THESE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de

L'UNIVERSITÉ LUMIÈRE LYON 2

École Doctorale : ED 512

Informatique et Mathématiques

Discipline : Informatique

Soutenue publiquement le 27 septembre 2016, par :

Wararat JAKAWAT

Graphs enriched by Cubes (GreC): a new approach for OLAP on information networks /

(Graphes enrichis par des Cubes (GreC) : une nouvelle approche pour l'OLAP sur des réseaux d'information)

Devant le jury composé de :

Esteban ZIMANYI, Professeur d'université, Université libre de Bruxelles, Président

Fatma BOUALI, Professeure des universités, Université Lille 2, Rapporteure

Olivier TESTE, Professeur des universités, Université Toulouse 2, Rapporteur

Guillaume CABANAC, Maître de conférences Université Toulouse 3, Examinateur

Sabine LOUDCHER, Professeure des université, Université Lumière Lyon 2, Directrice de thèse

Cécile FAVRE, Maître de conférences, Université Lumière Lyon 2, Co-Directrice

THÈSE

pour obtenir le grade de

Docteur en Informatique

présentée par

Wararat Jakawat

Graphs enriched by Cubes (GreC): a new approach for OLAP on information networks

(Graphes enrichis par des Cubes (GreC) : une nouvelle approche pour l'OLAP sur des réseaux d'information)

préparée au sein du laboratoire

ENTREPÔTS, REPRÉSENTATION
& INGÉNIERIE des CONNAISSANCES

sous la direction de **Sabine Loudcher**

et le co-encadrement de **Cécile Favre**

Soutenue publiquement le 27 Septembre 2016 devant le jury:

Fatma Bouali	Rapportrice	(PR, Université Lille 2)
Olivier Teste	Rapporteur	(PR, Université Toulouse 2 Jean-Jaurès)
Guillaume Cabanac	Examinateur	(MCF, Université Toulouse 3 Paul Sabatier)
Esteban Zimányi	Examinateur	(PR, Université Libre de Bruxelles)
Cécile Favre	Co-encadrante	(MCF, Université Lumière Lyon 2)
Sabine Loudcher	Directrice	(PR, Université Lumière Lyon 2)

Abstract

Online Analytical Processing (OLAP) is one of the most important technologies in data warehouse systems, which enables multidimensional analysis of data. It represents a very powerful and flexible analysis tool to manage within the data deeply by operating computation. OLAP has been the subject of improvements and extensions across the board with every new problem concerning domain and data; for instance, multimedia, spatial data, sequence data and etc. Basically, OLAP was introduced to analyze classical structured data. However, information networks are yet another interesting domain. Extracting knowledge inside large networks is a complex task and too big to be comprehensive. Therefore, OLAP analysis could be a good idea to look at a more compressed view. Many kinds of information networks can help users with various activities according to different domains. In this scenario, we further consider bibliographic networks formed on the bibliographic databases. This data allows analyzing not only the productions but also the collaborations between authors. There are research works and proposals that try to use OLAP technologies for information networks and it is called Graph OLAP. Many Graph OLAP techniques are based on a cube of graphs.

In this thesis, we propose a new approach for Graph OLAP that is graphs enriched by cubes (GreC). In a different and complementary way, our proposal consists in enriching graphs with cubes. Indeed, the nodes or/and edges of the considered network are described by a cube. It allows interesting analyzes for the user who can navigate within a graph enriched by cubes according to different granularity levels, with dedicated operators. In addition, there are four main aspects in GreC. First, GreC takes into account the structure of network in order to do topological OLAP operations and not only classical or informational OLAP operations. Second, GreC has a global view of a network considered with multidimensional information. Third, the slowly changing dimension problem is taken into account in order to explore a network. Lastly, GreC allows data analysis for the evolution of a network because our approach allows observing the evolution through the time dimensions in the cubes.

To evaluate GreC, we implemented our approach and performed an experimental study on a real bibliographic dataset to show the interest of our proposal. GreC approach includes different algorithms. Therefore, we also validated the relevance and the performances of our algorithms experimentally.

Keywords: Online Analytical Processing (OLAP), Information networks, Bibliographic data, Data cube, Graph database.

Résumé

L'analyse en ligne OLAP (Online Analytical Processing) est une des technologies les plus importantes dans les entrepôts de données, elle permet l'analyse multidimensionnelle de données. Cela correspond à un outil d'analyse puissant, tout en étant flexible en terme d'utilisation pour naviguer dans les données, plus ou moins en profondeur. OLAP a été le sujet de différentes améliorations et extensions, avec sans cesse de nouveaux problèmes en lien avec le domaine et les données, par exemple le multimedia, les données spatiales, les données séquentielles, etc. A l'origine, OLAP a été introduit pour analyser des données structurées que l'on peut qualifier de classiques. Cependant, l'émergence des réseaux d'information induit alors un nouveau domaine intéressant qu'il convient d'explorer. Extraire des connaissances à partir de larges réseaux constitue une tâche complexe et non évidente. Ainsi, l'analyse OLAP peut être une bonne alternative pour observer les données avec certains points de vue. Différents types de réseaux d'information peuvent aider les utilisateurs dans différentes activités, en fonction de différents domaines. Ici, nous focalisons notre attention sur les réseaux d'informations bibliographiques construits à partir des bases de données bibliographiques. Ces données permettent d'analyser non seulement la production scientifique, mais également les collaborations entre auteurs. Il existe différents travaux qui proposent d'avoir recours aux technologies OLAP pour les réseaux d'information, nommé "graph OLAP". Beaucoup de techniques se basent sur ce qu'on peut appeler cube de graphes.

Dans cette thèse, nous proposons une nouvelle approche de "graph OLAP" que nous appelons "Graphes enrichis par des Cubes" (GreC). Notre proposition consiste à enrichir les graphes avec des cubes plutôt que de construire des cubes de graphes. En effet, les noeuds et/ou les arêtes du réseau considéré sont décrits par des cubes de données. Cela permet des analyses intéressantes pour l'utilisateur qui peut naviguer au sein d'un graphe enrichi de cubes selon différents niveaux d'analyse, avec des opérateurs dédiés. En outre, notons quatre principaux aspects dans GreC. Premièrement, GreC considère la structure du réseau afin de permettre des opérations OLAP topologiques, et pas seulement des opérations OLAP classiques et informationnelles. Deuxièmement, GreC propose une vision globale du graphe avec des informations multidimensionnelles. Troisièmement, le problème de dimension à évolution lente est pris en charge dans le cadre de l'exploration du réseau. Quatrièmement, et dernièrement, GreC permet l'analyse de données avec une évolution du réseau parce que notre approche permet d'observer la dynamique à travers la dimension temporelle qui peut être présente dans les cubes pour la description des noeuds et/ou arêtes.

Pour évaluer GreC, nous avons implémenté notre approche et mené une étude expérimentale sur des jeux de données réelles pour montrer l'intérêt de notre approche.

L'approche GreC comprend différents algorithmes. Nous avons validé de manière expérimentale la pertinence de nos algorithmes et montrons leurs performances.

Acknowledgments

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and motivation

In the recent years, data warehousing has experienced an unprecedent and has been the backbone of decision support systems [CD97]. It has been widely accepted and used in variety of application domains, such as manufacturing industry, transportation, telecommunications, e-commerce, insurance, healthcare, education, research and government. One of the most important technologies in data warehouse systems enabling multidimensional analysis of data is Online Analytical Processing (OLAP) [CD97, Tho02, KR02, KR11]. OLAP is a very powerful and flexible tool to explore and analyze data deeply by operating computation. OLAP has seen improvements and extensions across with every new problem of domain and data, for instance, multimedia, spatial data, sequence data and etc. Given the underlying data, a cube can be created to provide a multi-dimensional and multi-level view. Traditionally, a data cube contains cells that include measures, which are valued based on a set of dimensions. Dimensions can be seen as analysis axes and may be organized into hierarchies with several levels. Dimension hierarchies make it possible to obtain views of data at different granularity levels, i.e., summarized or detailed through roll-up and drill-down operations, respectively. Basically, measures are numerical indicators which are calculated by aggregating data. This allows analyze data from different perspectives and with multiple granularities. Traditional OLAP was used to analyzing structured data. However, in recent years, more and more data sources have been represented as heterogeneous networks, in which there are multiple object and link types that have multiple attributes. Not only objects are important and interesting but also the interacting relationships among them.

Over the last few years, information networks have been quickly increasing due to the popular use of Web, blogs and various kinds of online databases. The importance of information networks is gaining increasing attention from research scientists. These networks play a crucial role in how we obtain information, how we conduct information to one another, and how we interact with other objects. Many information networks can help users with various activities according to different domains. In this scenario, we further consider bibliographic networks formed on the bibliographic databases such as the DBLP Bibliography[1], ACM Digital library [2] and etc. These databases cover all researchers publishing papers in various venues (e.g., conferences, journals, etc.), and their collaboration information for different conferences. Therefore, bibliographic data is useful for different purposes including collaborations networking, information sharing, discovery of new research topics or any combination of these in order to recommending a new reviewer, making or contacting researchers interested and online purchasing. Finally, these bibliographic databases provide a richfulness data sources in the context of Scientometrics that is the study of the quantitative features and characteristics of science and scientific research [Van97]. We not only obtain textual information from this data, but also have accessed networked data such as co-authorships network, citations network and etc.

