Query-Driven Discovery of Semantically Similar Substructures in Heterogeneous Networks

Xiao Yu, Yizhou Sun, Peixiang Zhao, Jiawei Han
Department of Computer Science, University of Illinois at Urbana-Champaign
{xiaoyu1, sun22, pzhao4, hanj}@illinois.edu

ABSTRACT

Heterogeneous information networks that contain multiple types of objects and links are ubiquitous in the real world, such as bibliographic networks, cyber-physical networks, and social media networks. Although researchers have studied various data mining tasks in information networks, interactive query-based network exploration techniques have not been addressed systematically, which, in fact, are highly desirable for exploring large-scale information networks.

In this demo, we introduce and demonstrate our recent research project on query-driven discovery of semantically similar substructures in heterogeneous networks. Given a subgraph query, our system searches a given large information network and finds efficiently a list of subgraphs that are structurally identical and semantically similar. Since data mining methods are used to obtain semantically similar entities (nodes), we use *discovery* as a term to describe this process. In order to achieve high efficiency and scalability, we design and implement a filter-and-verification search framework, which can first generate promising subgraph candidates using off-line indices built by data mining results, and then verify candidates with a recursive pruning matching process. The proposed system demonstrates the effectiveness of our query-driven semantic similarity search framework and the efficiency of the proposed methodology on multiple real-world heterogeneous information networks.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data Mining*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Information Networks*

General Terms

Experimentation, Algorithms

Keywords

Data Mining, Graph Query, Information Network

KDD '2012 Beijing, China

1. INTRODUCTION

Due to the rapid emergence of social networks and information networks extracted from various on-line databases, such as bibliographic databases, movie databases, and social media databases, heterogeneous information network analysis and mining begin to play an essential role in many large-scale systems. Problems including ranking, clustering [6], classification [1], entity similarity search [5] and relationship prediction [4] in information networks have been intensively studied and drawn research interests from a wide range of application domains. However, techniques which allow users to interactively explore large scale networks using query examples have not been studied yet.

Efficient subgraph isomorphism methods [7] [10] have been proposed in homogeneous information networks, with which users can locate identical subgraphs in a graph with a subgraph query. However, due to the rich information possessed in heterogeneous information networks, a user often wants to search for substructures with similar semantic meanings rather than identical subgraphs. Most of the time, it is easy for a user to use an example to describe a substructure, in order to find a list of semantically similar subgraphs. For example, a funding agency is looking for collaborations in academic institutes on graph analysis projects. They do not know many proficient researchers but they know that Christos Faloutsos and Jure Leskovec collaborate on graph analysis. Aiming to find more experts who collaborate on similar topics, a subgraph query can be written to represent a search criterion (Figure 1, left up corner). Desired results would be subgraphs that have identical structure and semantically similar entities so that more experts can be found. To our best knowledge, no existing search engine can answer such kind of queries.

In this demo, we study and demonstrate our newly designed "Semantically Similar SubStructure" discovery system, namely the S^4 system, which takes a subgraph query as input, and retrieve from an information network a list of subgraphs that satisfy the query criteria as mentioned above. Our system, with off-line data mining results as indices, utilizes semi-structure information encoded in information networks and answers semantic similarity based substructure queries efficiently.

In the rest of the paper, we briefly introduce the framework of the new search technique, and demonstrate the power of S^4 system with real-world heterogeneous information networks. For instance, if we input the above query using our interactive web-based interface into the S^4 system, we can get a list of subgraphs as the retrieval results, part of which are presented in Figure 1. By examining the results, one could learn other graph analysis experts who have collaboration relationships, such as Christos Faloutsos and Jon Kleinberg worked on entity connection problems, and Philip Yu and Jimeng Sun are collaborators on large-scale graph analysis, and so on. Results from S^4 provide valuable information and help

