A conceptual modeling framework for network analytics

Qing Wang
Research School of Computer Science, The Australian National University, Canberra, ACT 0200, Australia

ABSTRACT

In this paper we propose a conceptual modeling framework for network analysis applications. Within this framework, a data model called the Network Analytics ER model (NAER) is developed, which enables us to manage and analyze network data in a unified way. In particular, not only data requirements but also query requirements can be captured by the conceptual description of network analysis applications. This unified view provides us a flexible platform to build a number of topology schemas upon the underlying core schema for supporting network analysis queries. We also discuss how the semantics of network analysis queries can be modeled at the conceptual level, and explore three possible application areas of using our analytical framework for network analysis applications: (1) governing semantic integrity, (2) improving analysis efficiency, and (3) supporting network dynamics. We believe that conceptual modeling can play an important role in managing and analyzing network data, and contribute to the development of network analytics.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Network analysis has proliferated rapidly in recent years, and it has useful applications across a wide range of fields, such as social science, computer science, biology and archeology [2,3,10,13,15,16]. One key aspect of network analysis is to understand how entities and their interaction via various (explicit or implicit) relationships take place within a network that is often represented as a graph with possibly millions or even billions of vertices. In practice, network data are often managed in a database system, e.g., Facebook uses MySQL to store data like posts, comments, likes, and pages. Network analysis queries are thus processed in two steps: (i) data construction through extracting data from the underlying database; and (ii) data analysis through using some software tools that incorporate data mining and machine learning techniques [11]. Since different fragments of network data may be of interest for different analysis purposes, network analysis queries are usually performed in ad hoc and isolated environments. Therefore, there is a divorce of data models and query languages between managing network data and analyzing network data in many real-world applications. In relation to this, several questions arise.

• Semantic integrity.
  With more and more network analysis queries being performed, it becomes increasingly important to semantically align and mine their relationships. But how can we determine whether or not two or more network analysis queries are semantically relevant and consistent?

• Analysis efficiency.
  Network analysis queries are often computationally expensive. Regardless of implementation details that different network analysis queries may have, the need to capture semantics remains. Can the efficiency of network analysis queries be improved by leveraging their semantics at the conceptual level?

An earlier version of this paper was presented at the 33rd International Conference on Conceptual Modeling (ER 2014), Atlanta, GA, USA.
E-mail address: qing.wang@anu.edu.au.
Network dynamics.

Network analysis applications are dynamic and evolving over time. Can network analysis be dynamically performed at different scales and over different time periods so as to predict trends and patterns?

The root of these questions stems from two different perspectives on networks — one is from the data management perspective (i.e., how to control data), and the other is from the data analysis perspective (i.e., how to use data). These two perspectives are closely related but have different concerns. We believe that conceptual modeling can play an important role in bridging these two perspectives, and contribute to answering the above questions. This paper aims to explore this, and in a broader sense, it also attempts to envision the role of conceptual modeling in the era of Big-data analytics since network analysis is at the core of Big-data analytics.

Example 1. Fig. 1(a) depicts a simple network in which each vertex represents an author, and each edge represents that two authors have coauthored one or more articles. Suppose that we have two network analysis queries: (1) \( Q_c \) — find the collaborative communities of authors according to how closely they collaborate with each other to write articles together, and (2) \( Q_a \) — find the top-k influential researchers.

With the results of \( Q_c \) and \( Q_a \) available (i.e., as shown in Fig. 1(a) and (b)), we may further ask: (3) \( Q_{ca} \) — what are the collaborative communities of these top-k influential researchers? (4) \( Q_{ac} \) — are these top-k influential researchers the central ties in their collaborative communities? To answer \( Q_{ca} \) and \( Q_{ac} \), we would like to know whether \( Q_c \) and \( Q_a \) are semantically consistent (i.e., use the same set of authors and articles). If they are, we can leverage the results of \( Q_c \) and \( Q_a \) to efficiently answer \( Q_{ca} \) and \( Q_{ac} \). Ideally, we would also like to analyze the changes of collaborative communities and influential researchers over time to discover unknown interactions and trends.

1.1. Contributions

The first contribution of this paper is the development of a conceptual modeling method for network analysis applications. We propose the Network Analytics ER model (NAER) that extends the concepts of the traditional ER models in three aspects: (a) the structural aspect—analytical types are added; (b) the manipulation aspect—topological constructs are added; and (c) the integrity aspect—semantic constraints are extended. Then we introduce an analytical framework for network analysis applications, which has three components: a collection of query topics, a number of small topology schemas, and a relatively large core schema. The core schema consists of base types, while topology schemas consist of analytical types that have support from base types in the core schema. A query topic is a tree representing a hierarchy of object classes with each level being built from lower levels, and the leaves of such a tree can be specified using topological constructs over analytical types in one or more topology schemas, or using base types in the core schema. In general, the core schema is large and relatively static, which describes database structures based on data requirements, whereas topology schemas are usually small and dynamic, which describe topological structures of interest based on query requirements. The reason for having small topology schemas is to support flexible abstraction on topological structures.

