Chapter 3
Relevance Measure of Heterogeneous Objects

Abstract Similarity search is an important function in many applications, which usually focuses on measuring the similarity between objects with the same type. However, in many scenarios, we need to measure the relatedness between objects with different types. With the surge of study on heterogeneous networks, the relevance measure on objects with different types becomes increasingly important. In this chapter, we study the relevance search problem in heterogeneous networks, where the task is to measure the relatedness of heterogeneous objects (including objects with the same type or different types). And then, we introduce a novel measure HeteSim and its extended version.

3.1 HeteSim: A Uniform and Symmetric Relevance Measure

3.1.1 Overview

Similarity search is an important task in a wide range of applications, such as Web search [15] and product recommendations [11]. The key of similarity search is similarity measure, which evaluates the similarity of object pairs. Similarity measure has been extensively studied for traditional categorical and numerical data types, such as Jaccard coefficient and cosine similarity. There are also a few studies on leveraging link information in networks to measure the node similarity, such as Personalized PageRank [7], SimRank [6], and PathSim [21]. Conventional study on the similarity measure focuses on objects with the same type. That is, the objects being measured are of the same type, such as "document-to-document" and "Webpage-to-Webpage." There are very few studies on similarity measure on objects with different types. That is, the objects being measured are of different types, such as "author-to-conference" and "user-to-movie." It is reasonable. The similarity of objects with different types is a little against our common sense. Moreover, different from the similarity of objects with the same type, which can be measured on homogeneous situation (e.g., the same feature space or homogeneous link structure), it is even harder to define the similarity of objects with different types.

C. Shi and P.S. Yu, *Heterogeneous Information Network Analysis and Applications*, Data Analytics, DOI 10.1007/978-3-319-56212-4_3

However, the similarity of objects with different types is not only meaningful but also useful in some scenarios. For example, Prof. Jiawei Han is more relevant to KDD than IJCAI. Moreover, the similarity measure of objects with different types is needed in many applications. For example, in a recommended system, we need to know the relatedness between users and items to make accurate recommendations [5]. In an automatic profile extraction application, we need to measure the relatedness of objects with different types, such as authors and conferences, and conferences and organizations. Particularly, with the advent of study on heterogeneous information networks [20, 21], it is not only increasingly important but also feasible to study the relatedness among objects with different types. Heterogeneous information networks are the logical networks involving multiple-typed objects and multiple-typed links denoting different relations [4]. It is clear that heterogeneous information networks are ubiquitous and form a critical component of modern information infrastructure [4]. So it is essential to provide a relevance search function on objects with different types in such networks, which is the base of many applications. Since objects with different types coexist in the same network, their relevance measure is possible through link structure.

In this chapter, we study the relevance search problem in heterogeneous information networks. The aim of relevance search is to effectively measure the relatedness of heterogeneous objects (including objects with the same type or different types). Different from the similarity search which measures only the similarity of objects with the same type, the relevance search measures the relatedness among heterogeneous objects and it is not limited to objects with the same type. Distinct from relational retrieval [13, 23] in information retrieval domain, here relevance search is done on heterogeneous networks which can be constructed from metadata of objects. Moreover, we think that a desirable relevance measure should satisfy the symmetry property based on the following reasons: (1) The symmetric measure is more general and useful in many learning tasks. Although the symmetry property is not necessary in the query task, it is essential for many important tasks, such as clustering and collaborative filtering. Moreover, it is the necessary condition for a metric. (2) The symmetric measure makes more sense in many applications, especially for the relatedness of heterogeneous object pairs. For example, in some applications, we need to answer the question like who has similar importance to the SIGIR conference as Jiawei Han to KDD. Through comparing the relatedness of object pairs, we can deduce the information of their relative importance. However, it can only be done by the symmetric measure, not the asymmetric measure.

Inspired by the intuition that two objects are related if they are referenced by related objects, we propose a general framework, called HeteSim, to evaluate the relatedness of heterogeneous objects in heterogeneous networks. HeteSim is a path-based relevance measure, which can effectively capture the subtle semantics of search paths. Based on pairwise random walk model, HeteSim treats arbitrary search paths in a uniform way, which guarantees the symmetric property of HeteSim. An additional benefit is that HeteSim can evaluate the relatedness of objects with the same- or different types in the same way. Moreover, HeteSim is a semi-metric measure. In other words, HeteSim satisfies the properties of nonnegativity, identity of indiscernibles,

and symmetry. It implies that HeteSim can be used in many learning tasks (e.g., clustering and collaborative filtering). We also consider the computation issue of HeteSim and propose four fast computation strategies.

3.1.2 The HeteSim Measure

In many domains, similar objects are more likely to be related to some other similar objects. For example, similar researchers usually publish many similar papers, and similar customers purchase similar commodities. As a consequence, two objects are similar if they are referenced by similar objects. This intuition is also fit for heterogeneous objects. For example, a researcher is more relevant to the conferences that the researcher has published papers in, and a customer is more faithful to the brands that the customer usually purchases. Although the similar idea has been applied in SimRank [6], it is limited to homogeneous networks. When we apply the idea to heterogeneous networks, it faces the following challenges: (1) The relatedness of heterogeneous objects is path-constrained. The meta path not only captures the semantics information but also constrains the walk path. So we need to design a path-based similarity measure. (2) A uniform and symmetric measure should be designed for arbitrary paths. For a given path (symmetric or asymmetric), the measure can evaluate the relatedness of heterogeneous object pair (same or different types) with one single score. In the following section, we will illustrate these challenges and their solutions in detail.

3.1.2.1 Path-Based Relevance Measure

Different from homogeneous networks, the paths in heterogeneous networks have semantics, which makes the relatedness of object pair depend on the given meta path. Following the basic idea that similar objects are related to similar objects, we propose a path-based relevance measure: HeteSim.

Definition 3.1 (*HeteSim*) Given a meta path $P = R_1 \circ R_2 \circ \cdots \circ R_l$, the HeteSim score between two objects s and t ($s \in R_1.S$ and $t \in R_l.T$) is:

$$HeteSim(s, t|R_1 \circ R_2 \circ \cdots \circ R_l) = \frac{1}{|O(s|R_1)||I(t|R_l)|} \sum_{i=1}^{|O(s|R_1)|} \sum_{j=1}^{|I(t|R_l)|} HeteSim(O_i(s|R_1), I_j(t|R_l)|R_2 \circ \cdots \circ R_{l-1}) \quad (3.1)$$

where $O(s|R_1)$ is the out-neighbors of s based on relation R_1, and $I(t|R_l)$ is the in-neighbors of t based on relation R_l.

When s does not have any out-neighbors (i.e., $O(s|R_1) = \emptyset$) or t does not have any in-neighbors (i.e., $I(t|R_l) = \emptyset$) following the path, we have no way to infer any relatedness between s and t in this case, so we define their relevance score to be 0. Particularly, we consider objects with the same type to have **self-relation** (denoted as I relation), and each object only has self-relation with itself. It is obvious that an object is just similar to itself for I relation. So its relevance measure can be defined as follows:

Definition 3.2 (*HeteSim based on self-relation*) The HeteSim score between two same-typed objects s and t based on the self-relation I is:

$$HeteSim(s, t|I) = \delta(s, t) \tag{3.2}$$

where $\delta(s, t) = 1$, if s and t are same, or else $\delta(s, t) = 0$.

Equation 3.1 shows that the computation of $HeteSim(s, t|P)$ needs to iterate over all pairs $(O_i(s|R_1), I_j(t|R_l))$ of (s, t) along the path (s along the path and t against path), and sum up the relatedness of these pairs. Then, we normalize it by the total number of out-neighbors of s and in-neighbors of t. That is, the relatedness between s and t is the average relatedness between the out-neighbors of s and the in-neighbors of t. The process continues until s and t meet along the path. Similar to SimRank [6], HeteSim is also based on pairwise random walk, while it considers the path constraint. As we know, SimRank measures how soon two random surfers are expected to meet at the same node [6]. By contrast, $HeteSim(s, t|P)$ measures how likely s and t will meet at the same node when s follows along the path and t goes against the path.

3.1.2.2 Decomposition of Meta Path

Unfortunately, the source object s and the target object t may not meet along a given path P. For the similarity measure of same-typed objects, the meta paths are usually even-length, even symmetric, so the source object and the target object will meet at the middle objects. However, for the relevance measure of different-typed objects, the meta paths are usually odd-length. In this condition, the source and target objects will never meet at the same objects. Taking the $APVC$ path as an example, authors along the path and conferences against the path will never meet in the same objects. So the original HeteSim is not suitable for odd-length meta paths. In order to solve this difficulty, a basic idea is to transform odd-length paths into even-length paths, and thus, the source and target objects are always able to meet at the same objects. As a consequence, an arbitrary path can be decomposed as two equal-length paths.

When the length l of a meta path $P = (A_1A_2 \cdots A_{l+1})$ is even, the source objects (along the path) and the target objects (against the path) will meet in the **middle type** object $M = A_{\frac{l}{2}+1}$ on the **middle position** $mid = \frac{l}{2} + 1$, so the meta path P

can be divided into two equal-length path P_L and P_R. That is, $P = P_L P_R$, where $P_L = A_1 A_2 \cdots A_{mid-1} M$ and $P_R = M A_{mid+1} \cdots A_{l+1}$.