Conceptually information networks are characterized in the underlying graph, which have nodes (subject, object) and edges (predicate) linking nodes. Nowadays, analysis of graph data has emerged as a hot topic because graphs are able to model the most complex data structures. The goal is to understand the structure and the behavior of networks. Extracting knowledge from an information network could answer questions such as the main topics of a set of publications, the central entities in a community and etc. Moreover, with such knowledge, it is possible to understand past events and to predict events in the future. In the example of bibliographic networks, nodes can be authors, publications, institutions or conferences, etc. Links can be ≪is written by≫, co-authors relationship, ≪belongs to≫, etc. Graphs may include labels or weights. Apart from the topological structure encoded in the underlying graph, multiple attributes are often specified and associated with vertices, forming the so-called multidimensional networks [ZLXH11]. With the multiple attributes, a network can be seen in different ways. A multidimensional network is defined as a graph where nodes are associated to attributes and edges just stand for a simple relationship. In the co-authorships network, each node represents an author and the associated attributes can be the gender, the age, etc. In reality, there is a semantic information between nodes. Thus the description

---

[1]http://dblp.uni-trier.de/
[2]http://dl.acm.org/

of relationships is not simple; there can be described attributes [WFW$^+$14, ZHPL12]. For example, these co-authorships have multiple attributes such as order of author for a paper and institutions.

Basically, OLAP was introduced to analyze structured data in order to perform aggregation oriented analysis from multiple dimensions of interest. OLAP should be able to handle information networks and be also useful for monitoring, browsing and analyzing the content and the structure of bibliographic networks. Extracting knowledge inside large networks is a complex task and too big to be comprehensive. Thus, OLAP analysis could be a good idea to look at a more compressed view.

In literature, there are research works and proposals that try to use OLAP technologies for information networks. OLAP on information networks is called Graph OLAP. The concept of Graph OLAP was first proposed by J. Han's team [CYZ$^+$08, QZY$^+$11, ZLXH11]. They provided a cube of graphs where each cell stores a network instead of a numeric value. Two kinds of OLAP dimensions were defined (informational and topological dimensions) with two kinds of OLAP operations to navigate on the dimensions. In other work [WFW$^+$14], the aggregated graph is a multigraph, where several edges can be between two nodes. It allows users to see the different views.

The existing Graph OLAP techniques know several limitations while the decision makers try to analyze and study some complex data in real-world situations. The first one is about the slowly changing dimension problem [KR11, WER15]. This problem happens when an object (a fact, a node, etc.) changes its content over time and when this causes a change in the structure. For example, the author, Y. Sun, published a paper when he was at Northeastern University, then, he published another paper when he was at university of Illinois. To the best of our knowledge, the existing approaches in Graph OLAP are not complete with this problem. But from the authors network, if the user does an OLAP operation like a Roll-Up in order to see the institutions network, these two papers will be counted for both universities, and it is an incorrect answer. In this case, networked data is non-summarizable: a higher level network cannot be computed solely from the lower level network without accessing raw data. The other limitation is about the visualization of a multidimensional and multi-level view over graphs. For example, a cube, with a venue dimension and time dimension, can contain a cell for (ICDE, 2008) and another one for (DOLAP, 2008). In the first Graph OLAP approaches, in each cell there is a graph showing collaborations between authors for this venue and this year. Between two authors, we can see the collaborations only according to the venue and the year, we do not see a global view of all collaborations.

Therefore, our aim is to solve these two problems in Graph OLAP analysis. We combine information networks and OLAP in order to present a new approach called graphs enriched by cubes that allows greater multidimensional analysis possibilities. A user may gain insight within both network and cubes. In the next section we discuss the contributions of this thesis.

1.2 Contributions

An established and well-researched way of analyzing information networks is through the techniques of *social network analysis* which relies on network and graph theory to study connections and relationships among the network nodes, and it reflects on network growth and density along other parameters. In this thesis, we cosider a completely different way as we are interested in aggregating information networks by using OLAP analysis. Our aim in this thesis is to go deeply in the analysis of data generated in the information networks by proposing a new online analytical processing on graphs called **G**raphs *enriched by* **C**ubes *(GreC)*. The main idea of *GreC* is to provide a cube for each node and edge in the network considered. *GreC* permits the users to explore and study the network considered in a different and complementary way on traditional Graph OLAP approach. Furthermore, *GreC* keeps a history of a network through the data presented within the cubes. Therefore, users can extract the evolution of the network considered by considering the time dimension. It also allows the user to quickly analyze the information summarized into cubes in order to analyze the network considered from different perspectives and with different granularities. Therefore, the summary of contributions proposed in this thesis in terms of extending the OLAP technology on information networks is as follows:

- In a different and complementary way of "classical" Graph OLAP, our proposal consists in enriching graphs with cubes instead of proposing cubes of graphs. Indeed, the nodes or/and edges of the network considered are described by a cube. Two types of measures are introduced to graphs enriched by cubes. First, they are graphs enriched by cubes with classical measures. More, we propose to add centrality measures (degree, betweenness and closeness) in order to explore the role of nodes in each networks. Centrality is important because it indicates which node occupies critical positions in the network. Our approach allows interesting analyzes for the user who can navigate within a graph enriched by cubes according to different granularity levels. It supports Graph OLAP operations such as informational and topological operations and it solves the slowly changing dimension problem. The changing information over time is an inherent feature of real

data. To the best of our knowledge, the existing approaches in Graph OLAP are not complete with this problem as we said before. Our work enables to solve this problem, authors can change their institutions.

- The properties of graph are able to model various information networks by adding a set of attributes to each node or edge. By analyzing the properties of a graph of an information network, we may acquire more information and take better decisions. Therefore, we used the properties of graph to design a graph model for bibliographic networks. Its content comes from multiple bibliographic databases in a way that allows us to build several different networks such as co-authorships, institutions of author, etc. This model is mapped easily to support a variety of use cases.

- Our approach is graphs enriched by cubes, it is not the classical data warehouse. Therefore we do not built a multidimensional conceptual model which are designed in classical data warehouses. In order to explain clearly our approach, we propose the definitions and notations that allows us to present the principle and algorithms. Our definitions and notations are presented by extending the concept of OLAP and Graph OLAP.

- To achieve graphs enriched by cubes, we propose two types of algorithms. The first type is to build the graph for analysis. The second one deals with computing the cubes. Four different algorithms are proposed according to the type of the measure considered: cube computation with numerical measures and cube computation with three centrality measures.

- Since, the nodes or/and edges of the network considered are described by a cube, it allows interesting analyzes for a user who can navigate within a graph enriched by cubes according to different granularity levels, with dedicated operators. As we said before, the semantics of graph OLAP operations are categorized into two major subcases: informational OLAP and topological OLAP. These operations are necessary to demonstrate the network considered. We adapted and extended these navigation operations to provide different analysis possibilities to the users. These operations are rather into account the slowly changing dimension problem. We consider two types of operations. The first one is to navigate in the cubes. In this case the structure of a network does not change, this type refers to informational operation in "classical" Graph OLAP. The second one deals with a network. In this case, operations can take into account the structure of network. It goes from one view of this network to another one. This type refers to topological operations.

- We provided a prototype tool that we developed to propose a proof of concept (POC) of our approach. It allows the user to quickly analyze information that has been summarized into cubes and by viewing the graph.

Our approach is generic and could be used in different domains. However, in this thesis, we chose to apply it on bibliographic data for scitometrics purposes.

The related works and our contributions published under various forms of papers: an international workshop [JFL13], national and international conferences [LFJ13, JFL15] and international journals [LJMF15, JFL16].

With this in mind, the next section describes the structure of this thesis.

1.3 Dissertation Organization

The remainder of this thesis is structured as follows:

In Chapter 2, a background of what is relevant to this thesis is presented. This chapter starts with an overview of bibliographic data, which is the running example of this thesis. Then it gives the definition of information networks, highlights the types of networks and shows the evolution of networks. An introduction to data warehouse concepts is then presented. Details on multidimensional modeling, OLAP and operators are given with relevant examples on bibliographic data.

In Chapter 3, a state of the art of OLAP on information networks is presented. General definitions of Graph OLAP are first given. Then a comparison between traditional OLAP and Graph OLAP is addressed. A literature review allows us to compare the different approaches, to address the limitations and to motivate our works.

In Chapter 4, we propose the graphs enriched by cubes approach. First we describe the process which is a user-centric process. Then, we introduce a graph model for bibliographic data. After definitions and notations, we describe the algorithms of computing a graph enriched by cubes. We describe the extension of OLAP operations to *GreC*. These take into account the structure of the network in order to do topological OLAP operations and not only classical or informational OLAP operations. A comparison between the basic graph OLAP and *GreC* is discussed at the end of the chapter.

In Chapter 5, we presente the prototype based on the framework proposed. In addition, we give an example of analysis by using real academic publications. We aim

at experimentally validate our algorithms by comparing them to a state-of-the-art approach close to our approach. Furthermore, we study the performance of *GreC* and the basic Graph OLAP according to different queries.

In Chapter 6, finally, we provide a summary of this thesis. In addition, we discuss the conclusions that can be drawn from the results in the evaluation of the process. Furthermore, we discuss future extensions to this work.