We develop the design guidelines of establishing such an analytical framework for network analysis applications. The key idea is that, in addition to data requirements, query requirements should also be taken into account in the modeling process. This enables an integrated view on the semantics of analysis tasks, and can thus provide a conceptual platform for sharing the theories and algorithms behind different analytical models. In doing so, such a conceptual model can circumvent the design limitations of traditional modeling techniques which do not consider analysis queries. It thus brings us several significant advantages for managing analysis tasks in networks, such as managing the complexity of computational models, handling the semantic integration of different data analysis results, and enabling comparative network analysis.

Fig. 1. (a) A simple network with collaborative communities described by dashed circles; and (b) a collection of influential researchers.
We further explore three possible application areas of using our analytical framework for network analysis applications. That is: (1) governing semantic integrity, (2) improving analysis efficiency, and (3) supporting network dynamics. Some preliminary solutions have been proposed, together with discussions on their practical implications.

1.2. Outline

The remainder of the paper is structured as follows. We start with a motivating example in Section 2. Then we introduce the NAER model in Section 3. After that, we present a high-level overview for the analytical framework of network analysis applications, and discuss the general design principles that underlie the development of such an analytical framework in Section 4. Three possible application areas are discussed in Section 5. We present the related works in Section 6 and conclude the paper in Section 7.

2. Motivating example

We start with a bibliographical network, i.e., each article is written by one or more authors, an article is published in a conference or a journal, and one article may cite a number of other articles. Using the traditional ER approaches [5, 19], one can design a simple ER diagram as depicted in Fig. 2.

Based on this network, a variety of network analysis tasks can be performed. Typical examples include: community detection [8, 10] that is to identify sets of entities that have certain common properties, cocitation analysis [4] that is to identify sets of articles that are frequently cited together, and link prediction [14] that is to find out links among entities which will probably appear in the future. We exemplify some of such analysis tasks by the following queries.

Q1 (Collaborative communities) Find the communities that consist of authors who collaborate with each other to publish articles together.

Q2 (Most influential articles) Find the most influential article of each VLDB conference, together with the authors of the article.

Q3 (Top-k influential researchers) Find the top k influential researchers in terms of the influence of articles (i.e., the citation counts) they have published.

Q4 (Correlation citation) Find the correlation groups of journals which publish articles that are often cited by each other.

Conceptually, these network analysis queries either require or generate some entities and relationships that are not explicitly represented in Fig. 2. For instance, the query Q1 generates a set of author groups, each being referred as a collaborative community, and the detection of such collaborative communities is based on the coauthorship relationships between authors, i.e., two authors have written an article together. Capturing implicit entities and relationships, and represent them explicitly in a conceptual model can bring several benefits for network analysis applications: (1) It enables semantic integrity checking across different analysis results. (2) It supports comparative analysis on different dimensions in order to predict trends and discover new insights. (3) It can improve query performance by reformulating queries in a way that can leverage existing results whenever possible. Nevertheless, how should we specify such entities and relationships? Take the query Q1 for example, the question is how to model the concept of collaborative community and the relationship of coauthorship among authors. In most cases, they are algorithmically defined, without a precise a priori definition. Motivated by these questions, we will discuss the NAER model in Section 3.

3. Network analytics ER model

Our NAER model extends the concepts of the traditional ER models in three aspects: (a) the structural aspect – analytical types are added; (b) the manipulation aspect – topological constructs are added; and (c) the integrity aspect – semantic constraints are extended.

![Fig. 2. An ER diagram.](image-url)
3.1. Base types vs analytical types

Two kinds of entities and relationships are distinguished in the NAER model: (1) base entity and relationship types contain entities and relationships, respectively, as defined in the traditional ER models; and (2) analytical entity and relationship types contain analytical entities and relationships, respectively, such that

- An analytical entity is an object of being analyzed, which may be a concrete thing or an abstract concept; and
- An analytical relationship is a link among two or more analytical entities.

Base and analytical types serve rather different purposes. Base types specify first-class entities and relationships from the data management perspective, while analytical types specify first-class entities and relationships from the data analysis perspective. These two perspectives may lead to different decisions about which entities and relationships to emphasize, and which to ignore. For example, coauthorship and author are often interesting analytical types to consider in network analysis queries like Q1, but the corresponding base types author, write and article are more natural and informative for being managed in a database system. In the NAER model, base types serve as the ground from which analytical types can be derived. Let \( B(T) \) be a set of base types in a network \( T \). Then a set \( A(T) \) of analytical types in \( T \) can be defined over \( B(T) \) such that each \( A \in A(T) \) is determined by a subset of base types in \( B(T) \), and the base types that define \( A \) are called the support of \( A \), denoted as \( supp(A) \). To ensure that analytical types are well-defined, the following criteria must be applied:

- \( supp(A) \subseteq B(T) \) for each analytical type \( A \); and
- \( supp(A_\text{r}) \subseteq supp(A_\text{e}) \) for each analytical relationship type \( A_\text{r} \) and every analytical entity type \( A_\text{e} \) that participates in \( A_\text{r} \).

