal{P}$. Note that we do count multiple occurrences of a path instance as the weight of the path instance, which is the product of weights of all the links in the path instance.

Table 4.3 presents in three measures the results of finding top-5 similar authors for "Anhai Doan," who is an established young researcher in the database field, under the meta-path $APVPA$ (based on their shared venues), in the database and information system (DBIS) area. P-PageRank returns the most similar authors as those published substantially in the area, that is, highly ranked authors; SimRank returns a set of authors that are concentrated on a small number of venues shared with Doan; whereas PathSim returns Patel, Deshpande, Yang and Miller, who share very similar publication records and are also rising stars in the database field as Doan. Obviously, PathSim captures desired semantic similarity as peers in such networks.

Table 4.3: Top-5 similar authors for "AnHai Doan" in the *DBIS* area

Rank	P-PageRank	SimRank	PathSim
1	AnHai Doan	AnHai Doan	AnHai Doan
2	Philip S. Yu	Douglas W. Cornell	Jignesh M. Patel
3	Jiawei Han	Adam Silberstein	Amol Deshpande
4	Hector Garcia-Molina	Samuel DeFazio	Jun Yang
5	Gerhard Weikum	Curt Ellmann	Renée J. Miller

The calculation of PathSim between any two objects of the same type given a certain meta-path involves matrix multiplication. Given a network $G = (\mathcal{V}, \mathcal{E})$ and its network schema T_G, we call the new adjacency matrix for a meta-path $\mathcal{P} = (A_1 A_2 \dots A_l)$ *a relation matrix*, and is defined as $M = W_{A_1A_2} W_{A_2A_3} \dots W_{A_{l-1}A_l}$, where $W_{A_iA_j}$ is the adjacency matrix between type A_i and type A_j. $M(i, j)$ represents the number of path instances between object $x_i \in A_1$ and object $y_j \in A_l$ under meta-path $\mathcal{P}$.

For example, relation matrix M for the meta-path $\mathcal{P} = (APA)$ is a co-author matrix, with each element representing the number of co-authored papers for the pair of authors. Given a symmetric meta-path $\mathcal{P}$, PathSim between two objects x_i and x_j of the same type can be calculated as $s(x_i, x_j) = \frac{2M_{ij}}{M_{ii}+M_{jj}}$, where M is the relation matrix for the meta-path $\mathcal{P}$, M_{ii} and M_{jj} are the visibility for x_i and x_j in the network given the meta-path.

It is easy to see that the relation matrix for the reverse meta-path of $\mathcal{P}_l$, which is $\mathcal{P}_l^{-1}$, is the *transpose* of relation matrix for $\mathcal{P}_l$. In this paper, we only consider the meta-path in the round trip form of $\mathcal{P} = (\mathcal{P}_l \mathcal{P}_l^{-1})$, to guarantee its symmetry and therefore the symmetry of the PathSim measure. By viewing PathSim in the meta-path-based similarity framework, $f(p) = 2\frac{w(a_1,a_2)\dots w(a_{l-1},a_l)}{M_{ii}+M_{jj}}$, for any path instance p starting from x_i and ending with x_j following the meta-path ($a_1 = x_i$ and $a_l = x_j$), where $w(a_m, a_n)$ is the weight for the link $\langle a_m, a_n \rangle$ defined in the adjacency matrix.

Some good properties of PathSim, such as symmetric, self-maximum, and balance of visibility, are shown in Theorem 4.3. For the balance property, we can see that the larger the difference of the visibility of the two objects, the smaller the upper bound for their PathSim similarity.

Theorem 4.3 (Properties of PathSim)

1. **Symmetric**. $s(x_i, x_j) = s(x_j, x_i)$.
2. **Self-maximum**. $s(x_i, x_j) \in [0, 1]$, and $s(x_i, x_i) = 1$.
3. **Balance of visibility**. $s(x_i, x_j) \leq \frac{2}{\sqrt{M_{ii}/M_{jj}}+\sqrt{M_{jj}/M_{ii}}}$.