When the path length l is odd, the source objects and the target objects will meet at the relation $A_{\frac{l+1}{2}} A_{\frac{l+1}{2}+1}$. In order to let the source and target objects meet at same-typed objects, we can add a middle type object E between the atomic relation $A_{\frac{l+1}{2}} A_{\frac{l+1}{2}+1}$ and maintain the relation between $A_{\frac{l+1}{2}}$ and $A_{\frac{l+1}{2}+1}$ at the same time. Then, the new path becomes $\mathtt{P}' = (A_1 \cdots E \cdots A_{l+1})$ whose length is $l+1$, an even number. The source objects and the target objects will meet in the **middle type** object $M = E$ on the **middle position** $mid = \frac{l+1}{2} + 1$. As a consequence, the new relevance path $\mathtt{P}'$ can also be decomposed into two equal-length paths P_L and P_R.

Definition 3.3 (*Decomposition of meta path*) An arbitrary meta path $P = (A_1 A_2 \cdots A_{l+1})$ can be decomposed into two equal-length path P_L and P_R (i.e., $P = P_L P_R$), where $P_L = A_1 A_2 \cdots A_{mid-1} M$ and $P_R = M A_{mid+1} \cdots A_{l+1}$. M and mid are defined as above.

Obviously, for a symmetric path, $P = P_L P_R$, P_R^{-1} is equal to P_L. For example, the meta path $P = APCPA$ can be decomposed as $P_L = APC$ and $P_R = CPA$. For the meta path $APSPVC$, we can add a middle type object E in SP, and thus, the path becomes $APSEPVC$, so $P_L = APSE$ and $P_R = EPVC$.

The next question is how we can add the middle type object E in an atomic relation R between $A_{\frac{l+1}{2}}$ and $A_{\frac{l+1}{2}+1}$. In order to contain original atomic relation, we need to make the R relation be the composition of two new relations. To do so, for each instance of relation R, we can add an instance of E to connect the source and target objects of the relation instance. An example is shown in Fig. 3.1a, where the middle type object E is added in between the atomic relation AB along each path instance.

Definition 3.4 (*Decomposition of atomic relation*) For an atomic relation R, we can add an object type E (called edge object) between the $R.S$ and $R.T$ ($R.S$ and $R.T$ are the source and target object type of the relation R). And thus the atomic relation R is decomposed as R_O and R_I where R_O represents the relation between $R.S$ and E and R_I represents that between E and $R.T$. For each relation instance $r \in R$, an instance $e \in E$ connects $r.S$ and $r.T$. The paths $r.S \to e$ and $e \to r.T$ are the instances of R_O and R_I, respectively.

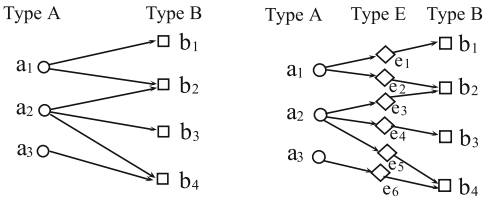
It is clear that the relation decomposition has the following property, whose proof can be found in [18].

Property 3.1 *An atomic relation R can be decomposed as R_O and R_I, $R = R_O \circ R_I$, and this decomposition is unique.*

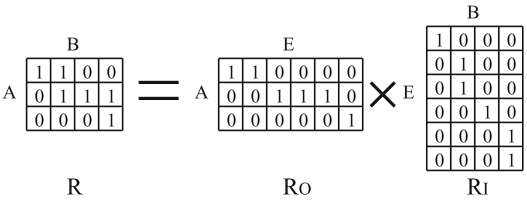
Based on this decomposition, the relatedness of two objects with an atomic relation R can be calculated as follows:

Definition 3.5 (*HeteSim based on atomic relation*) The HeteSim score between two different-typed objects s and t based on an atomic relation R ($s \in R.S$ and $t \in R.T$) is:

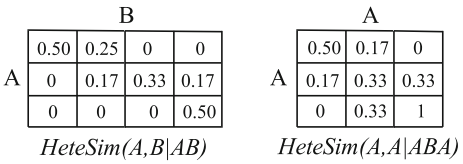
Fig. 3.1 Decomposition of atomic relation and its HeteSim calculation



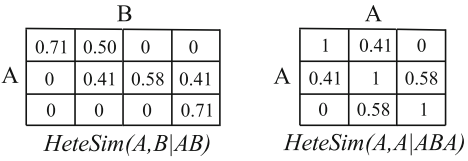
(a) Add middle type object



(b) Decomposition of atomic relation



(c) HeteSim scores before normalization



(d) HeteSim scores after normalization

$$HeteSim(s,t|R) = HeteSim(s,t|R_O \circ R_I) = \frac{1}{|O(s|R_O)||I(t|R_I)|} \sum_{i=1}^{|O(s|R_O)|} \sum_{j=1}^{|I(t|R_I)|} \delta(O_i(s|R_O), I_j(t|R_I)) \tag{3.3}$$

It is easy to find that $HeteSim(s,t|I)$ is a special case of $HeteSim(s,t|R)$, since, for the self-relation I, $I = I_O \circ I_I$ and $|O(s|I_O)| = |I(t|I_I)| = 1$. Definition 3.5 means that HeteSim can measure the relatedness of two different-typed objects with an atomic relation R directly through calculating the average of their mutual influence.

Example 3.1 Figure 3.1a shows an example of decomposition of atomic relation. The relation AB is decomposed into the relations AE and EB. Moreover, the relation AB is the composition of AE and EB as shown in Fig. 3.1b. Two HeteSim examples are illustrated in Fig. 3.1c. We can find that HeteSim justly reflects relatedness of

objects. Taking a_2 as example, although a_2 equally connects with b_2, b_3, and b_4, it is more close to b_3, because b_3 only connects with a_2. This information is correctly reflected in the HeteSim score of a_2 based on AB path.

We also find that the similarity of an object and itself is not 1 in HeteSim. Taking the right figure of Fig. 3.1c as example, the relatedness of a_2 and itself is 0.33. It is obviously unreasonable. In the following section, we will normalize the HeteSim and make the relevance measure more reasonable.

3.1.2.3 Normalization of HeteSim

Firstly, we introduce the calculation of HeteSim between any two objects given an arbitrary meta path.

Definition 3.6 (*Transition probability matrix*) For relation $A \xrightarrow{R} B$, W_{AB} is an adjacent matrix between type A and B. U_{AB} is a normalized matrix of W_{AB} along the row vector, which is the transition probability matrix of $A \longrightarrow B$ based on relation R. V_{AB} is a normalized matrix of W_{AB} along the column vector, which is the transition probability matrix of $B \longrightarrow A$ based on relation R^{-1}.

It is easy to prove that the transition probability matrix has the following property. The proof can be found in [18].

Property 3.2 *$U_{AB} = V'_{BA}$ and $V_{AB} = U'_{BA}$, where V'_{BA} is the transpose of V_{BA}.*

Definition 3.7 (*Reachable probability matrix*) Given a network $G = (V, E)$ following a network schema $S = (\mathtt{A}, \mathtt{R})$, a reachable probability matrix PM for a path $P = (A_1 A_2 \cdots A_{l+1})$ is defined as $PM_P = U_{A_1A_2} U_{A_2A_3} \cdots U_{A_lA_{l+1}}$ (PM for simplicity). $PM(i, j)$ represents the probability of object $i \in A_1$ reaching object $j \in A_{l+1}$ under the path P.

According to the definition and Property 3.2 of HeteSim, the relevance between objects in A_1 and A_{l+1} based on the meta path $P = A_1 A_2 \cdots A_{l+1}$ is

$$\begin{aligned} &HeteSim(A_1, A_{l+1}|P) = HeteSim(A_1, A_{l+1}|P_L P_R) \\ &= U_{A_1A_2} \cdots U_{A_{mid-1}M} V_{MA_{mid+1}} \cdots V_{A_lA_{l+1}} \\ &= U_{A_1A_2} \cdots U_{A_{mid-1}M} U'_{A_{mid+1}M} \cdots U'_{A_{l+1}A_l} \\ &= U_{A_1A_2} \cdots U_{A_{mid-1}M} (U_{A_{l+1}A_l} \cdots U_{A_{mid+1}M})' \\ &= PM_{P_L} PM'_{P_R{}^{-1}} \end{aligned} \tag{3.4}$$

The above equation shows that the relevance of A_1 and A_{l+1} based on the path P is the inner product of two probability distributions that A_1 reaches the middle type object M along the path and A_{l+1} reaches M against the path. For two instances a and b in A_1 and A_{l+1}, respectively, their relevance based on path P is

$$HeteSim(a, b|P) = PM_{P_L}(a, :)PM'_{P_R^{-1}}(b, :) \tag{3.5}$$

where $PM_P(a, :)$ means the ath row in PM_P.