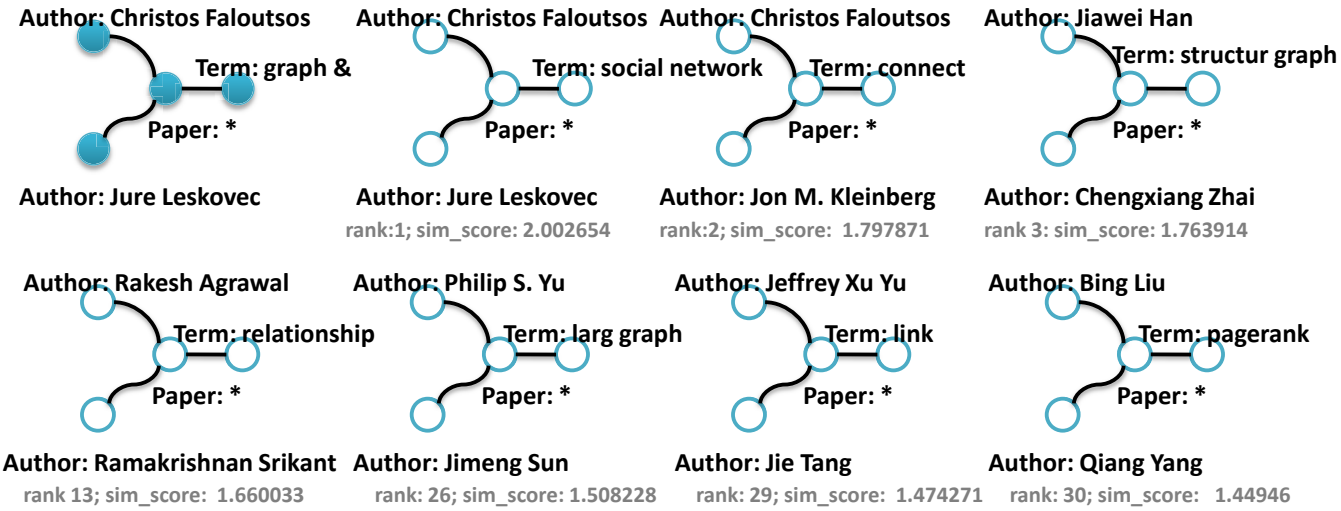

Figure 1: Example of entity similarity based subgraph search, where the subgraph with solid nodes is the query, and the subgraphs with hollow nodes are some retrieval results; similarity score (sim_score) and ranking ($rank$) in the entire retrieval results of each subgraph is also presented. Constraint notations including $*$ and $\&$ can be found in Section

users explore the contents of the information network.

2. PROBLEM DESCRIPTION

We briefly introduce background concepts related to heterogeneous information networks and define semantically similar substructure discovery problem in this section.

As defined in [6], a heterogeneous information network is a directed graph, which contains multiple types of entities and / or multiple types of links. An information network is usually denoted as $G = (V, E)$, where V is the set of nodes (entities) and E is the set of edges (links between entities). If $|typeof(v \in V)| > 1$ and / or also $|typeof(e \in E)| > 1$, the network is a *heterogeneous information network*; otherwise it is a *homogeneous information network*.

Similar to relational databases, heterogeneous information networks usually follow a schema, in which rules of how entities and links coexist are defined. An example of heterogeneous information network and the corresponding network schema can be found as follows.

EXAMPLE 1. *DBLP*[1] *(Digital Bibliography & Library Project) is a computer science bibliographic dataset, which can be described as a heterogeneous information network. It contains four different types of entities (papers, venues, authors and terms). Links exist between papers and authors, papers and venues, papers and terms as well as within paper entities, representing citation relations. The schema of DBLP can be found in Figure 2.*

The semantically similar substructure discovery problem we study and present in this demo is defined in the scope of heterogeneous information networks, although it can also be applied on homogeneous networks. The problem can be defined as follows. Given a subgraph Q with concrete entities and valid links as query, *i.e.*, the query follows network schema, and a heterogeneous information network G, the system will return a Top-K list of subgraphs of G, each of which has identical same structure of of Q, but has different yet semantically similar entities. As shown in Figure 1,

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

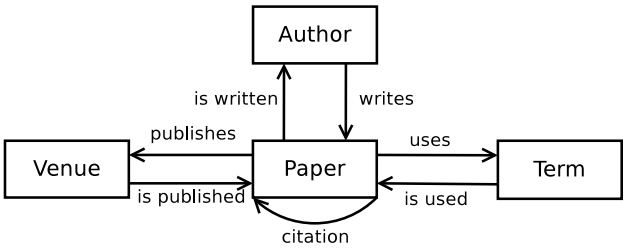

Figure 2: DBLP network schema

retrieval results can help users access and explore Web-scale information networks with the issuing of the aforementioned structure similarity queries.