Although using meta-path-based similarity we can define similarity between two objects given *any* round trip meta-paths, the following theorem tells us a *very long* meta-path is not very

meaningful. Indeed, due to the sparsity of real networks, objects that are similar may share no immediate neighbors, and longer meta-paths will propagate similarities to remote neighborhoods. For example, as in the DBLP example, if we consider the meta-path APA, only two authors that are co-authors have a non-zero similarity score; but if we consider longer meta-paths like $APVPA$ or $APTPA$, authors will be considered to be similar if they have published papers in a similar set of venues or sharing a similar set of terms no matter whether they have co-authored. But how far should we keep going? The following theorem tells us that a very long meta-path may be misleading. We now use $\mathcal{P}^k$ to denote a meta-path repeating k times of the basic meta-path pattern of $\mathcal{P}$, e.g., $(AVA)^2 = (AVAVA)$.

Theorem 4.4 (Limiting behavior of PathSim under infinity-length meta-path) Let meta-path $\mathcal{P}^{(k)} = (\mathcal{P}_l\mathcal{P}_l^{-1})^k$, $M_{\mathcal{P}}$ be the relation matrix for meta-path $\mathcal{P}_l$, and $M^{(k)} = (M_{\mathcal{P}}M_{\mathcal{P}}^T)^k$ be the relation matrix for $\mathcal{P}^{(k)}$, then by PathSim, the similarity between objects x_i and x_j as $k \to \infty$ is:

$$\lim_{k\to\infty} s^{(k)}(i,j) = \frac{2\mathbf{r}(i)\mathbf{r}(j)}{\mathbf{r}(i)\mathbf{r}(i) + \mathbf{r}(j)\mathbf{r}(j)} = \frac{2}{\frac{\mathbf{r}(i)}{\mathbf{r}(j)} + \frac{\mathbf{r}(j)}{\mathbf{r}(i)}},$$

where $\mathbf{r}$ is the primary eigenvector of M, and $\mathbf{r}(i)$ is the i_{th} item of $\mathbf{r}$.

As primary eigenvectors can be used as authority ranking of objects [66], the similarity between two objects under an infinite meta-path can be viewed as a measure defined on their rankings ($\mathbf{r}(i)$ is the ranking score for object x_i). Two objects with more similar ranking scores will have higher similarity (e.g., SIGMOD will be similar to AAAI). Later experiments (Table 4.9) will show that this similarity, with the meaning of global ranking, is not that useful. Note that, the convergence of PathSim with respect to path length is usually very fast and the length of 10 for networks of the scale of DBLP can almost achieve the effect of a meta-path with an infinite length. Therefore, in this paper, we only aim at solving the top-k similarity search problem for a *relatively short* meta-path.

Even for a relatively short length, it may still be inefficient in both time and space to materialize all the meta-paths. Thus, we propose in Section 4.3 materializing relation matrices for short length meta-paths, and concatenating them online to get longer ones for a given query.

4.3 ONLINE QUERY PROCESSING FOR SINGLE META-PATH

Compared with P-PageRank and SimRank, the calculation for PathSim is much more efficient, as it is a local graph measure. But it still involves expensive matrix multiplication operations for top-k search functions, as we need to calculate the similarity between a query and every object of the same type in the network. One possible solution is to materialize all the meta-paths within a given length. Unfortunately, it is time and space expensive to materialize all the possible meta-paths. For example, in the DBLP network, the similarity matrix corresponding to a length-4 meta-path,

$APVPA$, for identifying similar authors publishing in common venues is a $710K \times 710K$ matrix, whose non-empty elements reaches $5G$, and requires storage size more than $40GB$.

In order to support fast online query processing for large-scale networks, we propose a methodology that partially materializes short length meta-paths and then concatenates them online to derive longer meta-path-based similarity. First, a baseline method (*PathSim-baseline*) is proposed, which computes the similarity between query object x and all the candidate objects y of the same type. Next, a co-clustering based pruning method (*PathSim-pruning*) is proposed, which prunes candidate objects that are not promising according to their similarity upper bounds. Both algorithms return *exact* top-k results for the given query. Note that the same methodology can be adopted by other meta-path-based similarity measures, such as random walk and pairwise random walk, by taking a different definition of similarity matrix accordingly.