We have stated that HeteSim needs to be normalized. It is reasonable that the relatedness of the same objects is 1, so the HeteSim can be normalized as follows:

Definition 3.8 (*Normalization of HeteSim*) The normalized HeteSim score between two objects a and b based on the meta path P is:

$$HeteSim(a, b|P) = \frac{PM_{P_L}(a, :)PM'_{P_R^{-1}}(b, :)}{\sqrt{\|PM_{P_L}(a, :)\|\|PM'_{P_R^{-1}}(b, :)\|}} \tag{3.6}$$

In fact, the normalized HeteSim is the cosine of the probability distributions of the source object a and target object b reaching the middle type object M. It ranges from 0 to 1. Figure 3.1d shows the normalized HeteSim scores. It is clear that the normalized HeteSim is more reasonable. The normalization is an important step for HeteSim with the following advantages. (1) The normalized HeteSim has nice properties. The following Property 3.4 shows that HeteSim satisfies the identity of indiscernibles. (2) It has a nice interpretation. The normalized HeteSim is the cosine of two vectors representing reachable probability. As Fouss et al. pointed out [3], the angle between the node vectors is a much more predictive measure than the distance between the nodes. In the following section, the HeteSim means the normalized HeteSim.

3.1.2.4 Properties of HeteSim

HeteSim has good properties, which make it useful in many applications. The proof of these properties can be found in [18].

Property 3.3 (Symmetric) $HeteSim(a, b|P) = HeteSim(b, a|P^{-1})$.

Property 3.3 shows the symmetric property of HeteSim. Although PathSim [21] also has the similar symmetric property, it holds only when the path is symmetric and a and b are with the same type. The HeteSim has the more general symmetric property not only for symmetric paths (note that P is equal to P^{-1} for symmetric paths) but also for asymmetric paths.

Property 3.4 (Self-maximum) $HeteSim(a, b|P) \in [0, 1]$. $HeteSim(a, b|P)$ *is equal to 1 if and only if* $PM_{P_L}(a, :)$ *is equal to* $PM_{P_R^{-1}}(b, :)$.

Property 3.4 shows HeteSim is well constrained. For a symmetric path P (i.e., $P_L = P_R^{-1}$), $PM_{P_L}(a, :)$ is equal to $PM_{P_R^{-1}}(a, :)$, and thus, $HeteSim(a, a|P)$ is equal to 1. If we define the distance between two objects (i.e., $dis(s, t)$) as $dis(s, t) = 1 - HeteSim(s, t)$, the distance of the same object is zero (i.e., $dis(s, s) = 0$). As a consequence, HeteSim satisfies the identity of indiscernibles. Note that it is a general

identity of indiscernibles. For two objects with different types, their HeteSim score is also 1 if they have the same probability distribution on the middle type object. It is reasonable, since they have the similar structure based on the given path.

Since HeteSim obeys the properties of nonnegativity, identity of indiscernibles, and symmetry, we can say that HeteSim is a semi-metric measure [22]. Because of a path-based measure, HeteSim does not obey the triangle inequality. A semi-metric measure has many good merits and can be widely used in many applications [22].

Property 3.5 (Connection to SimRank) *For a bipartite graph* $G = (V, E)$ *based on the schema* $S = (\{A, B\}, \{R\})$*, suppose the constant* C *in SimRank is 1,*
$SimRank(a_1, a_2) = \lim_{n \rightsquigarrow \infty} \sum_{k=1}^{n} HeteSim(a_1, a_2 | (RR^{-1})^k)$,
$SimRank(b_1, b_2) = \lim_{n \rightsquigarrow \infty} \sum_{k=1}^{n} HeteSim(b_1, b_2 | (R^{-1}R)^k)$.
where $a_1, a_2 \in A$, $b_1, b_2 \in B$ *and* $A \xrightarrow{R} B$. *Here HeteSim is the non-normalized version.*

This property reveals the connection of SimRank and HeteSim. SimRank sums up the meeting probability of two objects after all possible steps. HeteSim just calculates the meeting probability along the given meta path. If the meta paths explore all possible meta paths among the two types of objects, the sum of HeteSim based on these paths is the SimRank. So we can say that HeteSim is a path-constrained version of SimRank. Through meta paths, HeteSim can subtly evaluate the similarity of heterogeneous objects with fine granularity. This property also implies that HeteSim is more efficient than SimRank, since HeteSim only needs to calculate the meeting probability along the given relevance path, not all possible meta paths.

Moreover, we compare six well-established similarity measures in Table 3.1. There are three similarity measures for heterogeneous networks (i.e., HeteSim, PathSim, and PCRW) and three measures for homogeneous networks (i.e., P-PageRank, SimRank, and RoleSim), respectively. Although these similarity measures all evaluate the similarity of nodes by utilizing network structure, they have different properties and features. Three measures for heterogeneous networks all are path-based, since meta paths in heterogeneous networks embody semantics and simplify network structure. Two RW model-based measures (i.e., P-PageRank and PCRW) do not satisfy the symmetric property. Because of satisfying the triangle inequation, RoleSim is a metric, while HeteSim, PathSim, and SimRank are semi-metric. Different from PathSim, which can only measure the similarity of objects with the same type under symmetric paths, the proposed HeteSim can measure the relevance of heterogeneous (same or different-typed) objects under arbitrary (symmetric or asymmetric) paths. Although HeteSim can be considered as a path-constrained extension of SimRank, HeteSim is a general similarity measure in heterogeneous networks with arbitrary schema, not limited to bipartite or N-partite networks.

Table 3.1 Comparison of different similarity measures. Here, RW means random walk, and PRW means pairwise random walk

	Symmetry	Triangle inequation	Path based	Model	Features
HeteSim	√	×	√	PRW	Evaluate relevance of heterogeneous objects based on arbitrary path
PathSim [21]	√	×	√	Path count	Evaluate similarity of same-typed objects based on symmetric path
PCRW [13]	×	×	√	RW	Measure proximity to the query nodes based on given path
SimRank [6]	√	×	×	PRW	Measure similarity of node pairs based on the similarity of their neighbors
RoleSim [9]	√	√	×	PRW	Measure real-valued role similarity based on automorphic equivalence
P-PageRank [7]	×	×	×	RW	Measure personalized views of importance based on linkage structure

3.1.3 Experiments

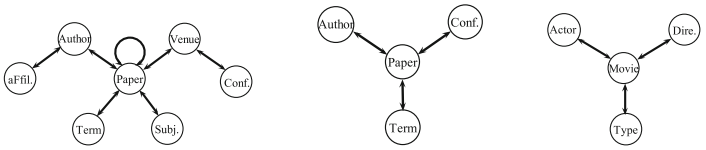
In the experiments, we validate the effectiveness of the HeteSim on three datasets with three case studies and two learning tasks.

3.1.3.1 Datasets

Three heterogeneous information networks are employed in our experiments.

ACM dataset: The ACM dataset was downloaded from ACM digital library[1] in June 2010. The ACM dataset comes from 14 representative computer science conferences: KDD, SIGMOD, WWW, SIGIR, CIKM, SODA, STOC, SOSP, SPAA, SIGCOMM, MobiCOMM, ICML, COLT, and VLDB. These conferences include 196 corresponding venue proceedings. The dataset has 12 K papers, 17K authors, and 1.8 K author affiliations. After removing stop words in the paper titles and abstracts, we get 1.5 K terms that appear in more than 1% of the papers. The network also includes 73 subjects of these papers in ACM category. The network schema of ACM dataset is shown in Fig. 3.2a. Furthermore, we label the data with the ACM category (i.e., subjects) information. That is, with three major subjects (i.e., H.3, H.2, and C.2), we label 7 conferences, 6772 authors, and 4526 papers.

[1] http://dl.acm.org/.



(a) ACM data (b) DBLP data (c) Movie data

Fig. 3.2 Network schema of heterogeneous informations

DBLP dataset [8]: The DBLP dataset is a subnetwork collected from DBLP Web site[2] involving major conferences in four research areas: database, data mining, information retrieval, and artificial intelligence, which naturally form four classes. The dataset contains 14 K papers, 20 conferences, 14K authors, and 8.9 K terms, with a total number of 17 K links. In the dataset, 4057 authors, all 20 conferences, and 100 papers are labeled with one of the four research areas. The network schema is shown in Fig. 3.2b.

Movie dataset [17]: The IMDB movie data comes from the Internet Movie Database,[3] which includes movies, actors, directors, and types. A movie heterogeneous network is constructed from the movie data, and its schema is shown in Fig. 3.2c. The movie data contains 1.5 K movies, 5K actors, 551 directors, and 112 types.

3.1.3.2 Case Study

In this section, we demonstrate the traits of HeteSim through case study in three tasks: automatic object profiling, expert finding, and relevance search.

Task 1: Automatic Object Profiling We first study the effectiveness of HeteSim on different-typed relevance measurement in the automatic object profiling task. If we want to know the profile of an object, we can measure the relevance of the object to objects that we are interested in. For example, the academic profile of Christos Faloutsos[4] can be constructed through measuring the relatedness of Christos Faloutsos with related objects, e.g., conferences, affiliations, and other authors. Table 3.2 shows the lists of top relevant objects with various types on ACM dataset. *APVC* path shows the conferences he actively participates. Note that KDD and SIGMOD are the two major conferences Christos Faloutsos participates, which are mentioned in his home page.[5] From the path *APT*, we can obtain his research interests: data mining, pattern discovery, scalable graph mining, and social network. Using *APS* path, we can discover his research areas represented as ACM subjects: database management

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

[3] www.imdb.com/.