Chapter 2

Background

2.1 Introduction

This chapter provides the background that inspired the work in this thesis. To support our approach, Section 2.2 gives an overview of bibliographic data that we used as a running example in this thesis. This section also illustrates the realistic problems and the examples of research goals in the bibliographic data analysis. It also reviews some existing works in different research fields. Next, in Section 2.3, information networks which are used in our research are also introduced. This section gives a definition of information networks, then we present the different types of networks and the evolution of networks. The relevant concepts and terminologies of data warehouses and OLAP (Online Analytical Processing) are explained in Section 2.4 and Section 2.5. They review the design and implementation architecture options for data warehousing and online analytical processing. The conclusion of this chapter is presented in Section 2.6

2.2 Running example: the case of bibliographic data

Scientometrics and bibliometrics have become a standard tool of science policy and research management [Van97]. Bibliographic data relies on information designed and stored in bibliographic databases. Bibliographic data can be extracted from databases such as DBLP Bibliography, ACM Digital library and etc. Bibliographic databases contain the published literature from conference proceedings, journals, books and store a collection of fundamental information such as title, authors, year, venue, references and citations of the publication. Users can have a quick access, online, to them thanks to digital libraries. Figure 2.1 p.10 shows an example of bibliographic data.

Bin Yang (Aalborg University and Fudan University), Hua Lu (Aalborg University) and Christian S. Jensen (Aalborg University), 'Probabilistic threshold k nearest neighbor queries over moving objects in symbolic indoor space', EDBT, 2010.

Jiawei Han (University of Illinois), Xifeng Yan (University of California) and Philip S. Yu (University of Illinois), 'Scalable OLAP and mining of information networks', EDBT, 2009.

Tianyi Wu (University of Illinois), Yizhou Sun (University of Illinois), Cuiping Li (Remin University) and Jiawei Han (University of Illinois), 'Region-based online promotion analysis', EDBT, 2010.

Yizhou Sun (University of Illinois), Yintao Yu (University of Illinois) and Jiawei Han (University of Illinois) 'Ranking-based clustering of heterogeneous information networks with star network schema', KDD, 2009.

Yizhou Sun (University of Illinois), Jiawei Han (University of Illinois), Xifeng Yan (University of California) and Philip S. Yu (University of Illinois), 'Integrating meta-path selection with user-guided object clustering in heterogeneous information networks', KDD, 2012.

Peixiang Zhao (University of Illinois), Jiawei Han (University of Illinois) and Yizhou Sun (University of Illinois), 'P-Rank: a comprehensive structural similarity measure over information networks', CIKM, 2009

Yizhou Sun (University of Illinois) and Jiawei Han (University of Illinois), 'RankClus: integrating clustering with ranking for heterogeneous information network analysis', EDBT, 2009

Bin Yang (Aalborg University and Fudan University) and Christian S. Jensen (Aalborg University), 'iPark: identifying parking spaces from trajectories', EDBT, 2013

…

Hongzhi Yin (Peking University) and Yizhou Sun (Northeastern University), 'LCARS: a location-content-aware recommender system', KDD, 2013.

FIGURE 2.1: Example of bibliographic data

Many research fields are interested in bibliographic data analysis because they contain very rich and useful information. This is not an easy task: due to the quantity and the variety of approaches that are focusing in this subject, with various goals, it is not possible to provide a comprehensive summary of all these approaches. In this section, we introduce some examples of research goals in the bibliographic data analysis and we review some existing works in different research fields. In the analysis of bibliographic data, different objectives can be interesting :

1. **Search engine** [QZY$^+$11, HV03, BBH$^+$08, KLR$^+$04, KRW$^+$06, ZCG09, ML10].
   By keyword(s) search, these tools are made to help users for searching information to prepare reports and documentation requiring the citation of relevant papers (according to authors, conferences and so on).

2. **Relationship studying** [BBH$^+$08, KLR$^+$04, KRW$^+$06, ZCG09, PK10, VT11, HYQQ09, Cab11].
   The structure of bibliographic data is also interesting for studying the relationships among entities. Each publication is composed of authors, venue and related data. Researchers have analyzed the patterns of collaborations in co-authorship, the centrality, the structured links between universities and the relationship in career of the authors (professor and student), etc.

3. **Ranking**. [BBH$^+$08, DKL08, SQU10]
   Ranking analysis can be used for research evaluation. It evaluates objects based

on mathematical functions and it compares objects of a same type. A lot of approaches have been proposed to rank journals (for example with impact factors), conferences, and authors. For example, an impact factor is a method for ranking journals.

4. **Community mining**. [ZCG09, ML10, VT11, CGP09, HYQQ09]
   The goal is to find groups of objects that share similar properties and that are connected to each other. Identifying these connections and locating objects in different communities are considered to be valuable to find potential collaborators for researchers, to discover communities in an author-conference social network, and also to find reviewers to be invited as program committee members, etc.

5. **Topic detection**. [ZCG09, DKL08]
   Topic detection can identify topics by exploring and organizing the content of textual data and aggregating information into clusters automatically. In the context of publications, topic detection can cluster publications according to their content, can find the main topics of a group of conferences, can detect the most relevant trends in a research field and so on.

6. **Multidimensional exploration**. [GT11, HV03, BBH$^+$08]
   Bibliographic databases are huge with a lot of data. However, users need only consistent and valuable information such as portion of objects, links or sub-networks. But bibliographic data features cannot be taken into account separately. So bibliographic data analysis can support multidimensional exploration and reporting. For instance, it could be useful to follow up the evolution of the discovered topics for a keyword over time.

7. **Prediction**. [HYQQ09]
   Many applications of bibliographic network analysis are focusing on predicting links or interactions among objects. A supervised model is used to learn the knowledge history. Then, it can predict new information such as research trends over time or in groups, the emergence of a new topic/conference in the future.

To achieve these goals, various methods can be used, they come from different fields such as:

- **Statistics**. The application of mathematics and statistical methods to analyze bibliographic data is not new. It started in the twenties and became more popular in the sixties [Hul23, Pri69]. At present it is widespread and used by the scientific community, thus its interest does not need to be more discussed.

- **Graph theory**. Graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A graph contains vertices or nodes representing objects and edges or links which are relationships between nodes [New03, Die00]. For example graphs can be used to represent a network of publications where nodes are authors and edges are the relationships between two authors written papers together.

- **Data Mining**. Data mining [FPSS^{+}96] is a process to discover hidden information (called knowledge) and meaningful structure from very large databases. It uses both supervised and unsupervised learning algorithms to cluster, classify, explain and predict data. It can help to discover, describe and predict links or trends within data.

- **OLAP analysis**. OLAP (Online Analytical Processing) [CD97] is the technology to exploit information in data warehouses. OLAP allows a multidimensional data analysis by building cubes; it provides easy navigation, visualization and fast analysis for decision making within a vast amount of data.

Among these different types of analysis, OLAP can provide the flexibility for navigating into networks, for summarizing networks at different granularity levels and from different points of view. The ability of OLAP offers users to access networks in multidimensional ways. OLAP could be a good tool in order to have a more compact view of networked data.

2.3 Information Networks

2.3.1 Definitions

An information network is made of a large number of interacting and multi-typed objects. Graphs have been widely used for modeling networks. Graphs are often used to visualize relationships between data, relationships which are not apparent when searching and browsing data.

A graph $G = (V, E)$ consists of V, a set of vertices or nodes and E, a set of edges or links. Each edge has two vertices associated with it. A node can be connected by one or more links. Each node represents an object or an entity, an edge or a link is a relationship between two nodes. In the example of bibliographic networks, entities can be authors, publications, institutions or conferences, etc. Links can be ≪is written by ≫, co-author

FIGURE 2.2: Example of authors network: a homogeneous network

relationship, ≪belongs to≫, etc. These links may include labels or weights. Apart from the topological structure encoded in the underlying graph, mutiple attributes are often specified and associated with vertices, forming the so-called multidimensional networks [ZLXH11]. A multidimensional network is defined as a graph G = (V, E, A), where A is a set of n vertex-specific attributes. A is called the dimensions of the network. In the co-authorships network, each node represents an author and the associated attributes can be the gender, the age, etc (Figure 2.2). In reality, there is a semantic information between nodes. Thus the description of relationships are not simple, there can be described attributes [WFW+14, ZHPL12]. For example, these co-authorships have two attributes such as order of authors and institutions.

Within bibliographic data, graphs are currently provided to show relationships between conferences and journals or authors. Klink *et al.* proposed `DBLBrowser`, a user friendly interface, for searching, browsing, and mining bibliographic data [KLR+04, KRW+06]. Their system combined both textual and visual browsing functionalities. It could find the related publications and their correct bibliographic data. During the browsing process, data are visualized by appropriate graphical techniques that help users to understand their research domain, helping them finding relevant authors or publications and above

FIGURE 2.3: Example of author-paper network: an heterogeneous network

all providing information about further researchers or important conferences or journals. Zaiane *et al.* introduced DBconnect, a prototype that exploits the social network analysis in the DBLP database [ZCG09]. They drew on a new random walk approach to reveal interesting knowledge about the research community and even to recommend collaborations. The system looked for finding research communities, relevant conferences, similar authors, interesting topics, etc. It combined a random walk algorithm, text mining techniques and social network analysis to compute relevance scores between data to extract knowledge. Muhlenbach and Lallich proposed a matrix formalization to consider the similarity and dissimilarity between social relationships [ML10]. They tried to discover research communities with a clustering method using the neighborhood graph obtained with the dissimilarity scoring. A graph-theoretic model for discovering research communities with DBLP database is also introduced. Pham and Klamma clustered research communities of similar venues [PK10]. They were interested in the structure of the networks of Computer Science journals, conferences and workshops using citations analysis. Social network analysis (SNA) was applied to determine clusters of venues by calculating two network analysis measures for each venue: betweenness and PageRank. Varlamis and Tsatsaronis proposed a new model for bibliographic data to identity the future research from a co-authorship network [VT11]. The new representation model combines co-authorship and content similarity information. Authors used

a graph visualization tool from the biological domain to provide comprehensive visualizations that help users uncover hidden relations between authors and suggest potential synergies between researchers or groups. Gupta *et al.* considered the two problems of clustering and evolution diagnosis of bibliographic networks [GAHS11]. They presented an algorithm, `ENetClus`, which performs such an agglomerative evolutionary clustering which is able to show variations in the clusters over time with a temporal smoothness approach. They used a probabilistic generative model from each cluster. They evaluated an object in clusters by a maximum likelihood approach, including ranking condition of object in current and previous clusters.