3. FRAMEWORK AND METHODS

As shown in Figure 3, the architecture of the S^4 system can be divided into two parts, which are *off-line analysis and indexing* and *on-line query answering*. Data mining and analysis over heterogeneous information network, including clustering, entity similarity measurement *etc*, are performed during the off-line stage. Two types of indices are built with the output of off-line mining and analysis. Similarity index utilizes the results of entity similarity calculation and indexes popular entities (entities with high rank or appear often in queries) with their semantically similar siblings. Structure index summarizes the output of structure index selectivity analysis and indexes frequent subgraph structures with a certain space budget.

During the on-line stage, given a subgraph query through a web-based interactive interface, the S^4 system will first verify the subgraph against network schema, and then pass the query down to index matching module. With the help of two indices built off-line, index matching module calculates possible query matching graph locations and entity combinations, denoted as subgraph candidates. If necessary, a revised subgraph isomorphism examination will be applied to each subgraph candidate, and further verify the search

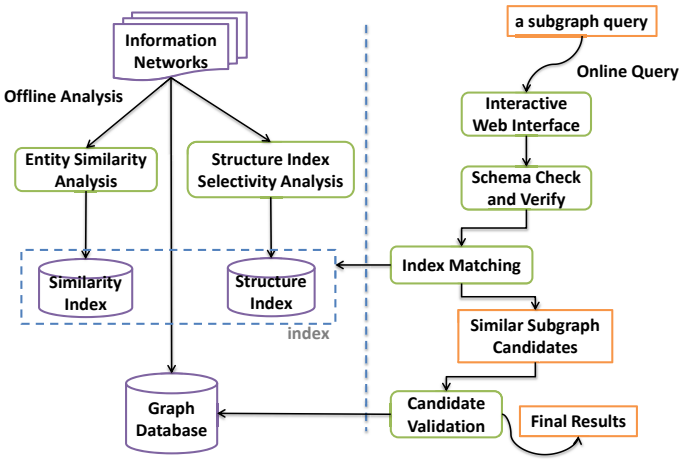

Figure 3: System Framework, where purple elements indicate off-line data flow, orange elements represent on-line data flow, and the green elements are system modules

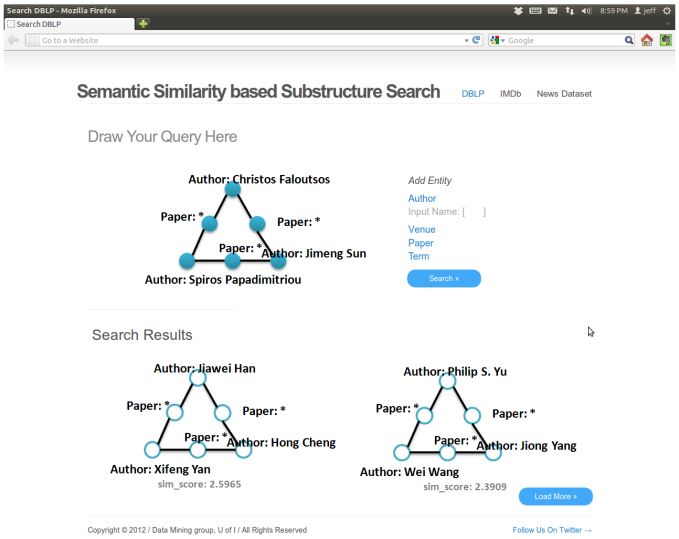

Figure 4: System Snapshot

criteria. More details of the important modules in the system are presented as follows.