[4] http://www.cs.cmu.edu/~christos/.

[5] http://www.cs.cmu.edu/~christos/misc.html.

Table 3.2 Automatic object profiling task on author "Christos Faloutsos" on ACM dataset

Path	*APVC*		*APT*		*APS*		*APA*	
Rank	Conf.	Score	Terms	Score	Subjects	Score	Authors	Score
1	KDD	0.1198	mining	0.0930	H.2 (database management)	0.1023	Christos Faloutsos	1
2	SIGMOD	0.0284	patterns	0.0926	E.2 (data storage representations)	0.0232	Hanghang Tong	0.4152
3	VLDB	0.0262	scalable	0.0869	G.3 (probability and statistics)	0.0175	Agma Juci M. Traina	0.3250
4	CIKM	0.0083	graphs	0.0816	H.3 (information storage and retrieval)	0.0136	Spiros Papadim-itriou	0.2785
5	WWW	0.0060	social	0.0672	H.1 (models and principles)	0.0135	Caetano Traina, Jr.	0.2680

(H.2) and data storage (E.2). Based on *APA* path, HeteSim finds the most important co-authors, most of which are his Ph.D students.

Task 2: Expert Finding In this case, we want to validate the effectiveness of HeteSim to reflect the relative importance of object pairs through an expert finding task. As we know, the relative importance of object pairs can be revealed through comparing their relatedness. Suppose we know the experts in one domain, the expert finding task here is to find experts in other domains through their relative importance. Table 3.3 shows the relevance scores returned by HeteSim and PCRW on six "conference–author" pairs on ACM dataset. The relatedness of conferences and authors is defined based on the *APVC* and *CVPA* paths which have the same semantics: authors publishing papers in conferences. Due to the symmetric property, HeteSim returns the same value for both paths, while PCRW returns different values for these two paths. Suppose that we are familiar with data mining area and already know that C. Faloutsos is an influential researcher in KDD. Comparing these HeteSim scores, we can find influential researchers in other research areas even if we are not quite familiar with these areas. J.F. Naughton, W.B. Croft, and A. Gupta should be influential researchers in SIGMOD, SIGIR, and SODA, respectively, since they have very similar HeteSim scores to C. Faloutsos. Moreover, we can also deduce that Luo Si and Yan Chen may be active researchers in SIGIR and SIGCOMM, respectively, since they have moderate HeteSim scores. In fact, C. Faloutsos, J.F. Naughton, W.B. Croft, and A. Gupta are top-ranked authors in their research communities. Luo Si and Yan Chen are young professors, and they have done good work in their research areas. However, if the relevance measure is not symmetric (e.g., PCRW), it is very hard to tell which authors are more influential when comparing these relevance scores. For example, the PCRW score of Yan Chen and SIGCOMM is the largest one in the *APVC* path. However, the value is the smallest one for the reversed path (i.e., *CVPA* path).

Table 3.3 Relatedness scores of authors and conferences measured by HeteSim and PCRW on ACM dataset

HeteSim		PCRW			
APVC and *CVPA*		*APVC*		*CVPA*	
Pair	Score	Pair	Score	Pair	Score
C. Faloutsos, KDD	**0.1198**	C. Faloutsos, KDD	0.5517	KDD, C. Faloutsos	**0.0087**
W.B. Croft, SIGIR	**0.1201**	W.B. Croft, SIGIR	0.6481	SIGIR, W.B. Croft	**0.0098**
J.F. Naughton, SIGMOD	**0.1185**	J.F. Naughton, SIGMOD	**0.7647**	SIGMOD, J.F. Naughton	0.0062
A. Gupta, SODA	**0.1225**	A. Gupta, SODA	**0.7647**	SODA, A. Gupta	**0.0090**
Luo Si, SIGIR	0.0734	Luo Si, SIGIR	**0.7059**	SIGIR, Luo Si	0.0030
Yan Chen, SIGCOMM	0.0786	Yan Chen, SIGCOMM	**1**	SIGCOMM, Yan Chen	0.0013

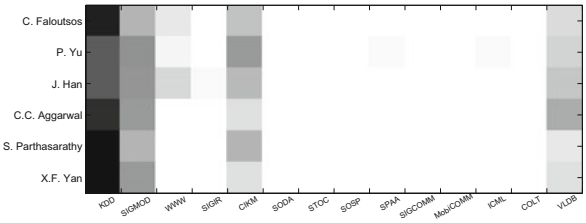


Fig. 3.3 Probability distribution of authors' papers on 14 conferences of ACM dataset

Task 3: Relevance Search based on Path Semantics As we have stated, the path-based relevance measure can capture the semantics of paths. In this relevance search task, we will observe the importance of paths and the effectiveness of semantics capture through the comparison of three path-based measures (i.e., HeteSim, PCRW, and PathSim) and SimRank. This task is to find the top 10 related authors to Christos Faloutsos based on the *APVCVPA* path which means authors publishing papers in same conferences. By ignoring the heterogeneity of objects, we directly run SimRank on whole network and select top ten authors from the rank results which mix different-typed objects together. The comparison results are shown in Table 3.4. At the first sight, we can find that three path-based measures all return researchers having the similar reputation with C. Faloutsos in slightly different orders. However, the results of SimRank are totally against our common sense. We think the reason of bad performances is that SimRank only considers link structure but ignores the link semantics.

In addition, let us analyze the subtle differences of results returned by three path-based measures. The PathSim finds the similar peer authors, such as P. Yu and J. Han. They have the same reputation in data mining field. It is strange for PCRW that the most similar author to C. Faloutsos is not himself, but C. Aggarwal and J. Han. It is

Table 3.4 Top 10 related authors to "Christos Faloutsos" based on *APVCVPA* path on ACM dataset

Rank	HeteSim		PathSim		PCRW		SimRank	
	Author	Score	Author	Score	Author	Score	Author	Score
1	Christos Faloutsos	1	Christos Faloutsos	1	Charu C. Aggarwal	0.0063	Christos Faloutsos	1
2	Srinivasan Parthasarathy	0.9937	Philip Yu	0.9376	Jiawei Han	0.0061	Edoardo Airoldi	0.0789
3	Xifeng Yan	0.9877	Jiawei Han	0.9346	Christos Faloutsos	0.0058	Leejay Wu	0.0767
4	Jian Pei	0.9857	Jian Pei	0.8956	Philip Yu	0.0056	Kensuke Onuma	0.0758
5	Jiong Yang	0.9810	Charu C. Aggarwal	0.7102	Alia I. Abdelmoty	0.0053	Christopher R. Palmer	0.0699
6	Ruoming Jin	0.9758	Jieping Ye	0.6930	Chris B. Jones	0.0053	Anthony Brockwell	0.0668
7	Wei Fan	0.9743	Heikki Mannila	0.6928	Jian Pei	0.0034	Hanghang Tong	0.0658
8	Evimaria Terzi	0.9695	Eamonn Keogh	0.6704	Heikki Mannila	0.0032	Evan Hoke	0.0651
9	Charu C. Aggarwal	0.9668	Ravi Kumar	0.6378	Eamonn Keogh	0.0031	Jia-Yu Pan	0.0650
10	Mohammed J. Zaki	0.9645	Vipin Kumar	0.6362	Mohammed J. Zaki	0.0027	Roberto Santos Filho	0.0648

obviously not reasonable. Our conjecture is that C. Aggarwal and J. Han published many papers in the conferences that C. Faloutsos participated in, so C. Faloutsos has more reachable probability on C. Aggarwal and J. Han than himself along the *APVCVPA* path. HeteSim's results are a little different. The most similar authors are S. Parthasarathy and X. Yan, instead of P. Yu and J. Han. Let us revisit the semantics of the path *APVCVPA*: authors publishing papers in the same conferences. Figure 3.3 shows the reachable probability distribution from authors to conferences along the path *APVC*. It is clear that the probability distribution of papers of S. Parthasarathy and X. Yan on conferences is more close to that of C. Faloutsos, so they should be more similar to C. Faloutsos based on the same conference publication. Although P. Yu and J. Han have the same reputation with C. Faloutsos, their papers are more broadly published in different conferences. So they are not the most similar authors to C. Faloutsos based on the *APVCVPA* path. As a consequence, the HeteSim more accurately captures the semantics of the path.

Since meta path can embody semantics, we can apply HeteSim to do semantic recommendation based on paths given by users. Following this idea, a semantic-based recommended system HeteRecom [17] has been designed.

3.1.3.3 Performance on Query Task

The query task will validate the effectiveness of HeteSim on query search of heterogeneous objects. Since PathSim cannot measure the relatedness of different-typed objects, we only compare HeteSim with PCRW in this experiment. On DBLP dataset, we measure the proximity of conferences and authors based on the *CPA* and *CPAPA* paths. For each conference, we rank its related authors according to their measure scores. Then, we draw the ROC curve of top 100 authors according to the labels of authors (when the labels of author and conference are the same, it is true, else it is false). After that, we calculate the AUC (Area Under ROC Curve) score to evaluate the performances of the ranked results. Note that all conferences and some authors on the DBLP dataset are labeled with one of the four research areas. The larger score means the better performance. We evaluate the performances on 9 representative conferences, and their AUC scores are shown in Table 3.5. We can find that HeteSim consistently outperforms PCRW in most conferences under these two paths. It shows that the proposed HeteSim method can work better than the asymmetric similarity measure PCRW on proximity query task.