All these proposals show us the interest of dealing with information networks, particularity in the context of bibliographic data. Now let us focus more on this context of network since different types could be envisaged.

2.3.2 Types of network

There are two types of networks. In the first type, networks are homogeneous. In the other type, networks are heterogeneous.

Homogeneous network. Homogeneous networks contain a single object type and a single link type such as co-authorships network. The co-authorships network (or the authors network) is a homogeneous network: each node represents an author; each edge between two authors represents a co-author relationship, in one or several papers, with attributes like conference, year and venue (Figure 2.2 p.13). There may be multiple edges between two nodes if two authors have co-written more than one paper together. For instance, authors Jiawei Han and Xifeng Yan wrote together one paper in 2009 at EDBT conference and one in KDD 2012. So, the weight 2 has been added on the edge between them.

Heterogeneous network. Heterogeneous networks are composed of multiple objects and link types. An example is given by the author-paper network (Figure 2.3 p.14). This network has two types of nodes: authors and papers. There are three types of edges. The first link is ≪written by≫ between authors and papers. The second represents co-author relationships and the last one relates papers written by the same authors. Each object is associated with a set of multidimensional attributes describing this object. For instance, paper object has venue and time attributes. But it is also associated to a title and keywords.

(a) 2009

(b) 2010

(c) 2012

(d) 2013

FIGURE 2.4: Co-authorships network

2.3.3 Evolution of networks

Let us continue by introducing the elements under discussions. The networks that we consider are graphs consisting of nodes connected by edges. Both nodes and edges could have some attributes. Most of the real networks extracted from various data sources evolve and change over time. There are several ways in which a network can evolve. We sum up as the following.

1. **Node evolution**

   In bibliographic networks, node evolution happens when a new node is added to the network or a node is removed from a network. For example, in co-authorships network it happens when a new author produces a paper for a year. In the same way, some authors may disappear. Figure 2.4 p.16 shows co-authorships network for each year from 2009 to 2013. A group of authors containing Bin Yang, Hua Lu and Christian S. Jensen does not publish any papers in 2009 (see Figure 2.4a p.16). In 2010 (see Figure 2.4b p.16), these authors appear in co-authorships network because they publish a paper. On the contrary, Yintao Yu, Philip S. Yu and Xifeng Yan are removed from co-authorships network in 2010. However, they publish a paper again in 2012 (see Figure 2.4c p.16).

2. **Edge evolution**

   The edge evolution consists of edge addition and edge deletion. There is a basic architecture that is essential to support the life of a network but the connections keep changing. The edge evolution could assume into two ways. First, a new edge appears in a network when a new node is added. For example, an edge between Tianyi Wu and Yizhou Sun is a new one in co-authorships network in 2010 (see Figure 2.4b p.16) comparing to co-authorships network in 2009 (see Figure 2.4a p.16). Likewise, an edge between Philip S. Yu and Jiawei Han is deleted from co-authorships network in 2010 (see Figure 2.4b p.16) compare with co-authorships network in 2009 (see Figure 2.4a p.16). Second, it is assumed that the second way appears when the number of nodes remains unchanged but the number of edges is modified. For example, there are three authors in co-authorships network in Figure 2.5a p.18. These authors published one paper together from 2009 to 2010. Then a paper written by Bin Yang and Christian S. Jensen, which is published in 2012. This means that there is a new edge. Due to the existing of this edge in a network, the number of edges does not change but there is a change of the number of publications only for the edge between Bin Ying and Hua Lu (Figure 2.5b p.18).

3. **Properties evolution**

   The properties could change over time. For example, Yzhou Sun published a paper
   in 2009 when he was at university of Illinois, whereas his other publications were
   published for Northeasten university (see Figure 2.1 p.10).

(a) 2009-2010

(b) 2009-2012

FIGURE 2.5: Example for edge evolution

Bibliographic networks are usually not a static structure and they may change over time.
Thereby, the set of nodes, edges and properties may vary over time. These dynamics
need to be represented and it could be modeled by a time dimension. To analyze the
evolution of the network, there may be two possible ways.

First, a set of static pictures (snapshots), as shown in Figure 2.4, is representing the
state of the network obtained in certain time intervals. Time window limits network
analysis to those nodes and edges that have existed in a period defined by the size of
the time window. The visualization of the network may be one-layered or multi-layered.

Missing information or change prediction is then possible by changing networks from successive time windows, e.g. time 1 and time 2 (Figure 2.4a and 2.4b). The second way is that there are some tools that extend the evolving network to the animation of network visualization. This is helpful to visualize the process of change rather than simply the final network.

2.4 Data warehouses

The term Data warehouse was first introcuced by W.H. Inmon in 1992 [Inm92]. In the following years, the data warehousing technology has known a tremendous growth and has been playing a key role in supporting decision making in a variety of application domains [CD97]. A data warehouse is specially prepared a data repository that is used to support decision making. A data warehouse is a "subject-oriented, integrated, time varying, non-volatile collection of data that is used primarily in organizational decision making." It also refers to as a data warehouse system architecture. Typically, the data warehouse is maintained separately from the organization's operational databases. There are many reasons for doing this. The data warehouse supports online analytical processing (OLAP). Data warehouses are targeted for decision support. Data warehouses contain consolidated data, from several operational databases, over potentially long periods of time. Therefore, historical, summarized and consolidated data is more important than detailed, individual records.

In the data warehouse literature, there are discussions and examples of various system architectures. However, a classical reference architecture is depicted in Figure 2.6 p.20 and comprises four stages. Each layer, namely, data source layer, ETL layer, data warehouse layer and analysis layer transforms raw data into actionable knowledge for decision makers.

Data sources layer. It represents a variety of data storage such as operational data stores (ODS), spreadsheets, reports, web documents, etc. These may come from the company's information systems or came from information system outside the company.

ETL process. It means extract, transform and load process. It encompasses processes required to extract data from multiple and mostly heterogeneous sources, transform them according to the target schema and then upload them into the data warehouse. The data stored within sources is extracted, cleaned to remove inconsistencies and fill gaps, and integrated to merge several sources into one schema. This process takes place

FIGURE 2.6: A typical data warehouse architecture

when the data warehouse needs to be updated with the new data. The data warehouse update can be event-driven, periodic or depending on a threshold of data volume. The ETL layer remains transparent to the end-user and applications.

Data warehouse layer. Preprocessed and transformed data is stored. The data warehouse can be directly accessed, but it can be also used as a source for creating data marts, which partially replicate data warehouse content and are designed according to analysis needs. These data must serve to support the information requirements of a business function or department. A meta data repository stores information on the sources, access procedures, data mart schemas, and so on.

Analysis layer. It converts data into actionable knowledge. This layer exhibits data analysis methods, techniques, and tools to process and analyze the underlying data in data marts and the data warehouse. It should include features of aggregate data navigators, complex query optimizers, and user interface.

To facilitate complex analyzes and visualization, the data in a warehouse is typically modeled multidimensionally. OLAP might be the main way to exploit information in a

data warehouse. It is the most popular one and it gives the opportunity to analyze and explore data interactively on the basis of the multidimensional model. It also enables users to access information from multidimensional data warehouses almost instantly, to cleanly specify and carry out sophisticated calculations and to view information in any way they like [Tho02].

2.5 OLAP (Online Analytical Processing)

2.5.1 Concepts of OLAP

The OLAP Council provides a definition of OLAP as a category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user [Def96]. The underlying technology and data models are expected to support the objectives stated in this definition. We first present definitions of the core elements of OLAP followed by a discussion on valid data modeling schemes.

Facts and Measures. Facts are recordable and usually measurable business events that form the subject of analysis. Facts are recorded at different levels of detail (granularity) depending on the subject. The finest grain of facts is stored in a fact table, a primary table in the multidimensional model. The scope of the measurement and the grain of the facts are defined by a set of dimensions [KR02]. Useful facts are usually measurable and hence are numeric, additive, continuously valued. Measures can undergo arithmetic operations such as plus, minus, multiply, divide and can also be aggregated using sum, average, etc., into a single logical measure only if the measures under consideration belong to the same type. In Figure 2.7 p.22, the facts are the publications and the measure is the number of papers.