An analytical type \( A \) may have attributes, each of which must be derivable from the base types in its support \( supp(A) \). To avoid redundant information, it is prohibited to have attributes in an analytical type as a copy of some attributes in base types. A (topology or support) schema \( S \) consists of a set of connected and well-defined entity and relationship types that are complete, i.e., if a relationship type \( T_\text{r} \in S \), then for every type \( T \) that participates in \( T_\text{r} \), we have \( T \in S \).

Example 2. Suppose that all the entity and relationship types in Fig. 2 are base types in the network \( T_\text{bib} \), then we can define several analytical types over these base types, as depicted in Fig. 3(a)–(c), i.e.,

- \( B(T_\text{bib}) = \{ \text{author}, \text{article}, \text{conference}, \text{journal}, \text{write}, \text{cite}, \text{publish} \} \); and
- \( A(T_\text{bib}) = \{ \text{author}^\ast, \text{coauthorship}, \text{article}^\ast, \text{citation}, \text{journal}^\ast, \text{cocitation} \} \).

Both coauthorship and cocitation may have an attribute weight, which indicate how many articles two authors have written together, and how many times two journals are cocited by articles, respectively. The analytical types in \( A(T_\text{bib}) \) have the following support:

- \( supp(\text{author}^\ast) = \{ \text{author} \} \) and \( supp(\text{coauthorship}) = \{ \text{author}, \text{article}, \text{write} \} \); and
- \( supp(\text{article}^\ast) = \{ \text{article} \} \) and \( supp(\text{citation}) = \{ \text{article}, \text{cite} \} \); and
- \( supp(\text{journal}^\ast) = \{ \text{journal} \} \) and \( supp(\text{cocitation}) = \{ \text{article}, \text{cite}, \text{journal}, \text{publish} \} \).

3.2. Topological constructs

A common scenario in network analysis is to analyze topological structures that are hidden underneath base entities and relationships. To explicitly represent a topological structure of interest, one can define analytical entities as vertices and analytical relationships as edges in a graph that may be directed or undirected, weighted or unweighted. However, as illustrated by the following example, base and analytical types alone are still not sufficient to provide a clearly defined conceptual description for network analysis applications.

Example 3. To analyze collaborative communities as described in the query Q1, we may design the coauthorship schema \( S_{\text{co}} \), i.e., Fig. 3(a), consisting of the analytical entity type \( \text{AUTHOR}^\ast \) and the analytical relationship type \( \text{COAUTHORSHIP} \). Nevertheless,
the problem of how to model the concept of collaborative community in terms of $S_{co}$ still remains. Solving this problem requires us to take into account topological measures and constructs, together with analytical types.

Topological measures play an important role in characterizing topology properties of a network \[2,12\]. Two of the most commonly used topological measures are centrality and similarity. Let $S$ be a topology schema and $A$ be an analytical type in $S$.

- Cent: $A \rightarrow \mathbb{N}$ over $S$ is a centrality measure that describes the centrality of elements of $A$ in a network of $S$, and return a rank $\text{CENT}(v)$ for an element $v$. This measure can be implemented in different ways, such as degree, betweenness and closeness centrality \[9\].
- Simi: $A \times A \rightarrow \mathbb{N}$ over $S$ is a similarity measure that describes the similarity between two elements of $A$ in a network of $S$, and generates a rank $\text{Simi}(v_1, v_2)$ for a pair $(v_1, v_2)$ of elements. This measure can also be implemented in different ways, such as q-gram, adjacency-based and distance-based similarity \[8\].

Based on topological measures, we introduce two families of topological constructs in the NAER model — clustering and ranking. Let $S$ be a schema, $A \in S$ be an analytical type, and $m$ be a topological measure. Then we have

1. CLUSTER-BY $(S, A, m)$ that contains a set of clusters over $A$, according to the structure specified by $S$ and the measure $m$; and
2. RANK-BY $(S, A, m)$ that contains a set of ranked elements over $A$, according to the structure specified by $S$ and the measure $m$.