4.3.1 SINGLE META-PATH CONCATENATION

Given a meta-path $\mathcal{P} = (\mathcal{P}_l\mathcal{P}_l^{-1})$, where $\mathcal{P}_l = (A_1 \cdots A_l)$, the relation matrix for path $\mathcal{P}_l$ is $M_{\mathcal{P}} = W_{A_1A_2}W_{A_2A_3} \cdots W_{A_{l-1}A_l}$, then the relation matrix for path $\mathcal{P}$ is $M = M_{\mathcal{P}}M_{\mathcal{P}}^T$. Let n be the number of objects in A_1. For a query object $x_i \in A_1$, if we compute the top-k most similar objects $x_j \in A_1$ for x_i on-the-fly, without materializing any intermediate results, computing M from scratch would be very expensive. On the other hand, if we have pre-computed and stored the relation matrix $M = M_{\mathcal{P}}M_{\mathcal{P}}^T$, it would be a trivial problem to get the query results. We only need to locate the corresponding row in the matrix for the query x_i, re-scale it using $(M_{ii} + M_{jj})/2$, and finally sort the new vector and return the top-k objects. However, fully materializing the relation matrices for all possible meta-paths is also impractical, since the space complexity ($O(n^2)$) would prevent us from storing M for every meta-path. Instead of taking the above extremes, we partially materialize relation matrix $M_{\mathcal{P}}^T$ for meta-path $\mathcal{P}_l^{-1}$, and compute top-k results online by concatenating $\mathcal{P}_l$ and $\mathcal{P}_l^{-1}$ into $\mathcal{P}$ without full matrix multiplication.

We now examine the concatenation problem, that is, when the relation matrix M for the full meta-path $\mathcal{P}$ is not pre-computed and stored, but the relation matrix $M_{\mathcal{P}}^T$ corresponding to the partial meta-path $\mathcal{P}_l^{-1}$ is available. In this case, we assume the main diagonal of M, that is, $D = (M_{11}, \ldots, M_{nn})$, is pre-computed and stored. Since for $M_{ii} = M_{\mathcal{P}}(i,:)M_{\mathcal{P}}(i,:)^T$, the calculation only involves $M_{\mathcal{P}}(i,:)$ itself, and only $O(nd)$ in time and $O(n)$ in space are required, where d is the average number of non-zero elements in each row of $M_{\mathcal{P}}$ for each object.

In this study, we only consider concatenating the partial paths $\mathcal{P}_l$ and $\mathcal{P}_l^{-1}$ into the form $\mathcal{P} = \mathcal{P}_l\mathcal{P}_l^{-1}$ or $\mathcal{P} = \mathcal{P}_l^{-1}\mathcal{P}_l$. For example, given a pre-stored meta-path APV, we are able to answer queries for meta-paths $APVPA$ and $VPAPV$. For our DBLP network, to store relation matrix for partial meta-path APV only needs around $25M$ space, which is less than 0.1% of the space for materializing meta-path $APVPA$. Other concatenation forms that may lead to different optimization methods are also possible (e.g., concatenating several short meta-paths). In the following discussion, we focus on the algorithms using the concatenation form $\mathcal{P} = \mathcal{P}_l\mathcal{P}_l^{-1}$.

4.3.2 BASELINE

Suppose we know the relation matrix $M_{\mathcal{P}}$ for meta-path P_l, and the diagonal vector $D = (M_{ii})_{i=1}^{n}$, in order to get top-k objects $x_j \in A_1$ with the highest similarity for the query x_i, we need to compute $s(i, j)$ for all x_j. The straightforward baseline is: (1) first apply vector-matrix multiplication to get $M(i, :) = M_{\mathcal{P}}(i, :)M_{\mathcal{P}}^T$; (2) calculate $s(i, j) = \frac{2M(i,j)}{M(i,i)+M(j,j)}$ for all $x_j \in A_1$; and (3) sort $s(i, j)$ to return the top-k list in the final step. When n is very large, the vector-matrix computation will be too time consuming to check every possible object x_j. Therefore, we first select x_j's that are not orthogonal to x_i in the vector form, by following the links from x_i to find 2-step neighbors in relation matrix $M_{\mathcal{P}}$, that is, $x_j \in CandidateSet = \{\bigcup_{y_k \in M_{\mathcal{P}}.neighbors(x_i)} M_{\mathcal{P}}^T.neighbors(y_k)\}$, where $M_{\mathcal{P}}.neighbors(x_i) = \{y_k | M_{\mathcal{P}}(x_i, y_k) \neq 0\}$, which can be easily obtained in the sparse matrix form of $M_{\mathcal{P}}$ that indexes both rows and columns. This will be much more efficient than pairwise comparison between the query and all the objects of that type. We call this baseline concatenation algorithm as *PathSim-baseline*.