Dimensions. According to Ralph Kimball and Margy Ross [KR11], dimension tables are integral companions to a fact table. The dimension tables contain the textual descriptors of the domain interested. In a well-designed dimensional model, dimension tables have many columns or attributes. Dimension attributes serve as the primary source of query constraints, groupings, and report labels. In a query or report request, attributes are identified by words. Dimension table attributes play a vital role in the data warehouse. Since they are the source of virtually all interesting constraints and report

FIGURE 2.7: A sample data cube with three dimensions

FIGURE 2.8: An entire view of a sample data cube

labels, they are play a key role to making the data warehouse usable and understand-able. In many ways, the data warehouse is only as good as the dimension attributes. The power of the data warehouse is directly proportional to the quality and depth of the dimensions. Dimensions allow analysts to look at the facts from various perspectives and aggregate them along logical and meaningful path(s) called dimensional hierarchies, or simply hierarchies. Hierarchies establish classically strict many-to-one relationships where facts roll up into higher levels of summarization [KR02]. An example of a hierar-chy in the venue dimension is given in Figure 2.7 p.22. The venue dimension hierarchy includes three levels: the support (the name of the conference, of the journal, or of the book), the research area (like databases, data mining, information retrieval, etc.) and the all level.

Let us note that this concept of multidimensional modeling with facts and dimensions can be emerged in the context of modeling the data warehouse itself. From the multidi-mensional modeling we can emerge the concept of multidimensional cube.

2.5.2 Multidimensional data cube

The multidimensional model is used to represent the fact to be analyzed and the analysis axis. The fact, which is a subject of analysis, is analyzed through one or more dimen-sions that constitute the analysis axes. Each fact measure is stored at the corresponding intersection of cooperating dimensions in a cell and is aggregated along dimensional hi-erarchies for analysis. Dimensions correspond to the aspects of analysis. There is no limit on the number of dimensions in a cube. It can be 2-dimensional, 3-dimensional and higher-dimensional even if classically we represent a 3-dimensional cube for a men-tal picture. Queries are performed on the cube to retrieve decision support information. For example, we have a database that contains information relating to the publications of scientific authors at a conference. The data cube could be a three-dimensional rep-resentation, with each cell of the cube representing a combination of values for author, venue and time. From an example of bibliographic data in Figure 2.1 p.10, a sample data cube for this combination is shown in Figure 2.7 p.22 and the detail of each cell is shown in Figure 2.8 p.22. The contents of each cell is counted from the number of times that specific combination of the values occurs together in the database. Cells that appear blank in the fact is a value of zero in this figure (Note that classically it could compare to missing data). The cube can then be used to retrieve information within the database about, for example, who are the leaders in the conference in order to emerge interesting the collaborations.

In the implementation of data cubes, the optimization of the OLAP performance with respect to materialization of cubes can be done using the following possible solutions:

1. **Full cube materialization** is that the entire cube is pre-computed (including all cuboids). In the relational context, all aggregates are stored in separate, called materialized views. Then queries run on the cube will be the fastest query response. The disadvantage is that it requires heavy precomputation and a lot of storage.

2. **None cube materialization** is to minimize storage requirements. It can pre-compute none of the cells in the cube. This gives the slowest query response time and always requires query evaluation. However, it needs smaller of among storage space. The disadvantage here is that queries on the cube will run more slowly because the cube will need to be rebuilt for each query.

3. **Partial cube materialization** is to select and materialize some parts of data cube. This implements a balance between the storage space and the response time, which will most likely be used for decision support queries.

Aggregated data is calculated on the basis of the hierarchical relationships defined in the dimension. Queries can be written, when a query requires to aggregate data. If the query cannot redirect to get a result set from an existing cubes, data is aggregated to answer this query on the fly. Basically, a cube is built from a data warehouse. However, we think that a cube can be created without a data warehouse. Here are the ways to prepare a cube:

1. A data warehouse is built. There is no any data cube and aggregation tables are not pre-defined and are not pre-summarized structures.

2. A data warehouse is built. Aggregation tables are pre-defined and are presummarized structures. However, some aggregations are not pre-defined. If the aggregated data needed for the result stored in the space, then it is simply retrieved. If the aggregated data does not exist, then it is calculated on the fly. For example, the data cube in Figure 2.7 p.22 is built in the pre-processing step. Suppose that a query needs to know how many publications for each author in each conference in all years. No cubes can answer this query, a new cube has to be computed.

3. The data cubes will be built from the data according to user's requirement. Data cubes are flexible. This can be used to compute data because data is complex relationships. This is also used to support real time and effective decision-making. For example, Mehdi *et al.* generated data cubes on the fly from syntactic sensor data to sustain decision making without using a data warehouse [MMC13].

FIGURE 2.9: A cube with Roll up operation on time dimension

FIGURE 2.10: A cube with Drill down operation on time dimension

2.5.3 Operations

Data, or facts, stored in a multidimensional cube can be accessed and manipulated by operators in many ways to support efficient navigation, analysis and achieve insights. There are five classic OLAP operations as follows.

- **Roll-up** takes the current data and does a group-by on one dimension in order to aggregate or summarize facts to higher granularity. Considering the Figure 2.9 where the cube has the number of papers as a measure and authors, venues and time as dimensions, a roll-up can aggregate, for instance (roll up according to the time dimension), the number of papers for each author on a venue for all years.

- **Drill-down** is the dual of the roll-up operator by giving more details and navigates from aggregated data to a lower level of details (see Figure 2.10). It performs the opposite roll-up operation.

- **Slice** is another way to explore the cube. It reduces the cube's dimensionality by projecting the data onto a subset of dimensions while setting other dimensions to selected values. It reduces dimensions for taking a sub-cube. Figure 2.11 p.26

FIGURE 2.11: A cube with Slice operation

FIGURE 2.12: A cube with Dice operation

shows the process and the results of slice in the cube where the time dimension is sliced down to a single value 2010.

- **Dice** reduces the size of slice by filtering its data along any dimension(s). Figure 2.12 shows the process and the results of dice in the cube where it is further diced by selecting value EDBT from the venue dimension and value 2010 from time dimension.

- **Pivot** is also known as rotation, which implies a change in layouts. It aims at analysing an individual group of information from different viewpoint. If you pivot data, you rotate the data axes in view in order to provide an alternative presentation of data of a new perspective. Consider the Figure 2.13 p.27 that shows the pivot operation on author dimension and venue dimension.

Dimensionality modeling generated from the fact data through other computations can be considered as a special case of slowly changing dimensions, in which the changes occur with a certain regularity. The state of the dynamic category is guaranteed to be fully up-to-date, if it is computed from the recent state of the underlying set of facts. It have to recompute the assignment each time new facts get inserted into a cube. We

FIGURE 2.13: A cube with Pivot operation

investigate proposals in bibliographic data on storing, maintaining and querying such dynamic dimensions. Therefore, in the following section, we give the overview of the slowly changing dimensions problem.

2.5.4 Slowly changing dimensions

One of the property of data sources is that their content is changing over time. The maintenance of the history of changes allows users to inquire about the state of the real world data at a given time. The standard approach has been proposed to handle the evolution of data, which is slowly changing dimension. The expression of "Slowly changing dimension (SCD)" was invented by Kimball and Ross, who are regarded as one of the original architects of data warehousing [KR11]. They argued that dimension attributes are not static and slowly changing in time. In reality, bibliographic data may have two problems. First, an entity concerns many different values in the same property. For example, author named Bin Yang works at Aalborg university and Fudan university in the same time (see Figure 2.1). Secondly, a property value is changing over time such as a change of institution. Look at Figure 2.1, Yzhou Sun published a paper in 2009 when he was at university of Illinois, whereas his other publications were published for Northeasten university. In our example bibliographic data, we can have several evolutions of data:

- A venue may be stopped because it cannot be organized. A discontinued venue may be reintroduced at a later time.

- Topics change due to author's interests or technological evolution.

- An author changes his or her institutions for another one to the same country or a different one.

- An author works at more than one institution in the same time. Or an author is interested in more than one research topics.

The approach to slowly changing dimensions offers seven different techniques referred to SCD Types to track the changes in attribute values. Notes that we speak about this term because when we aggregate the data considering one dimension, it can introduce problems in computing the result. For instance, aggregating the number of papers by institutions when the authors' institutions is changing over time. Next, we briefly describe Kimball's three basic responses to the problem of SCDs [KR11, WER15].

Type 1. The changes are handled by overwriting old values with new ones. With this option, no history is maintained. Consequently, there will be no possibility to analyze the evolution of those characteristics or to perform historically correct aggregation.

Type 2. It aims at correct preservation of the prior history by creating a new record to reflect each change. A single instance of a dimension is stored allowing multiple rows (one for each change) to refer to the same instance. A common extension of Type 2 storage is to add extra columns for storing the start and the end timestamp for each version. Even though this solution provides an accurate change tracking and ensures historically correct aggregation, it has a huge disadvantage of having multiple records for each instance and computing analysis is more complex and should adapted to the context of multi-version.

Type 3. A separate column is used to enable change tracking for each version of the changed attribute. A separate attribute is added to capture the history for each change. When an attribute's value is changed, its existing value is written in the separate attribute.

2.6 Conclusion

In this chapter, the necessary background was given. Bibliographic data was introduced. We saw that we could generate several networks such as authors network and institutions network, and their content is changing over time. Therefore, due to their characteristics and complexity, bibliographic data provides a good running example to illustrate after our contribution. A special feature of bibliographic data is that it can be seen as information networks. The other part of this chapter explored the world of information networks that has attracted the interests of many researches. Types of network are classified into

two types. First, homogeneous networks contain a single object type and a single link type. Second, heterogeneous networks are composed of multiple object and link types. A discussion on the evolution of information networks, their data that can change over time (i.e., a new nodes is added to the network or a node is removed from a network, etc.) and the ways to analyze the evolution (i.e., a set of static pictures or tools that extend the evolving network to the animation of network visualization.) are also presented.