A CLUSTER-BY construct classifies a set of elements over $A$ into a set of clusters (i.e., each cluster is a set of elements), while a rank-by construct assigns rankings to a set of elements over $A$. Both cluster-by and rank-by constructs need to be augmented with a topological measure. We consider the implementation of a topological measure as a black-box. In doing so, these topological constructs provide us an ability to specify existing prominent techniques of network analysis into the conceptual modeling process without being exposed to low-level implementation details. Take the seminal work of Girvan and Newman \[10\] for example, we may utilize their well-known algorithm (denoted as given-newman) to characterize the community structure of $A$ in a network described by $S$ as follows:

$$\text{CLUSTER-BY}(S, A, \text{GIVEN—NEWMAN}).$$

**Example 4.** Consider the following concepts relating to the queries Q1–Q4.

- **Collaborative community** in the query Q1 can be modeled using

  $$\text{CLUSTER-BY}(S_{co}, \text{AUTHOR'}, \text{CENT—CLOSENESS}).$$

  That is, each collaborative community is a group of authors in a network specified by $S_{co}$, and the measure for determining community membership is closeness centrality.

- **Influence of article** in the queries Q1–Q2 can be modeled using

  $$\text{RANK-BY}(S_{co}, \text{ARTICLE'}, \text{CENT—INDEGREE}).$$

  That is, each article is associated with a ranking that indicates its influence in terms of a network specified by $S_{co}$, and the measure for determining rankings is indegree centrality.

- **Correlation group** in the query Q4 can be modeled using

  $$\text{CLUSTER-BY}(S_{jo}, \text{JOURNAL'}, \text{CENT—BETWEENNESS}).$$

  That is, each correlation group contains journals that are correlated in a network specified by $S_{jo}$ and the measure for determining the correlation among journals is betweenness centrality.

In a nutshell, the use of topological constructs is to describe topological structures of interest based on topology schemas. Since a topology schema contains only analytical types, it serves as a mechanism for separation of concerns, which conceptually separates base types used for the data management purpose from analytical types used for the data analysis purpose. This thus brings many benefits provided by separation of concerns, such as reduced complexity and improved reusability. In Section 5.2, we will further discuss how to leverage the connections between base and analytical types to improve the efficiency of network analysis.

3.3. Integrity constraints

The NAER model can support integrity constraints from the traditional ER models, and also allows extending them to analytical entity and relationship types in a similar manner. In addition to these, integrity constraints can also be defined over topological constructs, and some of such constraints are as follows:

- **DISJOINT (resp. overlapping)** constraints on CLUSTER-BY.

  Clusters identified by a cluster-by construct must be disjoint, i.e., no element can be a member of more than one cluster, (resp. can be overlapping).
• **CONNECTED constraints on CLUSTER-BY.**
  For each cluster identified by a cluster-by construct, there is a path between each pair of its members, running only through elements of the cluster.

• **EDGE-DENSITY constraints on CLUSTER-BY.**
  For each cluster identified by a cluster-by construct, its members have more edges inside the cluster than edges with other members who are outside the cluster.

• **TOTAL (resp. partial) constraints on RANK-BY.**
  Every element of a given analytical type must be (resp. may not necessarily be) ranked by a rank-by construct.

4. Analytical framework

In this section, we discuss how to use the NAER model to establish an analytical framework for network analysis applications at the conceptual level.

4.1. High-level overview

Fig. 4 illustrates an analytical framework of the bibliographical network described in our motivating example. In general, such an analytical framework has three components \( (S_q, S_t, S_c) \): (1) a collection of query topics \( S_q \), (2) a number of small topology schemas \( S_t \), and (3) a relatively large core schema \( S_c \). The core schema \( S_c \) contains a set of base types. Each topology schema \( S_t \) contains a set of analytical types, and the support of each analytical type in \( S \) is a subset of base types in \( S_c \). Each query topic in \( S_q \) is a tree representing a hierarchy of object classes with each level being built from lower levels, and the leaves of such a tree can be specified using topological constructs CLUSTER-BY or RANK-BY over one or more topology schemas, or using the core schema if the attributes of base types need to be processed.

In Fig. 4, three topology schemas \( \{S_{co}, S_{ci}, S_{jo}\} \) are built upon the core schema, which represent three topological structures that are of interest for network analysis queries over \( \Upsilon_{bib} \): (1) the coauthorship schema \( S_{co} \) for query Q1, (2) the citation schema \( S_{ci} \) for the queries Q2 and Q3, and (3) the cocitation schema \( S_{jo} \) for query Q4. Consequently, the four queries Q1–Q4 discussed in Section 2 lead to four query topics, in which the query topics of the queries Q2 and Q3 are overlapping and having the same leave “Influence of article” (will be discussed in detail in Section 4.3).
4.2. Design principles

We now present the design guidelines that support the development of an analytical framework for network analysis applications. The central idea is to incorporate both data and queries into the conceptual modeling process. Generally, there are five steps involved:

1. Identify data requirements (i.e., a set of business rules of interest);
2. Design the core schema based on the data requirements;
3. Identify query requirements (i.e., a set of analysis queries of interest);
4. Design topology schemas based on the query requirements; and
5. Identify constraints on the query topics, and core and topology schemas.