The *PathSim-baseline* algorithm, however, is still time consuming if the candidate set is large. Although $M_{\mathcal{P}}$ can be relatively sparse given a short length meta-path, after concatenation, M could be dense, i.e., the $CandidateSet$ could be very large. Still, considering the query object and one candidate object represented by query vector and candidate vector, the dot product between them is proportional to the size of their non-zero elements. The time complexity for computing PathSim for each candidate is $O(d)$ on average and $O(m)$ in the worst case, that is, $O(nm)$ in the worst case for all the candidates, where n is the row size of $M_{\mathcal{P}}$ (i.e., the number of objects in type A_1), m the column size of $M_{\mathcal{P}}$ (i.e., the number of objects in type A_l), and d the average non-zero element for each object in $M_{\mathcal{P}}$. We now propose a co-clustering based top-k concatenation algorithm, by which non-promising target objects are dynamically filtered out to reduce the search space.

4.3.3 CO-CLUSTERING-BASED PRUNING

In the baseline algorithm, the computational costs involve two factors. First, the more candidates to check, the more time the algorithm will take; second, for each candidate, the dot product of query vector and candidate vector will at most involve m operations, where m is the vector length. The intuition to speed up the search is to prune unpromising candidate objects using simpler calculations. Based on the intuition, we propose a co-clustering-based (i.e., clustering rows and columns of a matrix simultaneously) path concatenation method, which first generates co-clusters of two types of objects for partial relation matrix, then stores necessary statistics for each of the blocks corresponding to different co-cluster pairs, and then uses the block statistics to prune the search space. For better illustration, we call clusters of type A_1 as **target clusters**, since the objects in A_1 are the targets for the query; and call clusters of type A_l as **feature clusters**, since the objects in A_l serve as features to calculate the similarity between the query and the target objects. By partitioning A_1 into different target clusters, if a whole target cluster is not similar to the query, then all the objects in the target cluster are likely not in the final top-k lists and can be pruned. By partitioning A_l into different feature clusters, cheaper calculations on the dimension-reduced query vector and candidate vectors

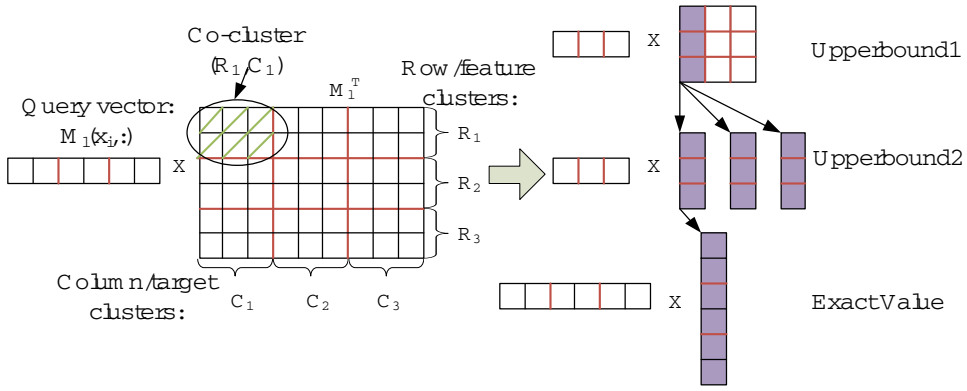


Figure 4.2: Illustration of pruning strategy. Given the partial relation matrix M_l^T and its 3×3 co-clusters, and the query vector $M_l(x_i, :)$ for query object x_i, first the query vector is compressed into the aggregated query vector with the length of 3, and the upper bounds of the similarity between the query and all the 3 target clusters are calculated based on the aggregated query vector and aggregated cluster vectors; second, for each of the target clusters, if they cannot be pruned, calculate the upper bound of the similarity between the query and each of the 3 candidates within the cluster using aggregated vectors; third, if the candidates cannot be pruned, calculate the exact similarity value using the non-aggregated query vector and candidate vectors.

can be used to derive the similarity upper bounds. This pruning idea is illustrated in Figure 4.2 using a toy example with 9 target objects and 6 feature objects. The readers may refer to the PathSim paper [65] for the concrete formulas of the upper bounds and their derivations.

Experiments show that *PathSim-Pruning* can significantly improve the query processing speed comparing with the baseline algorithm, without affecting the search quality.