3.1.3.4 Performance on Clustering Task

Due to the symmetric property, HeteSim can be applied to clustering tasks directly. In order to evaluate its performance, we compare HeteSim with five well-established similarity measures, including two path-based measures (i.e., PathSim and PCRW) and three homogeneous measures (i.e., SimRank, RoleSim, and P-PageRank). These measures use the same information to determine the pairwise similarity between

Table 3.5 AUC values for the relevance search of conferences and authors based on different paths on DBLP dataset

Paths	Methods	KDD	ICDM	SDM	SIGMOD	VLDB	ICDE	AAAI	IJCAI	SIGIR
CPA	HeteSim	0.811	0.675	0.950	0.766	0.826	0.732	0.811	0.875	0.613
	PCRW	0.803	0.673	0.939	0.758	0.820	0.726	0.806	0.871	0.606
CPAPA	HeteSim	0.845	0.767	0.715	0.831	0.872	0.791	0.817	0.895	0.952
	PCRW	0.844	0.762	0.710	0.822	0.886	0.789	0.807	0.900	0.949

objects. We evaluate the clustering performances on DBLP and ACM datasets. There are three tasks: conference clustering based on *CPAPC* path, author clustering based on *APCPA* path, and paper clustering based on *PAPCPAP* path. For asymmetric measures (i.e., PCRW and P-PageRank), the symmetric similarity matrix can be obtained through the average of similarity matrices based on paths P and P^{-1}. For RoleSim, it is applied in the network constructed by path P. For SimRank and P-PageRank, they are applied in the subnetwork constructed by path P_L (note that the three paths in the experiments are symmetric). Then, we apply normalized cut [16] to perform clustering based on the similarity matrices obtained by different measures. The number of clusters are set as 4 and 3 for DBLP and ACM datasets, respectively. The *NMI* criterion (Normalized Mutual Information) [19] is used to evaluate the clustering performances on conferences, authors, and papers. *NMI* is between 0 and 1 and prothe higher the better. In experiments, the damping factors for P-PageRank, SimRank, and RoleSim are set as 0.9, 0.8, and 0.1, respectively.

The average clustering accuracy results of 100 runs are summarized in Table 3.6. We can find that, on all six tasks, HeteSim achieves best performances on four of them as well as good performances on other two tasks. The mediocre results of PCRW and P-PageRank illustrate that, although symmetric similarity measures can be constructed by the combination of two random walk processes, the simple combination cannot generate good similarity measures. RoleSim aims to detect role similarity, a little bit different from structure similarity, so it has bad performances in these clustering tasks. The experiments show that HeteSim not only does well on similarity measure of same-typed objects but also has the potential as the similarity measure in clustering.

3.1.4 Quick Computation Strategies and Experiments

HeteSim has a high-computation demand for time and space. It is not affordable for online query in large-scale information networks. So a primary strategy is to compute relevance matrix off-line and do online queries with these matrices. For frequently used meta paths, the relatedness matrix $HeteSim(A, B|P)$ can be materialized ahead of time. The online query on $HeteSim(a, B|P)$ will be very fast, since it only needs to locate the row and column in the matrix. However, it also costs much time and space to materialize all frequently used paths. As a consequence, we propose four strategies to fast compute the relevance matrix. Moreover, experiments validate the effectiveness of these strategies.

3.1.4.1 Quick Computation Strategies

The computation of HeteSim includes two phases: matrix multiplication (denoted as MUL, i.e., the computation of PM_{P_L} and $PM_{P_R^{-1}}$) and relevance computation (denoted as REL, i.e., the computation of $PM_{P_L} * PM_{P_R^{-1}}$ and normalization). Through

Table 3.6 Comparison of clustering performances for similarity measures on DBLP and ACM datasets

Methods	DBLP dataset						ACM dataset					
	Venue NMI		Author NMI		Paper NMI		Venue NMI		Author NMI		Paper NMI	
	Mean	Dev.	Mean	Dev.	Mean	Dev.	Mean	Dev.	Mean	Dev.	Mean	Dev.
HeteSim	0.768	0.071	**0.728**	0.083	**0.498**	0.067	**0.843**	0.140	0.405	0.1	**0.439**	0.063
PathSim	0.816	0.078	0.672	0.085	0.383	0.058	0.785	0.164	0.378	0.091	0.432	0.087
PCRW	0.709	0.072	0.710	0.080	0.488	0.039	0.840	0.141	**0.414**	0.092	0.429	0.074
SimRank	**0.888**	0.092	0.685	0.066	0.469	0.031	0.835	0.139	0.375	0.115	0.410	0.073
RoleSim	0.278	0.034	0.501	0.040	0.388	0.049	0.389	0.095	0.293	0.016	0.304	0.017
P-PageRank	0.731	0.086	0.441	0.001	0.421	0.063	0.840	0.164	0.363	0.104	0.407	0.093

analyzing the running time of HeteSim on different phases and paths (the details can be seen in [18]), we find two characteristics of HeteSim computation. (1) The relevance computation is the main time-consuming phase. It implies that the speedup of matrix multiplication may not significantly reduce HeteSim's running time, although this kind of strategies is widely used in accelerating SimRank [6] and PCRW [12]. (2) The dimension and sparsity of matrix greatly affect the efficiency of HeteSim. Although we cannot reduce the running time of relevance computation phase directly, we can accelerate the computation of HeteSim through adjusting matrix dimension and keeping matrix sparse. Based on above idea, we design the following four quick computation strategies.

Dynamic Programming Strategy The matrix multiplication obeys the associative property. Moreover, different computation sequences have different time complexities. The dynamic programming strategy (DP) changes the sequence of matrix multiplication with the associative property. The basic idea of DP is to assign low-dimensioned matrix with the high-computation priority. For a path $P = R_1 \circ R_2 \circ \cdots \circ R_l$, the expected minimal computation complexity of HeteSim can be calculated by the following equation and the computation sequence is recorded by i.

$$Com(R_1 \cdots R_l) = \begin{cases} 0 & l = 1 \\ |R_1.S| \times |R_1.T| \times |R_2.T| & l = 2 \\ \arg\min_i \{Com(R_1 \cdots R_i) + Com(R_{i+1} \cdots R_l) + |R_1.S| \times |R_i.T| \times |R_l.T|\} & l > 2 \end{cases} \tag{3.7}$$

The above equation can be easily solved by dynamic programming method with the $O(l^2)$ complexity. The running time can be omitted, since l is much smaller than the matrix dimension. Note that the DP strategy only accelerates the MUL phase (i.e., matrix multiplication) and it does not change relevance result, so the DP is an information-lossless strategy.

Truncation Strategy The truncation strategy is based on the hypothesis that removing the probability on those less important nodes would not significantly degrade the performance, which has been proved by many researches [12]. One advantage of this strategy is to keep matrix sparse. The sparse matrix greatly reduces the amount of space and time consumption. The basic idea of truncation strategy is to add a truncation step at each step of random walk. In the truncation step, the relevance value is set with 0 for those nodes when their relevance values are smaller than a threshold ε. A static threshold is usually used in many methods (e.g., Ref. [12]). However, it has the following disadvantage: It may truncate nothing for matrix whose elements all have high probability, and it may truncate most nodes for matrix whose elements all have low probability. Since we usually pay close attention to the top k objects in query task, the threshold ε can be set as the top k relevance value for each search object. For a similarity matrix with size $M \times L$, the k can be dynamically adjusted as follows.

$$k = \begin{cases} L & \text{if } L \leq W \\ \lfloor (L - W)^{\beta} \rfloor + W (\beta \in [0, 1]) & \text{others} \end{cases}$$

where W is the number of top objects, decided by users. The basic idea of dynamic adjustment is that the k slowly increases for super object type (i.e., L is large). The W and β determine the truncation level. The larger W or β will cause the larger k, which means a denser matrix. It is expensive to determine the top k relevance value for each object, so we can estimate the value by the top kM value for the whole matrix. Furtherly, the top kM value can be approximated by the sample data with ratio γ from the raw matrix. The larger γ leads to more accurate approximation with longer running time. In summary, the truncation strategy is an information-loss strategy, which keeps matrix sparse with small sacrifice on accuracy. In addition, it needs additional time to estimate the threshold ε.

Hybrid Strategy As discussed above, the DP strategy can accelerate the MUL phase and the truncation strategy can indirectly speed up the REL phase by keeping sparse matrix. So a hybrid strategy can be designed to combine these two strategies. For the MUL phase, the DP strategy is applied. After obtaining the PM_{P_L} and $PM_{P_R{}^{-1}}$, the truncation strategy is added. Different from the above truncation strategy, the hybrid strategy only truncates the PM_{P_L} and $PM_{P_R{}^{-1}}$. The hybrid strategy utilizes the benefits of DP and truncation strategies. It is also an information-loss strategy, since the truncation strategy is employed.