The steps (1) and (2) are exactly the same as in the traditional ER models, the steps (3) and (4) are additional but critical for network analysis applications, and the step (5) extends integrity constraints of the traditional ER models to analytical types in topology schemas and topological constructs in query topics accordingly.

4.3. Key issues

Now we discuss three key issues in designing this analytical framework: (i) what are data and query requirements? (ii) How are query requirements and query topics related? (iii) How are the core and topology schemas designed?

4.3.1. Data and query requirements

Data and queries are two different kinds of requirements. Data requirements describe what information an application should manage, while query requirements describe how the information of an application should be used. Although our NAER model can conceptually represent both data and query requirements for network analysis applications, the questions to be clarified are: (a) do we need to consider all queries? (b) If not, what are the query requirements of interest?

Queries in network analysis applications may exist in various forms. For example, database queries in the traditional sense, such as “find all journal articles published in 2013”, often use a database language (e.g., SQL) to process data, and analysis queries from a topological perspective, such as the queries Q1 and Q4, often use certain data mining and machine learning techniques to process data. In a nutshell, database queries and analysis queries are fundamentally different in two respects:

- **Logical vs topological**: Database queries are concerned with the logical properties of entities and relationships, while analysis queries focus on the topological properties of entities and relationships. In most cases, analysis queries are formulated using software tools in a much more complicated way than database queries.
- **Indefinite vs definite**: Analysis queries often have indefinite answers, which depends on not only the underlying structure but also the choice of topological measures. It can be difficult to know which measure is better than the others, and which answer is optimal. In contrast, database queries have definite answers that are determined by the underlying structure in a database.

In many real-life applications, analysis and database queries are commonly combined to find useful information [17]. For example, query Q2 can be viewed as the combination of an analysis query “find the most influential articles” and a database query “find articles of each VLDB conference, together with the authors of the article”. When designing a conceptual model for network analysis applications, we are more interested in analysis queries. There are two reasons: (1) analysis queries are often more computationally expensive, and modeling analysis queries are likely to improve performance significantly; and (2) analysis queries are often processed in isolation, and modeling analysis queries can help maintain their semantic integrity. Nevertheless, the ultimate goal of this analytical framework is to provide a conceptual design for supporting any meaningful queries in network analysis applications, regardless of what kinds of queries they are.

4.3.2. Query requirements and topics

After identifying query requirements, i.e., queries of interest, we need to analyze these queries to understand their semantics and required computations. Analyzing queries is to unravel the structures of queries, which has at least two aspects to consider: (1) the structure of a query, and (2) the structure among a set of queries. Since queries may be described in various syntactical forms, here we focus on exploiting the semantic structures of queries. For each query Q, we associate it with a query topic t(Q), which is a tree with each edge being an object class, and an edge from a node C1 to a node C2 expressing that C1 depends on C2. This dependence relation between object classes is closed under transitivity, i.e., if C1 depends on C2 and C2 depends on C3, then C1 depends on C3. The query topic of a query can be defined at a flexible level of abstraction. That is, the level of granularity for nodes in a query tree is a design choice depending on individual applications. For each query Q, we thus have a set of object classes that are in one-to-one correspondence with the nodes of t(Q). A node C1 ∈ t(Q) in one query topic may have certain relationships with a node C2 ∈ t(Q) in a different query topic. Such relationships include that: (1) C1 depends on C2; or (2) C1 and C2 are the same. Nevertheless, it is impossible that C1 depends on C2, and meanwhile C2 depends on C1 or any of its descendant nodes. If two different query topics t(Q1) and t(Q2) contain the same node C, then t(Q1) and t(Q2) are connected by the node C, and the trees are merged.
Example 5. Consider the queries Q1–Q4 in our motivating example. We have one query topic for each of the queries as depicted in Fig. 5(a). The trees corresponding to these query topics are merged as depicted in Fig. 5(b).

4.3.3. Core and topology schemas

For a network analysis application, the design of its core schema and topology schemas are carried out in two steps. First, the core schema is designed based on data requirements as in the traditional ER models. Second, the topology schemas are designed based on query requirements. When designing the topology schemas, the leaves of query topics that are associated with query requirements are grouped. More specifically, the leaves are grouped in terms of what analytical types they need for analysis, and each of such groups correspond to one topology schema. In general, the central idea is that all data requirements should be captured by the core schema, and the analysis part of all query requirements should be captured by a collection of topology schemas.

Example 6. For the query topics of Q1–Q4, we can group their leaves as shown in Table 1, and design three topology schemas \( \{S_{co}, S_{ci}, S_{jo}\} \) as described in Fig. 4.