3.1 Entity Similarity Analysis

Entity similarity calculation, as one of the most fundamental problems in heterogeneous information networks, has been studied from different aspects with various techniques [4] [3] [9]. Entity similarity analysis results can be used in link prediction, recommendation, and entity query systems. Previous studies [5] [8] also suggest similarity calculation along different *meta-paths* can be used to interpret different semantic meanings.

The similarity analysis module in this system measures entity similarity with different meta-path-based measurements, and stores results of popular entities as similarity index. Given an entity from the query subgraph, a set of ranked lists of entities of the same type which are semantically similar can be retrieved efficiently if indexed. The on-line index matching module will further investigate the semantic meaning of the query, select corresponding similar entities and pass the intermediate results to the following search module.

3.2 Structure Index Selectivity Analysis

Similarity index is an entity level semantic index, which can retrieve lists of similar entities with different semantic meanings, and find semantically similar entity combinations of the given query in the network efficiently. The other part of the search criteria is identical structure matching of the given subgraph query, which is a similar problem to subgraph isomorphism [7]. Subgraph isomorphism, which is a computational task in which two graphs G and H are given as input, and one must determine whether G contains a subgraph that is isomorphic to H, is NP-hard. Many recent studies [2] [11] [10] on subgraph query in graph or network propose different techniques to ease the subgraph isomorphism curse. Researchers discovered that with correct structure index over the large scale network, one can reduce the number of isomorphism checking significantly; hence reduce the overall query time.

However, structure index takes time and space to be built, and one can not index all possible subgraph structures off-line. A decision of which substructures should be indexed has to be made based on the availability of resource, selectivity of structures as well as frequency of substructures in information networks. The S^4 system takes different factors as well as the network schema into consideration when building structure index. Given a subgraph query, with proper structure index, the on-line index matching module retrieves and sorts graph locations which possibly possess the same substructure as the query. The candidate validation module verifies partial results generated from both similarity index and structure index and calculates final retrieval results.

3.3 Candidate Validation

With the assistance of similarity index and structure index, the S^4 system can calculate promising network locations and entity combinations efficiently, which are denoted as subgraph candidate. These candidates have high probability to match the structure of the subgraph identically, and have semantically similar entities as the subgraph query.

Candidate validation module is the last step of the on-line process. This module examines candidate in batch using a recursive strategy, picks up candidates which meet the criteria of the query subgraph and outputs them as the final retrieval results. The candidate validation process is similar to the subgraph isomorphism examination so the process can be very slow when the structure of subgraph query is large or complex.

To make the S^4 system response fast with most of the queries, two principles have been followed in system design. The first principle is to reduce the number of candidate validation as much as possible, which requires that in order to generate Top-K results, the number of query candidates has to be small and comparable to K. The second principle is to reduce the size of subgraph structure we need to verify as much as possible. To achieve the first principle, we need to improve the quality (selectivity) of the two indices during off-line data mining stage. To achieve the second principle, we need to analyze the subgraph query on-line, and calculate the best examine order for the recursive validation process. What's more, if similarity index and structure index can provide enough evidence for part of substructure, the system should be able to identify such scenarios and shrink the original subgraph query along with all candidates to an even smaller size, so that each validation will take less time. The S^4 system incorporates with both principles in the candidate validation module and improves the overall performance

```
N 1 author Christos Faloutsos
N 2 author Jure Leskovec
N 3 paper *
N 4 term graph &
E 1 3
E 2 3
E 3 4
```

Figure 5: Subgraph Definition Language Example

significantly.

3.4 Implementation Details

The S^4 system is designed to be scalable, efficient and user friendly. What's more, the system works well with limited resource, *e.g.*, hard drive and memory. When the hard drive is limited, the system will create smaller indices, aiming to make the most common cases faster, and search unindexed entities or structures dynamically.

Similarity index is stored in relational database, and structure index is stored in a nonsql graph database, neo4j [2]. Both back-ends and front-ends modules are written in Python with networkx [3]. The web based interactive interface is implemented using Django [4] and jquery.