4.4 MULTIPLE META-PATHS COMBINATION

In Section 4.3, we presented algorithms for similarity search using single meta-path. Now, we present a solution to combine multiple meta-paths. Formally, given r round trip meta-paths from Type A back to Type A, $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$, and their corresponding relation matrix $M_1, M_2, \ldots, M_r$, with weights $w_1, w_2, \ldots, w_r$ specified by users, the combined similarity between objects $x_i, x_j \in A$ are defined as: $s(x_i, x_j) = \sum_{l=1}^{r} w_l s_l(x_i, x_j)$, where $s_l(x_i, x_j) = \frac{2M_l(i,j)}{M_l(i,i)+M_l(j,j)}$.

Example 4.5 (Multiple meta-paths combination for venue similarity search) Following the motivating example in the introduction section, Table 4.4 shows the results of combining two meta-paths $\mathcal{P}_1 = VPAPV$ and $\mathcal{P}_2 = VPTPV$ with different weights specified by w_1 and w_2, for query "DASFAA."

Table 4.4: Top-5 similar venues to "DASFAA" using multiple meta-paths

Rank	$w_1 = 0.2, w_2 = 0.8$	$w_1 = 0.5, w_2 = 0.5$	$w_1 = 0.8, w_2 = 0.2$
1	DASFAA	DASFAA	DASFAA
2	Data Knowl. Eng.	DEXA	DEXA
3	CIKM	CIKM	WAIM
4	EDBT	Data Knowl. Eng.	CIKM
5	Inf. Syst.	EDBT	APWeb

The reason why we need to combine several meta-paths is that each meta-path provides a unique angle (or a unique feature space) to view the similarity between objects, and the ground truth may be a cause of different factors. Some useful guidance of the weight assignment includes: longer meta-path utilize more remote relationships and thus should be assigned with a smaller weight, such as in P-PageRank and SimRank; and, meta-paths with more important relationships should be assigned with a higher weight. For automatically determining the weights, users could provide training examples of similar objects to learn the weights of different meta-paths using learning algorithms.

We now evaluate the quality of similarity measure generated by combined meta-paths, according to their performance for clustering tasks in the "*four-area*" dataset. First, two meta-paths for the venue type, namely, VAV and VTV (short for $VPAPV$ and $VPTPV$), are selected and their linear combinations with different weights are considered. Second, two meta-paths with the same basic path but different lengths, namely AVA and $(AVA)^2$, are selected and their linear combinations with different weights are considered. The clustering accuracy measured by NMI for conferences and authors is shown in Table 4.5, which shows that the combination of multiple meta-paths can produce better similarity than the single meta-path in terms of clustering accuracy.

Table 4.5: Clustering accuracy for PathSim for meta-path combinations on the "*four-area*" dataset

w_1	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**1**
w_2	**1**	**0.8**	**0.6**	**0.4**	**0.2**	**0**
$VAV; VTV$	0.7917	0.7936	**0.8299**	**0.8587**	0.8123	0.8116
$AVA; (AVA)^2$	0.6091	0.6219	0.6506	**0.6561**	**0.6508**	0.6501

4.5 EXPERIMENTAL RESULTS

To show the effectiveness of the PathSim measure and the efficiency of the proposed algorithms, we use the bibliographic networks extracted from DBLP and Flickr in the experiments.

We use the DBLP dataset downloaded in November 2009 as the main test dataset. It contains over $710K$ authors, $1.2M$ papers, and $5K$ venues (conferences/journals). After removing stopwords

Table 4.6: Case study of five similarity measures on query "PKDD" on the $DBIS$ dataset

Rank	P-PageRank	SimRank	RW	PRW	PathSim
1	PKDD	PKDD	PKDD	PKDD	PKDD
2	KDD	Local Pattern Detection	KDD	Local Pattern Detection	ICDM
3	ICDE	KDID	ICDM	DB Support for DM Appl.	SDM
4	VLDB	KDD	PAKDD	Constr. Min. & Induc. DB	PAKDD
5	SIGMOD	Large-Scale Paral. DM	SDM	KDID	KDD
6	ICDM	SDM	TKDE	MCD	DMKD
7	TKDE	ICDM	SIGKDD Expl.	Pattern Detection & Disc.	SIGKDD Expl.
8	PAKDD	SIGKDD Expl.	ICDE	RSKD	Knowl. Inf. Syst.
9	SIGIR	Constr. Min. & Induc. DB	SEBD	WImBI	JIIS
10	CIKM	TKDD	CIKM	Large-Scale Paral. DM	KDID