One distinguished feature of topology schemas is that, rather than taking objects in all their complexity, topology schemas only focus on specifying a simple but concise representation for objects. Therefore, topology schemas need to be designed in accordance with the following criteria:

1. **Topology schemas should be small.** Topology schemas are the basic building blocks of supporting analysis queries. The smaller topology schemas are, the easier they can be composed to support flexible modeling needs.
2. **Topology schemas should be dynamic.** Query requirements may be changing over time. Correspondingly, topology schemas need to be adaptive enough to reflect the dynamics of query requirements.

Two topology schemas in an analytical framework may be overlapping. In fact, certain degree of overlapping can facilitate comparative analysis over different topology schemas. Nevertheless, duplicate topology schemas should be avoided because this would cause redundant storage and inconsistence. The following example shows that our analytical framework supports an integrated and coherent view on the core and topology schemas.

### Table 1

<table>
<thead>
<tr>
<th>Query</th>
<th>Core schema</th>
<th>Topology schemas</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( S_{co} )</td>
</tr>
<tr>
<td>Q1</td>
<td>VLDB article</td>
<td>Collaborative community</td>
</tr>
<tr>
<td>Q2</td>
<td>Researcher</td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 7. The three topology schemas \( S_{co}, S_{ci}, S_{jo} \) can be composed by leveraging base types in the core schema. Fig. 6 shows three possible compositions: (a) three topology schemas are composed by an analytical relationship type has that is determined by several base types; and (b) two schemas are composed by a base relationship type (i.e., publish and write) directly.

5. Applications

In this section we discuss how the proposed analytical framework can be used to support network analysis applications, with the aim of answering the questions raised earlier in Section 1.

5.1. Semantic integrity

Given a number of queries that are processed within the same analytical framework, i.e., their corresponding topology schemas may be different but must be built upon the same core schema, a question that commonly arises is whether the analysis results of these queries are semantically consistent or relevant. To answer this question, we need to further exploit the semantic relationships among queries in terms of which analytical types are involved in the queries, and what are the supports of these analytical types. More specifically, there are two aspects we need to consider:

- How are queries formalized over topology schemas? For example, which analytical types are used, and how topological constructs are defined over such analytical types?
- How are analytical types built upon the core schema? For example, are they defined over base types using generalization, specification or aggregation as in the traditional ER model?

Let \( Q \) be a query. We define the base of \( Q \), denoted as \( \text{base}(Q) \), to be the set of base types (or their fragments) that support at least one analytical type used in \( Q \).

- Two analysis queries \( Q \) and \( Q' \) are **semantically equivalent** if they have the same base, and their query topics are the same. Different from traditional database queries for which two semantically equivalent queries should always return the same result, two semantically equivalent queries in network analysis applications may not necessarily yield the same analysis results. This is because different topological measures can be used in these queries.
- Two analysis queries \( Q \) and \( Q' \) are **semantically consistent** if they have the same base, regardless whether or not their query topics are the same. Intuitively, two semantically consistent queries mean that their analysis results are based on exactly the same data sources, consistently representing possibly different properties of the same network data. If two queries are semantically equivalent, then they must also be semantically consistent, and conversely it does not hold.
- Two analysis queries \( Q \) and \( Q' \) are **semantically relevant** if they have different but overlapping bases, and their query topics are different. Two semantically relevant queries are related to each other using some base types in the core schema. Therefore, although two queries may correspond to two different topology schemas, they can still have certain semantic relevance through base types.

Fig. 7 illustrates how the notions of semantically equivalent, consistent and relevant queries are related. Fig. 8 depicts four cases of possible interaction between the bases of two queries. In Fig. 8(a), two queries have the same base, and may be semantically...
equivalent if their query topics are also the same, or semantically consistent otherwise. In Fig. 8(a)–(b), two queries have the overlapping bases and they are semantically relevant. In Fig. 8(d), two queries have disjoint bases so that they are not semantically relevant.

In the following, we present several examples explaining the notions of semantically equivalent, consistent and relevant queries.

Example 8. (Semantically equivalent queries) Consider two queries about collaborative communities, where one is modeled as

\[ \text{CLUSTER-BY}(S_{c}, \text{AUTHOR}, \text{CENT-CLOSENESS}) \]

and the other is modeled as

\[ \text{CLUSTER-BY}(S_{c}, \text{AUTHOR}, \text{CENT-BETWEENNESS}) \]

These two queries are semantically equivalent. Nevertheless, they may produce different results due to the use of different topological measures.

Example 9. (Semantically consistent queries) Consider the query \( Q_1 \) and the following two queries \( Q_5 \)–\( Q_6 \):

\[ Q_5 \text{ Find the top k most collaborative authors in each collaborative community in terms of their centrality in the collaborative community; and } \]
\[ Q_6 \text{ Find the top k most collaborative authors in terms of their centrality in the whole collaborative network. } \]

These three queries have the same base but different semantics. Although they are not semantically equivalent, they are semantically consistent.