With the introduction of bibliographic data, we discussed this data according to the types of analysis. We determined the types of analysis by five criteria: statistics, data mining, graph theory and OLAP to achieve different objectives in bibliographics (relationship study, ranking, community mining, etc.). Among these different types of analysis, OLAP can provide the flexibility for navigating into networks, for summarizing networks at different granularity levels and from different points of view. Therefore, in the next section, we introduced the relevant terms, concepts required to establish OLAP analysis, which is an interest technology in data warehouses.

Traditional OLAP did a great job in collecting data providing answers on classical data. OLAP technologies support multidimensional analysis, however, they cannot recognize patterns among process graphs and analyzing multidimensional graph data. Therefore, OLAP faces challenges in processing networked data. Usually, dynamic graphs are the different screenshots with time windows. In the context of OLAP on information networks, it allows to analyze the evolution of networks with time dimension. In the next chapter, we will give the definition of OLAP analysis on information networks and we will review the existing approaches.

Chapter 3

OLAP on information networks: a state of the art

3.1 Introduction

Traditionally, data warehouses and OLAP are used to store, to model, to analyze and to visualize relational structured or semi-structured data and more recently textual data and XML data. Data warehouses traditionally present information in tables with rows and columns. A table is a collection of objects (records or rows) of a same type. Relationships occur between tables but records are not considered as interconnected and interrelated objects across multiple types of relations.

In many cases, data of interest can be described as heterogeneous information networks. In real applications, networks contain several and complex types of relationships. It is difficult to explore information in-depth with many relationships. The ability of OLAP for analyzing classical data is clear. However, the insights of OLAP remain hidden in the interactions among objects. We believe that OLAP analysis helps users to access data from different points of views in order to explore knowledge from that networks in a multidimensinal way. Therefore, OLAP must change if we want to make online analysis of data from information networks which are modeled as graphs. In literature, there are several expressions for speaking about OLAP on information networks. One of them is Graph OLAP and it is a generalization of Social OLAP which is OLAP on data from social networks.

This chapter emphasizes on the study of OLAP analysis on information networks that is thus called Graph OLAP. The strengths and weaknesses of current development practices are also explored and discussed in order to clearly position our research contribution. Consequently, we start with illustrating the relevant background on Graph OLAP.

Section 3.2 explains the general definitions of the extensions of OLAP technology on information networks i.e., the definitions of dimensions and measures, the semantic of OLAP operations in Graph OLAP. Then, in Section 3.3, we compare the differences between OLAP and Graph OLAP. To position our research contribution, Section 3.4 and Section 3.5 express literature reviews of approaches of OLAP analysis on information networks by discussing and comparing them according to different criteria. The conclusion of this chapter is presented in Section 3.6.

3.2 General definitions

The concept of Graph OLAP was first proposed by Chen *et al.* [CYZ$^+$08] in with general framework for OLAP on information networks. Graph OLAP is a collection of network snapshots where each snapshot i has k informational attributes describing the snapshot and has a graph $G_i = (V_i, E_i)$. Such snapshots represent different sets of the same objects in real applications. For instance, with regard to the author-paper network of the Figure 2.3b p.14, venue and time informational attributes can mark the status of each individual snapshot e.g. CIKM 2009 and EDBT 2010. An authorID is a node attribute defining each node, and collaboration frequency is an edge attribute reflecting the connection strength of each edge. For instance, Figure 3.1 is a cube of graphs or a cube of snapshots. Dimension and measure concepts, found in traditional OLAP domain, are adapted for Graph OLAP.

There are actually two types of dimensions in Graph OLAP.

Informational dimension. The first one is an informational dimension, and it is based on an informational attribute. This kind of dimension has two roles: organizing snapshots into groups based on different perspectives and granularity (each group corresponds to a cell in the OLAP cube) and controlling snapshot views. But they do not touch the inside of any individual snapshot. For example, the two informational attributes venue and time with their respectively hierarchical concepts semester, year, decade, all and support, research area, all can be used as informational dimensions. We can look at a network of authors by summing a set of graphs for the EDBT conference for all years (Figure 3.2 p.33).

Topological dimension. The second type of dimension is a topological dimension coming from the attributes of topological elements. Topological dimensions operate on nodes and edges within individual networks. Let us consider the organization dimension for instance, the organization dimension with the hierarchy institution, country, all can

FIGURE 3.1: A cube of graphs

FIGURE 3.2: Example of aggregated network for informational dimension

be used as topological dimension and allows to merge all authors from the same institution in a more general node. A new graph with more generalized nodes is generated by summarizing the original network (see Figure 3.3 p.34). Topological dimensions operate on nodes and edges within networks. We think that topological dimensions are a real added value in modeling because they allow to model the relationships between objects.

(a) An author-paper network

(b) An institution-paper network

FIGURE 3.3: Example of aggregated network for topological dimension

In the next section, we present the difference between OLAP and Graph OLAP.

3.3 Comparison between traditional OLAP and Graph OLAP

We propose a comparison between traditional OLAP and what is or what Graph OLAP should be. Our comparison is summarized in Table 3.1 p.36.

As we said, data warehouses support OLAP technology and they have been very useful for efficient analysis onto structured data or semi-structured data and more recently textual data and XML data. Data warehouses are used to store, to model, to analyze and to visualize all these kinds of data. Information of data warehouses is collected in a collection of objects with rows and columns. Relationships occur between objects but rows are not considered as interconnected and interrelated objects across multiple types of relations. However, in Graph OLAP, information are interconnected and are in the form of networks. In real applications, networks contain several and complex types of relationships. It is difficult to explore information in-depth with many relationships. We believe that heterogeneous information networks can be considered as a generalization of databases, semi-structured data and even a kind of corpus of documents. For example, from a database of publications such as DBLP and ACM, where publications are linked via authors, citations, institutions, topics, etc., we can build a network of co-authors, a network of citations, a network of conferences, etc. OLAP is an interesting tool if we want to make online analysis of data from information networks and modeled as graphs. In traditional OLAP, cubes contain facts defined by dimensions and measures. With the use of operators like the roll up, aggregates are obtained. Aggregates are facts whose measure was aggregated according to dimensions. In Graph OLAP, cubes can contain graphs as input. Graphs are defined by a structure (entities and edges) and attributes. The aggregation of a graph gives a more general graph as output. Graph cubes consider both attributes and structures for network aggregation. A given network as input is changed into a new network as output.

Two types of dimension have been presented in Graph OLAP (informational and topological dimension) whereas there is only one type in traditional OLAP.

In term of measures, traditional OLAP has numeric measures and aggregation functions such as COUNT and SUM to summarize multiple records. There are two types of measures in Graph OLAP. First, the measure can take the form of a graph and the aggregation function is then specific to graph. The second type of measure is not graph but can be indicators coming from graph theory such as average degree and diameter. In traditional OLAP there is only one semantic for operators such as roll-up. The OLAP semantics accomplished through informational dimensions and topological dimensions are different and Chen *et al.* speak about informational OLAP (abbr. I-OLAP) and topological OLAP (abbr. T-OLAP), respectively. With roll-up in informational OLAP, snapshots are just different observations of the same underlying network, and they are grouped into one cell in the cube, without changing the network structure. For a roll-up

in topological OLAP, networks are not grouped but the reorganization is inside individual snapshots and a new generalized graph is built with a new topological structure.

Lastly, a traditional data warehouse does not consider relationships between records.

TABLE 3.1: Comparison between traditional OLAP and Graph OLAP

	Traditional OLAP	Graph OLAP
Input	Facts in cuboids	A given network with snapshots
Output	Aggregated measures	A new network more generalized
Dimensions	Attributes	Informational and topological attributes
Hierarchies	Yes	Yes (both for info. and topo. dimensions)
Measures	Numeric indicators Aggregation function (count, sum, average, ...)	Aggregated graph measure Measures coming from graph theory Specific aggregation functions
Operations	Roll-up, drill-down, slice & dice, pivot	Operations within informational or topological OLAP
Problems	Not considering links among data records	How taking interactions among entities into account

3.4 Literature review

The topic of OLAP on information networks is quite new. Only few research teams have been interested in this topic. To the best of our knowledge, the first works were published around 2008.

J. Han's team and his colleagues were among the first to investigate OLAP on information networks [CYZ+08, QZY+11, JHC+10, ZLXH11]. Chen *et al.* presented the basic definitions of OLAP on information networks and introduced a general framework called `Graph OLAP` [CYZ+08]. Qu *et al.* worked on topological OLAP operations to allow *roll-up* operations on topological dimensions by changing the topological structure of the aggregated graph [QZY+11]. The key problem is to efficiently compute measures for the newly aggregated networks and handle user queries with various constraints. Two effective computational techniques, `T-Distributiveness` and `T-Monotonicity` are proposed to achieve efficient query processing and cube materialization. Zhao *et al.* defined the concept of multidimensional networks to abstract the real networks and they introduced a new multidimensional model, called `Graph Cube`, to extend data warehouses to

large multidimensional networks [ZLXH11]. They worked with structure-enriched aggregate networks and they proposed a new type of query for multidimensional networks, called `crossboid query` in contrast with traditional queries named `cuboid query`: a `crossboid query` can cross more than one cubes in query, rather than a `cuboid query` is on a single cube. `Graph Cube` model also considers aggregation networks both on entities and relationships. Jin *et al.* proposed the `Visual Cube` model and OLAP analysis for image collections, such as Web images indexed in search engines, product images or photos shared on social networks [JHC+10]. `Visual Cube` provided online answers to user requests with summarized statistics of image information and helped users navigate and explore images efficiently. Four measures have been presented in the `Visual Cube` model. The first measure is to summarize information as in traditional OLAP. The other measures are unique for `Visual Cube`: summarized image feature (i.e. average color histogram), subset of images (i.e. clustering images and choosing the central one) and all images (ranking lists).