in paper titles, we get around $70K$ terms appearing more than once. This dataset is referred as the *full-DBLP* dataset. Two small subsets of the data (to alleviate the high computational costs of P-PageRank and SimRank) are used for the comparison with other similarity measures in effectiveness: (1) the *DBIS* dataset, which contains all the 464 venues and top-5000 authors from the database and information system area; and (2) the *four-area* dataset, which contains 20 venues and top-5000 authors from 4 areas: *database, data mining, information retrieval,* and *artificial intelligence* [64], and cluster labels are given for all the 20 venues and a subset of 1713 authors.

For additional case studies, we construct a Flickr network from a subset of the Flickr data, which contains four types of objects: images, users, tags, and groups. Links exist between images and users, images and tags, and images and groups. We use 10,000 images from 20 groups as well as their related 664 users and 10,284 tags appearing more than once to construct the network.

4.5.1 EFFECTIVENESS

Comparing* PathSim *with other measures When a meta-path $\mathcal{P} = (\mathcal{P}_l \mathcal{P}_l^{-1})$ is given, other measures such as random walk (RW) and pairwise random walk (PRW) can be applied to the same meta-path, and P-PageRank and SimRank can be applied to the sub-network extracted from $\mathcal{P}$. For example, for the meta-path $VPAPV$ (VAV in short) for finding venues sharing the same set of authors, the bipartite graph M_{CA}, derived from the relation matrix corresponding to VPA can be used in both P-PageRank and SimRank algorithms. In our experiments, the damping factor for P-PageRank is set as 0.9 and that for SimRank is 0.8.

First, a case study is shown in Table 4.6, which is applied to the $DBIS$ dataset, under the meta-path VAV. One can see that for query "PKDD" (short for "Principles and Practice of Knowledge Discovery in Databases," a European data mining conference), P-PageRank favors the venues with higher visibility, such as KDD and several well-known venues; SimRank prefers more concentrated venues (i.e., a large portion of publications goes to a small set of authors) and returns many not well-known venues such as "Local Pattern Detection" and KDID; RW also favors highly visible objects such as KDD, but brings in fewer irrelevant venues due to that it utilizes merely one short

Table 4.7: Comparing the accuracy of top-15 query results for five similarity measures on the *DBIS* dataset measured by nDCG

	P-PageRank	SimRank	RW	PRW	PathSim
Accuracy	0.5552	0.6289	0.7061	0.5284	**0.7446**

meta-path; PRW performs similar to SimRank, but brings in more not so well-known venues due to the short meta-path it uses; whereas PathSim returns the venues in both the area and the reputation similar to PKDD, such as ICDM and SDM.

We then labeled top-15 results for 15 queries from the venues in the *DBIS* dataset (i.e., SIGMOD, VLDB, ICDE, PODS, EDBT, DASFAA, KDD, ICDM, PKDD, SDM, PAKDD, WWW, SIGIR, TREC, and APWeb), to test the quality of the ranking lists given by 5 measures. We label each result object with a relevance score at one of the three levels: 0 (non-relevant), 1 (somewhat relevant), and 2 (very relevant). Then we use the measure nDCG (i.e., Normalized Discounted Cumulative Gain, with the value between 0 and 1, the higher the better) [27] to evaluate the quality of a ranking algorithm by comparing its output ranking results with the labeled ones (Table 4.7). The results show that PathSim gives the best ranking quality in terms of human intuition, which is consistent with the previous case study.