4. DEMONSTRATION

We demonstrate the power of the semantically similar substructure discovery system (the S^4 system) on multiple real life heterogeneous information datasets including DBLP and IMDb, and we are working on incorporating more newly generated datasets into the system as well.

A snapshot of the web based interactive interface can be found in Figure 4. From this simple interface, users can draw the subgraph query structure with a mouse or a stylus and a touch screen, and they can input the entity type and more information through the panel on the right. Alternatively, we provide a simple subgraph definition language input interface (which is not shown in the snapshot) for users who do not like drawing. The definition language is equivalent expressive as the drawing input method. An example of the definition language of the subgraph query in Figure 1 can be found in Figure 5.

In subgraph definition language, every line, which starts with either N as node or E as edge, defines a node or an edge in the query. For node definition, a node id within the query has to be specified, and the attributes of the node should be listed next in the same line. If the users do not know the attributes or do not need the actual node content for similarity matching, they can simply use $*$ to indicate this node is a dummy node, which only contribute to the structure of the query. Edges are defined using couples of node ids. By appending ^ at the end of an edge definition, it means this edge is directed, otherwise it's undirected. Similar to $*$ and ^, the S^4 system supports two more node constraints, which are & as shown in the example, which means this entity has to be substituted in the retrieval results, and also $ which means this entity can not be changed in the retrieval results.

After query input, users need to click on the query button and the system will return a list of results, each of which has the identical structure, and shares similar semantic meanings of the query subgraph.

[2]NoSql open source graph database neo4j: http://neo4j.org/
[3]Python graph library: http://networkx.lanl.gov/
[4]Python web framework: https://www.djangoproject.com/

5. FURTHER DISCUSSION

We study and demonstrate the power of query driven discovery of semantically similar substructure, which is one of the several possible ways of exploring large-scale information networks with subgraph examples. Given a subgraph query, our system searches in an information network for identical structure match and semantically similar entity match. There can be alternative search standards, such has identical entity match but flexible structure match. This type of system can be useful when users need to query additional relationships among given entities. Also, one may prefer a more flexible search standard: given a subgraph query, return those with similar structure and similar entities. These alternative standards can be interesting and need further studies.

The prototype of S^4 system we implemented can be applied to many different scenarios, including new recommendation systems for B2C or movie rental services, multidimensional vertical similarity search engines for lawyers and journalists, expert community exploration on specific areas for scientific researchers, and even multidimensional candidates search for company recruiting.

6. ACKNOWLEDGMENTS

The work was supported in part by the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA), NSF IIS-09-05215, and U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265.

7. REFERENCES

[1] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *Proceedings of SIGKDD*. ACM, 2011.

[2] A. Khan, N. Li, X. Yan, Z. Guan, S. Chakraborty, and S. Tao. Neighborhood based fast graph search in large networks. In *SIGMOD Conference*, 2011.

[3] N. Lao and W. W. Cohen. Fast query execution for retrieval models based on path-constrained random walks. In *Proceedings of SIGKDD*, 2010.

[4] Y. Sun, R. Barber, M. Gupta, C. Aggarwal, and J. Han. Co-Author Relationship Prediction in Heterogeneous Bibliographic Networks. In *ASONAM Conference*, 2011.

[5] Y. Sun, J. Han, X. Yan, S. P. Yu, and T. Wu. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. In *VLDB*. ACM, 2011.

[6] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of SIGKDD*. ACM, 2009.

[7] J. Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 1976.

[8] X. Yu, Q. Gu, M. Zhou, and J. Han. Citation prediction in heterogeneous bibliographic networks. In *Proccedings of the 2012 SIAM Conference on Data Mining (SDM 2012)*, 2012.

[9] X. Yu, A. Pan, L. Tang, Z. Li, and J. Han. Geo-friends recommendation in gps-based cyber-physical social network. In *ASONAM Conference*, 2011.

[10] P. Zhao and J. Han. On graph query optimization in large networks. *Proceedings of the VLDB Endowment*, 2010.

[11] P. Zhao, J. Yu, and P. Yu. Graph indexing: tree+ delta<= graph. In *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007.