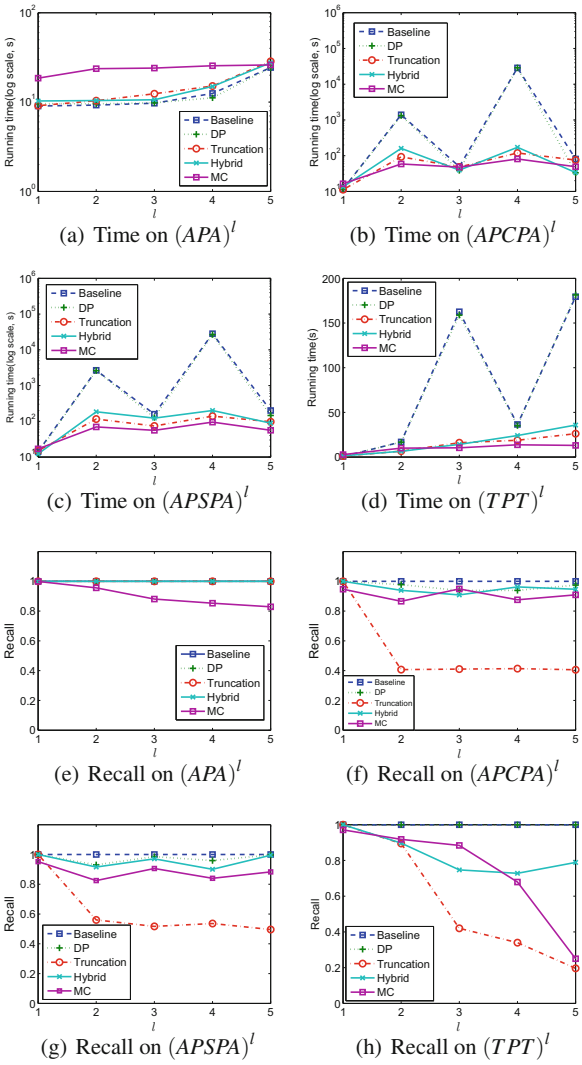
Monte Carlo Strategy Monte Carlo method (MC) is a class of computational algorithms that estimate results through repeating random sampling. It has been applied to compute approximate values of matrix multiplication [2, 12]. In this study, we applied the MC strategy to estimate the value of PM_{P_L} and $PM_{P_R{}^{-1}}$. The value of $PM_P(a, b)$ can be approximated by the normalized count of the number of times that the walkers visit the node b from a along the path P.

$$PM_P(a, b) = \frac{\#times \quad the \quad walkers \quad visit \quad b \quad along \quad P}{\#walkers \quad from \quad a}$$

The number of walkers from a (i.e., K) controls the accuracy and amount of computation. The larger K will achieve more accurate estimation with more time cost. An advantage of the MC strategy is that its running time is not affected by the dimension and sparsity of matrix. However, the high-dimension matrix needs larger K for high accuracy. As a sampling method, the MC is also an information-loss strategy.

3.1.4.2 Quick Computation Experiments

We validate the efficiency and effectiveness of quick computation strategies on the ACM dataset. The four paths are used: $(APA)^l$, $(APCPA)^l$, $(APSPA)^l$, and $(TPT)^l$. l means times of path repetition and ranges from 1 to 5. Four quick computation strategies and the original method (i.e., baseline) are employed. The parameters in



(a) Time on $(APA)^l$ (b) Time on $(APCPA)^l$

(c) Time on $(APSPA)^l$ (d) Time on $(TPT)^l$

(e) Recall on $(APA)^l$ (f) Recall on $(APCPA)^l$

(g) Recall on $(APSPA)^l$ (h) Recall on $(TPT)^l$

Fig. 3.4 Running time and accuracy of computing HeteSim based on different strategies and paths

truncation process are set as follows: the number of top objects W is 200, β is 0.5, and γ is 0.005. The number of walkers (i.e., K) in MC strategy is 500. The running time and accuracy of all strategies are recorded. In the accuracy evaluation, the relevance matrices obtained by the original method are regarded as the baseline. The accuracy is the *recall* criterion on the top 100 objects obtained by each strategy. All experiments are conducted on machines with Intel Xeon 8-Core CPUs of 2.13 GHz and 64 GB RAM.

Figure 3.4 shows the running time and accuracy of four strategies on different paths. The running time of these strategies is illustrated in Fig. 3.4a–d. We can observe

that the DP strategy almost has the same running time with the baseline. It only speeds up the HeteSim computation when the MUL phase dominates the whole running time (e.g., $(APCPA)^5$ and $(APSPA)^5$). It is not the case for the truncation and hybrid strategies, which significantly accelerate the HeteSim computation and have a close speedup ratio on most conditions. Except the *APA* path, the MC strategy has the highest speedup ratio among all four strategies on most conditions. Then, let us observe their accuracy from Fig. 3.4e–h. The accuracy of the DP strategy is always close to 1. The hybrid strategy achieves the second performances for most paths. The accuracy of the MC strategy is also high for most paths, while it fluctuates on different paths. Obviously, the truncation strategy has the lowest accuracy on most conditions.

As we have noted, the DP, as an information-lossless strategy, only speeds up the MUL phase which is not the main time-consuming part for most paths. So the DP strategy trivially accelerates HeteSim with the accuracy close to 1. The truncation strategy is an information-loss strategy to keep matrix sparse, so it can effectively accelerate HeteSim. That is the reason why the truncation strategy has the high speedup ratio but low accuracy. Because the hybrid strategy combines the benefits of DP and truncation strategy, it has a close speedup ratio to the truncation strategy with higher accuracy. In order to achieve high accuracy, more walkers in the MC strategy are needed for high-dimension or sparse matrix, while the fixed walkers in experiments (i.e., K is 500) make the MC strategy the poor accuracy on some conditions.

According to the analysis above, these strategies are suitable for different paths and scenarios. For very sparse matrix (e.g., $(APA)^l$) and low-dimension matrix (e.g., $(APCPA)^3$), all strategies cannot significantly improve efficiency. However, in these conditions, the HeteSim can be quickly computed without any strategies. For those dense (e.g., $(APCPA)^4$) and high-dimension matrix (e.g., $(APSPA)^4$) which have huge computation overhead, the truncation, hybrid, and MC strategies can effectively improve the HeteSim's efficiency. Particularly, the speedup of the hybrid and MC strategies are up to 100 with little loss in accuracy. If the MUL phase is the main time-consuming part for a path, the DP strategy can also speed up HeteSim greatly without loss in accuracy. The MC strategy has very high efficiency, but its accuracy may degrade for high-dimension matrix. So the appropriate K needs to be set through balancing the efficiency and effectiveness.

3.2 Extension of HeteSim

3.2.1 Overview

Many data mining tasks have been exploited in heterogeneous information network, such as clustering [19] and classification [10]. Among these data mining tasks, similarity measure is a basic and important function, which evaluates the similarity

of object pairs on networks. Although similarity measure on homogeneous networks have been extensively studied in the past decades, such as PageRank [15] and SimRank [6], the similarity measure in heterogeneous network is just beginning now and several measures have been proposed including PathSim [21], PCRW [13], and HeteSim [18]. All the three methods are based on meta path [18]. Specially, HeteSim, proposed by Shi et al., has the ability to measure relatedness of objects with the same or different types in a uniform framework. HeteSim has some good properties (e.g., self-maximum and symmetric) and has shown its potential in several data mining tasks. However, we can also find that it has several disadvantages. (1) HeteSim has relatively high computational complexity. Particularly, the adoption of path decomposition approach when it measures the relevance on odd-length path further increases the calculation complexity. (2) Besides, HeteSim cannot be extended to large-scale network with massive data, since its calculation process is based on memory computing. Therefore, it is desired to design a new similarity measure, which not only contains some good properties as HeteSim but also overcomes the disadvantages on computation.

In this chapter, we introduce a new relevance measure method, AvgSim, which is a symmetric measure and uniform measure to evaluate the relevance of same- or different-typed objects. The AvgSim value of two objects is the average of reachable probability under the given path and the reverse path. It guarantees that AvgSim can measure relevance of same or different-typed objects and it has symmetric property. In addition, compared with HeteSim which takes a pairwise random walk, AvgSim does not need to consider the length of path and there is no path decomposition involved. Thus, it is more simple and efficient. Furthermore, we take parallelization of this new algorithm on MapReduce in order to eliminate restriction of memory size and deal with massive data more efficiently in practical applications.

3.2.2 *AvgSim: A New Relevance Measure*

In this section, we will introduce the new meta path-based relevance measure which is called AvgSim and its definition is as follows.