Table 4.8: Top-10 similar authors to "Christos Faloutsos" under different meta-paths on the *full-DBLP* dataset

Rank	Author
1	Christos Faloutsos
2	Spiros Papadimitriou
3	Jimeng Sun
4	Jia-Yu Pan
5	Agma J. M. Traina
6	Jure Leskovec
7	Caetano Traina Jr.
8	Hanghang Tong
9	Deepayan Chakrabarti
10	Flip Korn

(a) Path: *APA*

Rank	Author
1	Christos Faloutsos
2	Jiawei Han
3	Rakesh Agrawal
4	Jian Pei
5	Charu C. Aggarwal
6	H. V. Jagadish
7	Raghu Ramakrishnan
8	Nick Koudas
9	Surajit Chaudhuri
10	Divesh Srivastava

(b) Path: *APVPA*

Semantic meanings of different meta-paths As we pointed out, different meta-paths give different semantic meanings, which is one of the reasons that similarity definitions in homogeneous networks cannot be applied directly to heterogeneous networks. Besides the motivating example

in the introduction section, Table 4.8 shows the author similarity under two scenarios for author Christos Faloutsos: *co-authoring papers* and *publishing papers in the same venues*, represented by the meta-paths APA and $APVPA$, respectively. One can see that the first path returns co-authors who have strongest connections with Faloutsos (i.e., students and close collaborators) in DBLP, whereas $APVPA$ returns those publishing papers in the most similar venues.

Table 4.9: Top-10 similar venues to "SIGMOD" under meta-paths with different lengths on the *full-DBLP* dataset

Rank	Venue	Score	Rank	Venue	Score	Rank	Venue	Score
1	SIGMOD	1	1	SIGMOD	1	1	SIGMOD	1
2	VLDB	0.981	2	VLDB	0.997	2	AAAI	0.9999
3	ICDE	0.949	3	ICDE	0.996	3	ESA	0.9999
4	TKDE	0.650	4	TKDE	0.787	4	ITC	0.9999
5	SIGMOD Record	0.630	5	SIGMOD Record	0.686	5	STACS	0.9997
6	IEEE Data Eng. Bul.	0.530	6	PODS	0.586	6	PODC	0.9996
7	PODS	0.467	7	KDD	0.553	7	NIPS	0.9993
8	ACM Trans. DB Sys.	0.429	8	CIKM	0.540	8	Comput. Geom.	0.9992
9	EDBT	0.420	9	IEEE Data Eng. Bul.	0.532	9	ICC	0.9991
10	CIKM	0.410	10	J. Comp. Sys. Sci.	0.463	10	ICDE	0.9984
(a) Path: $(VPAPV)^2$			(b) Path: $(VPAPV)^4$			(c) Path: $(VPAPV)^\infty$		

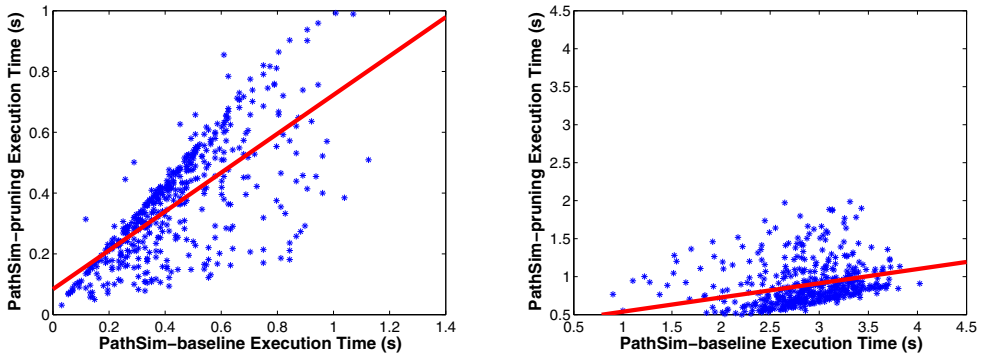
The impact of path length The next interesting question is how the length of meta-path impacts the similarity definition. Table 4.9 shows an example of venues similar to "SIGMOD" with three meta-paths, using exactly the same basic meta-path, but with different repeating times. These meta-paths are $(VPAPV)^2$, $(VPAPV)^4$, and its infinity form (global ranking-based similarity). Note that in $(VPAPV)^2$, two venues are similar if they share many similar authors who publish papers in *the same* venues; while in $(VPAPV)^4$, the similarity definition of those venues will be further relaxed, namely, two venues are similar if they share many similar authors who publish papers in *similar* venues. Since venue type only contains $5K$ venues, we are able to get the full materialization relation matrix for $(VPAPV)^2$. $(VPAPV)^4$ is obtained using meta-path concatenation from $(VPAPV)^2$. The results are summarized in Table 4.9, where longer paths gradually bring in more remote neighbors, with higher similarity scores, and finally, it degenerates into global ranking comparison. Through this study, one can see that a meta-path with relatively short length is good enough to measure similarity, where a long meta-path may even reduce the quality.