With regard to summarizing attributed networks in the context of OLAP analysis, the closest works to those of Han's team are those of Tian *et al.* They introduced two operations to summarize graphs in OLAP analysis [THP08]. The first operation, called `SNAP` (Summarization by Grouping Nodes on Attributes and Pairwise Relationships), merges homogeneous nodes, combines corresponding edges and aggregates a graph that displays relationships for generalized nodes. The second one, called `k-SNAP`, allows users to control the size of summarized graphs by specifying the number of k groups. Similarly, there are some works, which presented the conceptual graph cube model to aggregate attributed networks [ZHPL12, WFW+14]. Zhang *et al.* [ZHPL12] defined a new multidimensional network which contains attributes of nodes and edges. Nodes attributes were defined as dimensions in a graph cube while edge attributes were defined as dimensions in classical cube. In order to deal with this network, they proposed the model called `NestedCube`. To analyze on `NestedCube`, they proposed bidirectional two-ply OLAP query. This kind of OLAP query includes from node to edge and from edge to node. It means that a user can first perform OLAP query on outer graph cube. This obtains a measure network where nodes are the grouping of the same values on attribute and edges are the shared information between any two nodes. Finally, selected shared information can be aggregated as a data cube. Likewise, a user can perform OLAP query from the inner data cube to the outer graph cube. Wang *et al.* [WFW+14] introduced a new conceptual `Hyper Graph Cube` model. It is able to capture queries of all the aforementioned three categories into one model. To develop this model, they formally defined two types of dimensions in attributed graphs: vertex dimensions and edge dimensions. The aggregated graph is a multigraph, where several edges can be between two nodes. The `Hyper Graph Cube` belongs to topological OLAP.

Morfonios *et al.* did research on social bookmaking systems and they were also pioneers in the field of OLAP on information networks [MK08]. They proposed going beyond classical searches for resources based on keywords to exploring social data starting from any type of entity (user, resource or annotation) and requesting aggregated views of related entities based on the relationships defined between entities. They mapped this type of social searching to OLAP query processing and they studied various ways to support on-the-fly aggregations of social data. They described how data cubes can be used for precomputing and materializing the results of all possible aggregate queries over social data. In a similar way, Wu *et al.* worked on user profiles on social networks [WSR+12]. They proposed an OLAP serving system, called `Avatara`, to handle many and small cubes. The system provides a simple, expressive grammar for application developers to construct cubes and query them at scale.

Yin *et al.* criticized Chen's model to handle only homogeneous networks [YWZ12]. They defined the concept of entity dimensions to complete informational and topological dimensions and to handle heterogeneous networks. They also introduced two OLAP operations: `Rotate` to convert entities into relations and the inverse and `Stretch` to discover implicit relationships between entities. The third contribution consisted in two new models: `HMGraph OLAP`, a new multidimensional model of data warehouse for heterogeneous networks, and `HMGraph Cube`, a model for aggregating cubes of graphs.

Beheshti *et al.* blamed the existing approaches supporting only multi-dimensional and multi-level queries on graphs, not providing a semantic-driven framework and not supporting a language for n-dimensional computations [SMRBHRM12]. N-dimensional computations are frequent in OLAP analysis. For example, it could be interesting to analyze the reputation of a book, an author, or a publisher in a specific year. Such a query requires supporting n-dimensional computations on graphs, providing multiple views at different granularity levels. So authors proposed a graph data model, called `GOLAP`, extending decision support on multidimensional networks and considering both objects and links. They used the concepts of folder and path nodes to support multidimensional and multi-level views and to provide network semantics. Traditional dimensions and measures are redefined according to the relationships among entities. Finally, they also extended the language `SPARQL` to support n-dimensional computations on graphs and proposed new OLAP operations (`assignment`, `function`, `update`, `upsert`).

Yin and Gao worked on iceberg query in large graphs, which focus on the iceberg vertices for which aggregation of nodes's types and attributes [YG14]. They propose the definition of iceberg cube on heterogeneous information networks. The iceberg cube is realized by pruning on the two parts. First, random walk is used to aggregate the nodes in the networks for approximate computation. Consequently, by defining the meta-path

between different types of node, the probability of reaching another vertex with respect to the meta-path reflects how close the two vertexes are.

Ghrab *et al.* defined the multidimensional structure in the context of heterogeneous attributed graphs [AGZ15]. This approach is used for the extension of the property graph. In graphs, they distinguished two types of dimensions. The first one is Node-based dimensions, which are represented by the attributes of nodes. The second one is Edge-based dimensions, which are represented by the attribute of edges. Three measures has been presented in their model. The first measure is similar to the traditional measures such as the average of a movie. The other measures capture the topological properties of graphs and are obtained by applying graph algorithm. The last measure is presented by Chen *et al.* [CYZ$^+$08].

Kampgen *et al.* retrieved statistical information from multiple linked data sources to insert them into a data warehouse [KH11]. The authors proposed a mapping between linked data and multidimensional models by using the RDF Data Cube vocabulary in order to take into account the data semantic. It is regrettable that the mapping is relatively conventional with only traditional OLAP concepts without taking into account the topological structure of networks.

Kaya and Alhajj integrated two databases, DBLP and CiteSeerX, in order to have bibliographic information on major computer science conference proceedings and journals and to include citations, co-authorships, addresses, and affiliations of authors [KA14]. They developed three different information networks (Authors, Topic and Venue) with a cube based modeling method. In the networks, each node may represent an author, a topic or a venue with respect to the kind of network. Next, each node is represented by a data cube. In order to appropriately analyze the data cube, the OLAP technology is utilized. After that, they automatically found relevant persons, topics and venues for each network by the use of a multi-agent based algorithm.

3.5 Discussion

To conclude the state of the art about OLAP on information networks, we propose a comparison between the approaches [LJMF15]. In order to compare the previously cited approaches, we introduce criteria.

Two first criteria recall the data or domains which are studied and the type of networks built from these data (homogeneous or heterogeneous).

The other criteria deal with how information networks are designed in the multidimensional model and show how works adapt OLAP to networks. For each approach, the type of measure and the associated aggregation function are stated. There can be several kinds of dimensions: informational dimension (I), topological dimension(T) and entity dimension (Te).

Some works focus on efficient computation of cubes and users' queries and propose a full materialization (F), a partial materialization(P) and a non materialization (N). Finally, some specific OLAP tools or operations are sometimes created to answer users' queries. The empty cell means that authors do not mention about that criteria. With these criteria, we build a table (see Table 3.2 p.43).

In Graph OLAP, most approaches are dealing with bibliographic data because they are well known and constitute a suitable example of information networks. Usually co-authors network is built and there are different attributes such as time, venue, area and etc. Zhao *et al.* added an attribute, namely the productivity, by discretizing the publications number of an author into four different buckets (Excellent, Good, Fair and Poor). Sometimes approaches are dealing with other kinds of data such as images [JHC+10], social networks [MK08, WSR+12], movies networks [AGZ15] and statistical data [KH11]. In preprocessing, Kampgen *et al.* mentioned an ETL process for extracting, transforming and loading linked data into a data warehouse. Likewise, Ghrab *et al.* mentioned a Graph ETL process by combining the graph from external data sources that might be have various formats.

The two main limits of the studies [CYZ+08, ZLXH11, QZY+11, JHC+10, THP08, WSR+12] are that only homogeneous networks are built and usually only one network. We think, it would be better to build heterogeneous networks as proposed in [MK08, YWZ12, SMRBHRM12, YG14, AGZ15] and to build from a same database several networks (some of them being heterogeneous) in order to take several points of view into account. Studying both co-authors network, citations network, topics network and conferences network could give several points of view of a same database. But, to our knowledge, no approach does it.

The multidimensional model of networks is quite different from the traditional one with a redefinition of dimensions, measures and operations to adapt them to graphs and networks. As we said, J. Han's team was the first one to investigate OLAP on information networks and they introduced basic definitions in the general framework called `Graph OLAP` [CYZ+08]. The `Graph OLAP` framework was formally used by some other research teams and we found the same concepts of topological and informational dimensions,

specific measures and aggregation functions. Most of the time, the measure is a graph or comes from the graph theory such as a centrality degree [QZY+11], a number of relations [MK08] etc. When the measure is a graph, all approaches defined an aggregation function adapted to graphs. We think that the model must take into account many types of measures and not only one [AGZ15]. For each type of measure there should have an adapted aggregation function. For example, when the measure is a centrality degree, how can it be aggregated when a roll-up is done ? The aggregation function of a graph should also take both entities and structure into account. Another example is to cluster entities into groups that share similar properties and then it is possible to have an aggregation function like that of Jin [JHC+10]. In particular of dimensions, they are obtained from attributes of nodes. Unlike, Zhang *et al.* and Wang *et al.* defined on the dimensions, which are extracted from attributes of nodes and of edges.