Example 10. (Semantically relevant queries) Consider the queries \( Q_1 \)–\( Q_4 \), which correspond to three different topology schemas. However, the bases of these queries are overlapping, i.e., they all contain the base type article. Therefore, these four queries are semantically relevant.

5.2. Analysis Efficiency

In real-world applications, network analysis queries are often computationally expensive, which limits their applicability, particularly when network data are massive and complex. In light of previous studies in the traditional database theory, the next important question is “can we improve the efficiency of network analysis queries by leveraging their semantics at the conceptual level?”

In principle, instead of processing each analysis query from scratch, analysis queries that have been issued to the network data should be stored in the analytical framework, together with their analysis results. Then these analysis queries form a pool of recourses and become available for being reused for processing other queries. In doing so, the analysis efficiency of queries can be improved through the use of appropriate existing analysis results for constructing new results of queries in process. This however requires us to develop a query search engine that can: (a) understand the semantics of queries at the conceptual level, (b) search similar queries over a pool of recourses, and (c) combine “pieces” from similar queries to process the desired query. Nevertheless, developing such a query search engine is by no means easy. Three challenging problems, each corresponding to one of the above areas, are as follows:

- **Query modeling**: The semantics of queries needs to be modeled such that each query is represented by a pair \((t, r)\) consisting of a query topic \(t\), which is tree-structured as discussed in Section 4.3, and a query answer \(r\) that corresponds to the analysis result of the query.
- **Query matching**: Queries are matched in terms of how similar their semantics are. More specifically, given two queries \( Q_1 = (t_1, r_1) \) and \( Q_2 = (t_2, r_2) \), they are similar if their query topics are similar, e.g., \( t_1 \) and \( t_2 \) have a common subtree, and/or their analytical types are overlapping.
• **Query composition**: With similar queries found, we want to compose the analysis result of the desired query, which is possibly partial, based on the similarity of queries and the availability of their analysis results.

Different from query optimization studied in the traditional database context, such a query search engine can optimize queries by reusing existing topology schemas. Given a query $Q$ to be processed, there are three cases: (1) if the topology schemas of queries that are similar to $Q$ can be composed to form the topology schema required by $Q$, then there is no need to create a new topology schema for $Q$. This means that processing $Q$ can completely based on the network data over the topology schemas of the similar queries. (2) If the topology schema required by $Q$ can only be obtained by extending some of the existing topology schemas, these topology schemas are replaced by their extended ones. Accordingly, processing $Q$ is partially based on the network data over the topology schemas of the similar queries. (3) In other cases, a new topology schema for $Q$ has to be created in the analytical framework, and the network data required by $Q$ are entirely constructed from the underlying database. In a nutshell, the improvement of analysis efficiency depends on how the desired query is semantically similar to the existing queries modeled in the same analytical framework. Generally, the more similar it is to the existing queries, the more likely its analysis efficiency can be improved.

**Example 11.** Let us consider the queries $Q_1$, $Q_5$ and $Q_6$ again. As depicted in Fig. 9, these queries can be described by their query topics. Then, based on their query topics and corresponding analytical types, the query search engine calculates their similarities so as to improve the efficiency of query processing.

Now suppose that these three queries are issued in the order of $Q_1 < Q_5 < Q_6$ (i.e., $Q_1$ is the first, and $Q_6$ is the last). After $Q_1$ is processed, its query topic and answer are stored in the analytical framework. Then, when $Q_5$ is issued, the query search engine finds that $Q_1$ is similar to $Q_5$ because they both have “collaborative community” in their query topics and also need the same analytical types. As a result, $Q_5$ can be processed by leveraging the existing analysis result of $Q_1$, and does not need to repeatedly compute collaborative communities over $S_{co}$. Similarly, when $Q_6$ is issued, the query search engine finds that both $Q_1$ and $Q_5$ are similar to $Q_6$ because they use the same analytical types for analyzing data. Thus, $Q_6$ can be processed directly over $S_{co}$ without creating any new topology schema, and the data construction step is eliminated.

### 5.3. Network dynamics

Network analysis applications are “dynamic” by nature, due to factors such as the rapid growth of data, the frequent changes of requirements as well as newly emerged analysis techniques. It is thus desired that analysis queries can be performed at different scales and over different time periods so as to predict trends and patterns. To cope with this, the analytical framework for network analysis applications should be able to support a variety of analysis strategies for time-sensitive analysis of network data. Accordingly, we need to extend the modeling of queries by specifying the types of queries, and in the meantime associate each query with a timestamp to indicate when its analysis result is generated or most recently updated.