4.5.2 EFFICIENCY COMPARISON

The time complexity for SimRank is $O(KN^2d^2)$, where K is the number of iterations, N is the total number of objects, and d is the average neighbor size; the time complexity for calculating P-PageRank for one query is $O(KNd)$, where K, N, d has the same meaning as in SimRank, whereas

the time complexity for PathSim using $PathSim\text{-}baseline$ for single query is $O(nd)$, where $n < N$ is the number of objects in the target type, d is the average degree of objects in target type for partial relation matrix $M_{\mathcal{P}_l}$. The time complexity for RW and PRW are the same as PathSim. We can see that similarity measure only using one meta-path is much more efficient than those also using longer meta-paths in the network (e.g., SimRank and P-PageRank).

Two algorithms, *PathSim-baseline* and *PathSim-pruning*, introduced in Section 4.3, are compared, for efficiency study under different meta-paths, namely, $VPAPV$ and $(VPAPV)^2$ (denoted as VAV and $VAVAV$ for short). The results show that the denser the relation matrix corresponding to the partial meta-path (M_{VPAPV} in comparison with M_{VPA}), the greater the pruning power. The improvement rates are 18.23% and 68.04% for the 2 meta-paths.



(a) Meta-path: VAV

(b) Meta-path: $VAVAV$

Figure 4.3: Pruning power denoted by the slope of the fitting line under two meta-paths for type conference on the *full-DBLP* dataset. Each dot represents a query under the indicated meta-path.

4.5.3 CASE-STUDY ON FLICKR NETWORK

In this case study, we show that to retrieve similar images for a query image one can explore links in the network rather than the content information. Let "I" represent images, "T" tags that associated with each image, and "G" groups that each image belongs to. Two meta-paths are used and compared. One is ITI, which means common tags are used by two images at evaluation of their similarity. The results are shown in Figure 4.4. The other is $ITIGITI$, which means tags similarities are further measured by their shared groups, and two images can be similar even if they do not share many exact same tags as long as these tags are used by many images of the same groups. One can see that the second meta-path gives better results than the first, as shown in Figure 4.5, where the first image

is the input query. This is likely due to that the latter meta-path provides additional information related to image groups, and thus improves the similarity measure between images.

(a) top-1 (b) top-2 (c) top-3 (d) top-4 (e) top-5 (f) top-6

Figure 4.4: Top-6 images in Flickr network under meta-path ITI.

(a) top-1 (b) top-2 (c) top-3 (d) top-4 (e) top-5 (f) top-6

Figure 4.5: Top-6 images in Flickr network under meta-path $ITIGITI$.

Authors' Biographies

YIZHOU SUN

Yizhou Sun received her Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 2012. She will be an assistant professor in the College of Computer and Information Science at Northeastern University. Her principal research interest is in mining information and social networks, and more generally in data mining, database systems, statistics, machine learning, information retrieval, and network science, with a focus on modeling novel problems and proposing scalable algorithms for large-scale, real-world applications. Yizhou has over 30 publications in books, journals, and major conferences. Tutorials based on her thesis work on mining heterogeneous information networks have been given in several premier conferences, including EDBT 2009, SIGMOD 2010, KDD 2010, ICDE 2012, VLDB 2012, and ASONAM 2012. She received ACM KDD 2012 Best Student Paper Award.

JIAWEI HAN

Jiawei Han is the Abel Bliss Professor of Computer Science at the University of Illinois at Urbana-Champaign. His research includes data mining, information network analysis, database systems, and data warehousing, with over 600 journal and conference publications. He has chaired or served on many program committees of international conferences, including PC co-chair for KDD, SDM, and ICDM conferences, and Americas Coordinator for VLDB conferences. He also served as the founding Editor-In-Chief of ACM Transactions on Knowledge Discovery from Data and is serving as the Director of Information Network Academic Research Center supported by U.S. Army Research Lab. He is Fellow of ACM and Fellow of IEEE, and received 2004 ACM SIGKDD Innovations Award, 2005 IEEE Computer Society Technical Achievement Award, 2009 IEEE Computer Society Wallace McDowell Award, and 2011 Daniel C. Drucker Eminent Faculty Award at UIUC. His book, *Data Mining: Concepts and Techniques*, has been used popularly as a textbook worldwide.