Definition 3.9 (*AvgSim*) Given a meta path P which is defined on the composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_l$, AvgSim between two objects s and t (s is the source object and t is the target object) is:

$$AvgSim(s, t|P) = \frac{1}{2}[RW(s, t|P) + RW(t, s|P^{-1})] \tag{3.8}$$

$$RW(s, t|R_1 \circ R_2 \circ \ldots \circ R_l) = \frac{1}{|O(s|R_1)|} \sum_{i=1}^{|O(s|R_1)|} RW(O_i(s|R_1), t|R_2 \circ \ldots \circ R_l) \tag{3.9}$$

Equation 3.8 shows the relevance of source object and target object based on meta path P is the arithmetic mean value of random walk result from s to t along P and reversed random walk result from t to s along P^{-1}. Equation 3.9 shows the decomposed step of AvgSim, namely the measure of random walk. The measure takes a random walk step by step from the starting point s to the end point t along path P using an iterative method, where $|O(s|R_1)|$ is the out-neighbors of s based on relation R_1. If there is no out-neighbors of s on R_1, then the relevance value of s and t is 0 because s cannot reach t. We need to calculate the random walk probabilities for each out-neighbor of s to t iteratively, and then, sum them up. Finally, the summation should be normalized by the number of out-neighbors to get the average relatedness.

Then, we will study on how to calculate AvgSim generally with matrix. Given a simple directed meta path $A \xrightarrow{R} B$, where objects A and B are linked though relation R. The relationship between A and B can be expressed by adjacent matrix, denoted as M_{AB}. Two normalized matrices R_{AB} and C_{AB} are generated by normalizing M_{AB} according to row vector and column vector, respectively. R_{AB} and C_{AB} are **transition probability matrix** which represent $A \xrightarrow{R} B$ and $B \xrightarrow{R^{-1}} A$, respectively. According to properties of matrix, we can derive relations $R_{AB} = C'_{BA}$ and $C_{AB} = R'_{BA}$, where R'_{AB} is the transpose of R_{AB}.

If we extend the simple meta path to $P = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$ where R is a composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_l$, then the relationship between A_1 and A_{l+1} is expressed as **reachable probability matrix** which is obtained by multiplying the transition probability matrices along the meta path. The reachable probability matrix of P is defined as $RW_P = R_{A_1A_2}R_{A_2A_3}\ldots R_{A_lA_{l+1}}$, where RW suggests RW_P is the random walk relatedness matrix from object A_1 to A_{l+1} along path P.

Then, we can rewrite AvgSim using the reachable probability matrix according to Eqs. 3.8 and 3.9 as follows.

$$\begin{aligned} &AvgSim(A_1, A_{l+1}|P) \\ &= \frac{1}{2}[RW(A_1, A_{l+1}|P) + RW(A_{l+1}, A_1|P^{-1})] = \frac{1}{2}[RW_P + RW'_{P^{-1}}] \end{aligned} \tag{3.10}$$

According to $C_{AB} = R'_{BA}$, Eq. 3.11 is derived below. We notice that the calculation of AvgSim is unified as two-chain matrix multiplication of transition probability matrices. The only difference between two chains is the normalization form of original adjacent matrix.

$$\begin{aligned} AvgSim(A_1, A_{l+1}|P) &= \frac{1}{2}[R_{A_1A_2}R_{A_2A_3}\ldots R_{A_lA_{l+1}} + (R_{A_{l+1}A_l}R_{A_lA_{l-1}}\ldots R_{A_2A_1})'] \\ &= \frac{1}{2}[R_{A_1A_2}R_{A_2A_3}\ldots R_{A_lA_{l+1}} + C_{A_1A_2}C_{A_2A_3}\ldots C_{A_lA_{l+1}}] \end{aligned} \tag{3.11}$$

AvgSim can measure the relevance of any heterogeneous or homogeneous object pair based on symmetrical path (e.g., *APCPA*) or asymmetrical path (e.g., *APS*). Besides, the method has a symmetric property, which can be verified easily from the definition equation of AvgSim. However, the calculation of AvgSim mainly lies in the chain matrix multiplication which is time-consuming and restricted of memory size. In order to apply the algorithm in real large-scale heterogeneous information network, we have to consider how to improve the efficiency of AvgSim.

3.2.3 Parallelization of AvgSim

Parallelism [1] is an effective method for processing massive data and improving algorithm's efficiency. According to the features and application scenarios of AvgSim, we parallelize it as the following steps.

1. Since the core calculation of AvgSim is the chain matrix multiplication, we firstly change the order of matrix multiplication operations applying dynamic programming strategy.
2. After Step 1, we turn to focus on large-scale matrix multiplication and it can be parallelized on Hadoop distributed system using MapReduce programming model.

As we know, different orders of operations in chain matrix multiplication leads to different computation time. There exists an optimal order of chain matrix multiplication using dynamic programming, which consumes the shortest computation time. Thus, we can apply dynamic programming to improve the efficiency of parallelized AvgSim.

Parallelization of AvgSim is mainly the parallelization of matrix multiplication after the dynamic programming process. Here, we use the "block matrix multiplication" method on MapReduce to transform multiplication of two large matrices into several multiplications of smaller matrices. This method is flexible for selecting dimensions of block matrix according to the configuration of Hadoop cluster, and it avoids exceeding the memory size. The parallelization of block matrix multiplication is implemented by two-round MapReduce computing. The detailed algorithms can be found in [14].

Applying two-round MapReduce algorithm above iteratively to the chain matrix multiplication which is reordered by dynamic programming, we can obtain one of the two reachable probability matrices of AvgSim (e.g., RW_P, which is measured in the given meta path P), and the other probability matrix ($RW'_{P^{-1}}$) can be obtained in the same procedure. Finally, the relevance matrix is derived by taking the arithmetic mean of these two reachable probability matrices.

3.2.4 Experiments

Three datasets, ACM dataset, DBLP dataset, and Matrix dataset, are used in experiments. In detail, the ACM dataset contains 17 K authors, 1.8K author affiliations, 12K papers, and 14 computer science conferences including 196 corresponding venue proceedings. We also extract 1.5 K terms and 73 subjects from these papers. The DBLP dataset contains 14 K papers, 14K authors, 20 conferences, and 8.9 K terms. And we label 20 conferences, 100 papers, and 4057 authors in the dataset with four research areas including database, data mining, information retrieval, and artificial intelligence for experiment use. And the matrix dataset *(40 matrices in total)* contains several artificially generated large-scale sparse square matrices, whose dimensions are 1000×1000, 5000×5000, $10{,}000 \times 10{,}000$, $20{,}000 \times 20{,}000$, $40{,}000 \times 40{,}000$, $80{,}000 \times 80{,}000$, $100{,}000 \times 100{,}000$, and $150{,}000 \times 150{,}000$, respectively. And the sparsity of each matrix is 0.0001, 0.0003, 0.0005, 0.0007, and 0.001.

3.2.4.1 Effectiveness of AvgSim

In this section, we design experiments to validate the effectiveness and efficiency of AvgSim. We design two tasks to verify the effectiveness of AvgSim, which are query task and clustering task, respectively.

In the query task, we compare the performance of AvgSim with both HeteSim and PCRW though measuring the relevance of heterogeneous objects on DBLP dataset. Based on labels of the dataset, we calculate the AUC score to evaluate the performances of different methods, where the query task is to find authors for each conference based on the path *CPA*. We evaluated 9 out of 20 marked conferences, whose AUC values are shown in Table 3.7. We notice that AvgSim gets the highest value on 8 conferences, which means AvgSim performs better than other two methods in this query task.

In the clustering task, we compare the performance of AvgSim with both HeteSim and PathSim through measuring the similarity of homogeneous objects on DBLP dataset. We firstly apply three algorithms, respectively, to derive the similarity matrices on three meta paths including *CPAPC*, *APCPA*, and *PAPCPAP*. We perform clustering task based on the similarity matrices with normalized cut and

Table 3.7 AUC values for relevance search of conferences and authors based on CPA path on DBLP dataset

	KDD	ICDM	SDM	SIGMOD	VLDB	ICDE	AAAI	IJCAI	SIGIR
HeteSim	0.8111	0.6752	**0.6132**	0.7662	0.8262	**0.7322**	0.8110	0.8754	0.9504
PCRW	0.8030	0.6731	0.6068	0.7588	0.8200	0.7263	0.8067	0.8712	0.9390
AvgSim	**0.8117**	**0.6753**	0.6072	**0.7668**	**0.8274**	0.7286	**0.8114**	**0.8764**	**0.9525**

Table 3.8 Clustering accuracy results for path-based relevance measures on DBLP dataset

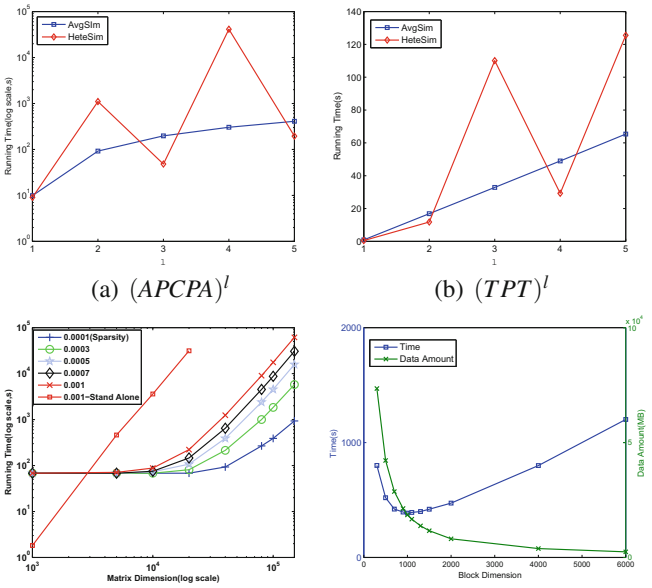
	Venue NMI	Author NMI	Paper NMI
PathSim	0.8162	0.6725	0.3833
HeteSim	0.7683	0.7288	0.4989
AvgSim	**0.8977**	**0.7556**	**0.5101**

then evaluate the performances on conferences, authors, and papers using *NMI* criterion (Normalized Mutual Information). The clustering accuracy result is shown in Table 3.8, and AvgSim obtains the highest *NMI* value in all the three tasks. In all, the results of query task and clustering task suggest that AvgSim performs well in effectiveness.