Formally, let $Q$ be a set of all possible queries in the analytical framework, and $R$ be a set of all possible query answers. Then each query $Q \in Q$ is associated with two additional functions in dynamic networks:

1. $\tau : Q \mapsto TP$ where $TP = \{static, dynamic, periodic\}$ refers to the types of queries; and
2. $\pi : R \mapsto TS$ where $TS$ refers to an infinite set of timestamps.

![Fig. 9. Three queries Q1, Q5 and Q6 in the same analytical framework.](image-url)
Thus, queries in dynamic networks can be classified as being static, dynamic or periodic:

- Each static query Q (i.e., $\tau(Q) = static$) has exactly one query answer $r$ that is never changeable and the timestamp $\pi(r)$ is fixed.
- Each dynamic query Q (i.e., $\tau(Q) = dynamic$) has exactly one query answer $r$ that is changeable and the timestamp $\pi(r)$ can be updated.
- Each periodic query Q (i.e., $\tau(Q) = periodic$) has a sequence $\langle r_1, r_2, \ldots \rangle$ of query answers, and their timestamps are ordered in time such that $\bigwedge_{1 \leq i < j \leq t} \pi(r_i) < \pi(r_j)$, and $\pi(r_{i+1}) = \pi(r_i) = \varepsilon$ for a specified time interval $\varepsilon$ and all $i \geq 1$.

Note that, the implementation methods of a dynamic query Q may vary, which can be incremental in some cases, depending on whether Q satisfies the following property:

$$Q(I_3) = Q(I_1) \cup Q(I_2) \quad \text{if} \quad I_3 = I_1 \cup I_2,$$

where $I_1$, $I_2$ and $I_3$ are different network data over the same topology schemas required by Q.

To predict trends and patterns, comparative queries that are performed over a series of time-sensitive analysis queries are often important. Essentially, comparative queries are higher-order analysis queries that allow us to discover some interesting trends or patterns which cannot be discovered by normal analysis queries. In other words, we model analysis queries as (and their analysis results) as temporal objects, on which techniques for time series analysis can be further applied to discover trends and patterns over time.

Example 12. Consider the following comparative query:

\[
Q_7 \quad \text{“Find the top k collaborative communities that have been fastest-growing over the past ten years”}.
\]

For this query, a series of time-sensitive analysis queries about collaborative communities is needed. We can specify the query Q1 as being periodic and the time interval is two years as illustrated in Fig. 10, where $r_{(1,1)} - r_{(1,6)}$ indicates the query answers of Q1 that are processed over the collaborative networks containing authors and their coauthorship relationships up to year 2004–year 2014, respectively. Based the series of the query answers of Q1, the comparative query Q7 can be further processed, which can yield the query answer $r_7$.

6. Related works

Recently, a number of works have proposed to use database technologies for managing and analyzing network analysis [6,7,20]. However, they have mostly focused on designing logical data models and their corresponding query languages for supporting network analysis. So far, only very limited work has considered the design process of conceptual modeling [1]. In general, the previous works on modeling network analysis applications at the logical level fall into two lines of research:

1. Extending traditional database technologies (i.e., the relational model and SQL) to support data mining algorithms, such as SiQL [20] and Oracle Data Miner.
2. Extending object-oriented or graph database technologies to incorporate graph-theoretic and data mining algorithms, such as GOQL [18], and other works discussed in the survey paper [21].

Our work in this paper focused on the conceptual modeling of network analysis, and leaves the transformation to a logical model (e.g., the relational model, a graph model or a combination of several data models) as a decision of the user. For example, in [17], a hybrid memory and disk engine was developed for evaluating queries, which maintains topological structures in memory while the data is stored in a relational database. An analytical framework designed in our work can be well transformed into this data model and be implemented over the hybrid engine by separating topological structures specified by topology schemas from the database structure specified by the core schema.
7. Conclusions

In this paper, we proposed the NAER model and a conceptual modeling paradigm that incorporates both data and query requirements of network analysis. This was motivated by the rapid growth of network analysis applications. Such a conceptual view of network analysis applications can enable us to better understand the semantics of data and queries, and how they interact with each other. In doing so, we can avoid unnecessary computations in network analysis queries and support comparative network analysis in a dynamical modeling environment. As a proof-of-concept, we are currently developing a conceptual framework for network analysis applications which implements the design guidelines and concepts described in this paper. In the future, we plan to further study the techniques for conceptual query modeling, and develop a query search engine for processing queries based on their semantics.

References


Qing Wang received her Ph. D. in Computer Science from Christian-Albrechts-University Kiel, Germany, in 2010. She has worked in the IT industry for over ten years. From 2009 to 2012, she worked as an Information System Analyst at the Research Division, University of Otago, New Zealand. In April 2012, she joined the Research School of Computer Science, The Australian National University, Australia. Her research interests include database theory and applications, data mining, knowledge reasoning, and conceptual modeling.