3.2.4.2 Efficiency of AvgSim

In this section, we will verify the efficiency of AvgSim on ACM dataset. We take relevance measure experiments of AvgSim and HeteSim, respectively, based on meta paths including $(APCPA)^l$ and $(TPT)^l$, where l is the number of path repetitions with a range from 1 to 5.

Figure 3.5a, b shows the relationship between running time and different meta paths for each method. We notice that the running time of HeteSim exhibits great fluctuations with the change of path length, while AvgSim is much stable. According to the definition of AvgSim, the longer paths (i.e., l) it measures, the more matrices need to be multiplied, and thus, the running time increases persistently. In contrast, the calculation of HeteSim needs two steps including matrix multiplication and relevance computation. In the matrix multiplication step, HeteSim calculates the reachable probability matrices from source and target nodes to the middle node, respectively. The longer paths it measures, the more time it needs. In relevance computation step, the relevance matrix is the multiplication of two probability matrices in previous step. The time for the second step is determined by the scale of middle node. In all, the relevance computation of HeteSim affects its performance to a great extent and it will be relatively poor for large-scale matrices. Conversely, AvgSim performs much more stable, and its efficiency is only related to the matrix dimension and



(a) $(APCPA)^l$ (b) $(TPT)^l$

(c) Dimension and sparsity (d) Partition strategy

Fig. 3.5 Running time of AvgSim and HeteSim based on different meta paths and factors affecting parallelized block matrix multiplication: **a** Running time on $(APCPA)^l$; **b** Running time on $(TPT)^l$; **c** Matrix dimension and sparsity factors; **d** Partition strategy factor

meta path length, which can be improved by the parallelized matrix multiplication on MapReduce.

All parallelized matrix multiplication experiments are conducted in a cluster composed of 7 machines with 4-cores E3-1220 V2 CPUs of 3.10 GHz and 32 GB RAM running on RedHat 4 operating system. The experiments will study several factors affecting block matrix multiplication, including matrix, matrix sparsity, and partition strategy (i.e., dimensions of blocks). Results will reflect the performance of parallelized AvgSim algorithm.

Figure 3.5c shows the effect of matrix dimensions and matrix sparsity on the running time of parallelized block matrix multiplication. All the matrix multiplications are done on the *Matrix* dataset, and it applies the partition strategy of 1000×1000 block matrix. We notice from Fig. 3.5c that the larger dimensions or more density of matrix are, the more time in matrix multiplication is required. And the comparison results between stand-alone and parallelized matrix multiplication with the sparsity of 0.001 shows that the stand-alone algorithm costs shorter time for a quite small matrix dimension because the parallelized algorithm spends lots of time in the starting task nodes of Hadoop cluster and resources of cluster are not fully utilized for a small amount of calculations. However, the efficiency of parallelized algorithm is much better as the matrix dimension increases. Besides, the stand-alone algorithm is

restricted of memory size, so there are no results derived in the last three large-scale matrix multiplications shown in Fig. 3.5c.

Figure 3.5d shows the effect of intermediate data amount and partition strategy of block matrix multiplication. There are 11 kinds of partition strategies with the square block matrix dimensions from 300×300 to 6000×6000, where the square matrix is with the dimension of $100{,}000 \times 100{,}000$ and the sparsity of 0.0001 in the experiment. We notice from Fig. 3.5d that the intermediate data amount of matrix multiplication decreases gradually with the increase of block dimension. In contrast, the running time reaches its minimum value at 5th data point. Smaller intermediate data amount results in less disk IO operations and data amount transmitted by shuffle, which also leads to shorter time and better performance to a certain extent as the data points before 1000 near 1000 reflected. However, the excessive large block dimension will reduce the concurrent granularity and increase the amount of calculations for single node, which conversely results in longer time of computation as the data points after 1000 reflected.

In all, the appropriate partition strategy and sufficient sizes of cluster greatly affect the efficiency in parallelized block matrix multiplications. Applying parallelization method, AvgSim gains the ability to measure relevance in larger-scale networks with massive data efficiently.

3.3 Conclusion

In this chapter, we study the relevance search problem which measures the relatedness of heterogeneous objects in heterogeneous networks. We introduce a general relevance measure, called HeteSim. As a path-constraint and semi-metric measure, HeteSim can measure the relatedness of same-typed or different-typed objects in a uniform framework. In addition, we also present a modification of HeteSim. Extensive experiments validate the effectiveness and efficiency of the proposed measures on evaluating the relatedness of heterogeneous objects.

The similarity measure of objects in heterogeneous networks is an important and basic task, which can be used in many applications. There are some interesting directions for future work. Similarity measures are designed for more complex HIN, such as hybrid network integrating heterogeneous features and text information, and multiple or weighted meta paths. In addition, similarity measures are widely used in real applications where the network scales are usually huge. We need to design the efficient and parallelized computation methods.

References

1. Cao, L., Cho, B., Kim, H.D., Li, Z., Tsai, M.H., Gupta, I.: Delta-simrank computing on mapreduce. In: Big Data Workshop, pp. 28–35 (2012)
2. Fogaras, D., Rácz, B., Csalogány, K., Sarlós, T.: Towards scaling fully personalized PageRank: algorithms, lower bounds, and experiments. Internet Math. **2**(3), 333–358 (2005)
3. Fouss, F., Pirotte, A., Renders, J.M., Saerens, M.: Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. Knowl. Data Eng. **19**(3), 355–369 (2007)
4. Han, J.: Mining heterogeneous information networks by exploring the power of links. In: DS, pp. 13–30 (2009)
5. Jamali, M., Lakshmanan, L.V.S.: HeteroMF: recommendation in heterogeneous information networks using context dependent factor models. In: WWW, pp. 643–654 (2013)
6. Jeh, G., Widom, J.: SimRank: A measure of structural-context similarity. In: KDD, pp. 538–543 (2002)
7. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW, pp. 271–279 (2003)
8. Ji, M., Sun, Y., Danilevsky, M., Han, J., Gao, J.: Graph regularized transductive classification on heterogeneous information networks. In: ECML/PKDD, pp. 570–586 (2010)
9. Jin, R., Lee, V.E., Hong, H.: Axiomatic ranking of network role similarity. In: KDD, pp. 922–930 (2011)
10. Kong, X., Yu, P.S., Ding, Y., Wild, D.J.: Meta path-based collective classification in heterogeneous information networks. In: CIKM, pp. 1567–1571 (2012)
11. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news. Commun. ACM **40**(3), 77–87 (1997)
12. Lao, N., Cohen, W.: Fast query execution for retrieval models based on path constrained random walks. In: KDD, pp. 881–888 (2010)
13. Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. Mach. Learn. **81**(2), 53–67 (2010)
14. Meng, X., Shi, C., Li, Y., Zhang, L., Wu, B.: Relevance measure in large-scale heterogeneous networks. In: APWeb, pp. 636–643 (2014)
15. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. In: Stanford InfoLab, pp. 1–14 (1998)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
17. Shi, C., Zhou, C., Kong, X., Yu, P.S., Liu, G., Wang, B.: HeteRecom: a semantic-based recommendation system in heterogeneous networks. In: KDD, pp. 1552–1555 (2012)
18. Shi, C., Kong, X., Huang, Y., Philip, S.Y., Wu, B.: Hetesim: a general framework for relevance measure in heterogeneous networks. IEEE Trans. Knowl. Data Eng. **26**(10), 2479–2492 (2014)
19. Sun, Y., Han, J., Zhao, P., Yin, Z., Cheng, H., Wu, T.: RankClus: integrating clustering with ranking for heterogeneous information network analysis. In: EDBT, pp. 565–576 (2009)
20. Sun, Y., Yu, Y., Han, J.: Ranking-based clustering of heterogeneous information networks with star network schema. In: KDD, pp. 797–806 (2009)
21. Sun, Y.Z., Han, J.W., Yan, X.F., Yu, P.S., Wu, T.: PathSim: Meta path-based Top-K similarity search in heterogeneous information networks. In: VLDB, pp. 992–1003 (2011)
22. Xia, Q.: The geodesic problem in quasimetric spaces. J. Geom. Anal. **19**(2), 452–479 (2009)
23. Zhu, J., De Vries, A.P., Demartini, G., Iofciu, T.: Evaluating relation retrieval for entities and experts. In: Proceedings of the SIGIR 2008 Workshop on Future Challenges in Expertise Retrieval (fCHER), pp. 41–44 (2008)