Only one approach, that of Yin *et al.* [YWZ12], completed dimensions with the concept of entity dimension to handle heterogeneous networks. They also included in the multidimensional model two fact tables: one for entities and one for relationships between entities. However, they didn't mention attributes of edges.

With the introduction of topological dimensions, authors introduced topological OLAP operations. More, Tian *et al.* proposed new operations for summarizing graphs and users can freely choose the interesting attributes and the relationships [THP08]. In contrast, Yin *et al.*, Beheshti *et al.* and Ghrab *et al.* proposed new operations to view knowledge inside graph cubes [YWZ12, SMRBHRM12, AGZ15].

Other contributions focus on OLAP analysis on information networks. However, they still lack some limitations. To sum up the short related work about OLAP on information networks, we can add two remarks.

The first remark is about the slowly changing dimension problem. To the best of our knowledge, the existing approaches in Graph OLAP are not complete with this problem.

The second remark is about the visualization of a multidimensional and multilevel view over graphs. For example, a cube, with a venue dimension and time dimension, can contain a cell for (ICDE, 2008) and another one for (DOLAP, 2008) cell. In the first Graph OLAP approaches, in each cell there is a graph showing collaborations between authors for this venue and this year. Between two authors, we can see the collaborations only according to the venue and the year, we do not see a global view of the collaborations. Furthermore, Wang *et al.* proposed a graph with multiple edges. However, their approach needs to summarize a set of graphs with multiple edges and it is a complex

task. In contrast, Zhang *et al.* used a single graph as input rather than a set of graphs. Kaya *et al.* presented three networks where each node is represented by a cube.

We would like to point out that this thesis go deeply in OLAP analysis on information networks. In this thesis we investigated bibliographic data that can extract from several bibliographic databases in order to build several networks. This thesis proposed a new way to do Graph OLAP and it is called *Graphs enriched by Cubes (GreC)*. The global idea is that each node or each edge is couple with a cube according to user's requirements. It allows the user to quickly analyze information that has been summarized into cubes and by viewing the graph. We propose to view the first graph as a homogeneous network because it can be viewed from a network to others by using operations while an heterogeneous network shows a whole data at the same time. Our focus is more on the changing information over time. This thesis is an effort to enable Graph OLAP operations such as informational and topological operations to *GreC*.

3.6 Conclusion

OLAP technologies are widely used in a variety of application domains. There are not many studies that use the data that is coming out of information networks using OLAP technologies. In this chapter, we described the definitions of OLAP analysis on information networks and it is called Graph OLAP. Moreover, we explained the operations of Graph OLAP and the base technologies that we use in our work in order to provide background knowledge. Subsequently, we discussed the comparison between traditional OLAP and what Graph OLAP is or should be.

The next part of the chapter, we summed up the work related to OLAP on information networks (detailed in Section 3.4). We proposed a comparison between the approaches. To the best of our knowledge, studies that are conducted attempt to use OLAP on information networks, are not resolving the slowly changing dimension problem. Furthermore, the visualisation of a multidimensional and multi-level view over graphs are developed by showing a graph in each cell. We do not see a global view of graph. We see the opportunity to fill the gap in the literature by proposing a new way to do Graph OLAP. In a different and complementary way, our proposal consists in enriching graphs with cubes. Indeed, the nodes or/and edges of the considered network are described by a cube. It allows interesting analyzes for the user who can navigate within a graph enriched by cubes according to different granularity levels, with dedicated operators. In the next chapter, we discuss the details of the construction of our proposal.

TABLE 3.2: Works about OLAP on information networks

Paper	Data	Network	Measures	Aggregation function	Dimensions			Materialization			Operations
					I	T	Te	F	P	N	
C. Chen 2008 Graph OLAP [CYZ+08]	Bibliographic data	Homogeneous	Coming from	Aggregated graph	X	X		X	X	X	Topological
P. Zhao 2011 Graph Cube [ZLXH11]			graph theory		X	X		X	X		
Q. Qn 2011 [QZY+11]			or a graph		X	X			X		
X. Jin 2010 Visual Cube [JHC+10]	Images	Homogeneous	Image or image features	Clustering aggregation	X						Classical
Y. Tian 2008 [THP08]	Bibliographic data	Homogeneous	Graph	Aggregated graph	X	X					SNAP k-SNAP
K. Morfonios 2008 [MK08]	Social networks	Heterogeneous	Number of relations	COUNT	X			X			Classical
L. Wu 2012 Avatar [WSR+12]	Social networks	Homogeneous	Classical	Classical	X						Classical
M. Yin 2012 [YWZ12] HM Graph OLAP HM Graph Cube	Bibliographic data	Heterogeneous	Graph	Aggregated graph	X	X	X		X	X	Rotate Stretch
S. Beheshti 2012 GOLAP [SMRBHRM12]	Bibliographic data	Heterogeneous	Graph	Aggregated graph	X	X					Assignment Update Upsert
B. Kampgen 2012 [KH11]	Statistical data		Classical	Classical	X						Classical
J. Zhang 2012 [ZHPL12]	Social networks	Homogeneous	Graph	Aggregated graph	X	X					Classical
M. Kaya 2014 [KA14]	Bibliographic data	Homogeneous	Classical	Classical	X						Classical
Z. Wang 2014 [WFW+14]	Social networks	Homogeneous	Graph	Aggregated graph	X	X					Classical
D. Yin and H. Gao 2014 [YG14]	Bibliographic data	Homogeneous	Graph	Aggregated graph	X						Classical
A. Gharb 2015 [AGZ15]	Movie networks	Heterogeneous	Graph	Aggregated graph	X	X					Classical

Chapter 4

Graphs enriched by Cubes

4.1 Introduction

Graph OLAP refers to the use of OLAP for the multidimensional analysis of information networks. A first approach that has been proposed in the literature consists in building cubes of graphs and exploring them. Our approach, Graphs enriched by Cubes (GreC), takes a completely different way in order to analyze data generated in the information networks by using OLAP philosophy. GreC approach consists in enriching graphs with cubes. Indeed, the nodes or/and edges of the network considered are valuated by cubes.

To understand clearly the contribution, there are four main aspects in GreC. First, as all similar approaches of Graph OLAP, GreC takes into account the structure of the network in order to do topological OLAP operations and not only classical or informational OLAP operations. Secondly, GreC proposes a global view of a network considered with multidimensional information in the different way of other Graph OLAP approaches that propose a global view of a cube with parts of the graphs. Thirdly, unlike any approaches, the slowly changing dimension problem is taken into account in order to explore a network. Lastly, as some similar Graph OLAP approaches, GreC allows data analysis that takes into account the evolution of the network because our approach keeps the evolution when the user chooses to consider time dimension in the cubes.

As a result, we describe an overview of the overall process in Section 4.2. It is a user-centric process and it needs a graph model as an input data in order to build graphs enriched by cubes. In this thesis, we use bibliographic data as a running example. Therefore, Section 4.3 presents the existing graph models for bibliographic data and

compares them according to different criteria. Afterwards, we present our graph model for bibliographic data in order to fulfil the problems of bibliographic data as explained in Section 2.2. Section 4.4 presents the meta data used for building graphs enriched by cubes and for determining the generic interface. Section 4.5 gives the definition and notations for our approach. Section 4.6 presents types of measures. Consequently, section 4.7 explains the way to compute the graphs enriched by cubes. Then, in Section 4.8, we describe the extension of OLAP operations to be used in our approach of graph enriched by cubes. Finally, we draw the chapter to a close by discussing the process of graph enriched by cubes in Section 4.9.

4.2 A user-centric process

The process for graphs enriched by cubes is a user-centric process where the end-user's needs correspond to a focus at their requirements. The process is depicted in Figure 4.1 p.47. The major components of our process are described as follows.

A **PRE-PROCESSING**

Usually, ETL process is to extract, transform and load data into data warehouses. In our context, ETL process is used to integrate data from data sources into XML files and it is used to load data from such XML files into a graph database. We first access bibliographic databases to extract data into XML files. Bibliographic databases might have various formats (e.g., XML as for DBLP or a Web page as for ACM, etc.). We first start at DBLP to select some papers. DBLP lacks some information e.g., missing institutions, we need to get the institutions from ACM by comparing with a title of a paper. DBLP and ACM do not provide the area of venues, we get this data from Microsoft research area according to the venue name. Then we create the XML files in order to collect all data in the same format. After integration, data is loaded into a graph database. The data is then formatted following our graph model (as explained in Section 4.3 p.48).

B **GRAPHS with CUBES PROCESSING**

In our approach, we consider a network which is enriched by cubes. Therefore in our framework, we do not build a data warehouse but we create cubes; more precisely one cube for each node or edge according to the network considered as presented in Section 2.5.2 p.23. A graph enriched by cubes may be used easily to perform OLAP operations on a network and it provides multiple network views at different levels of granularity. It considers a single graph rather than a set of graphs. A graphs enriched by cubes depends on the facts. Each fact has a graph

FIGURE 4.1: GreC Process

and cubes. We computed all graphs which are useful for users. For example, the fact can be the co-authorship. Co-authorship is a network where nodes are authors and an edge between two of them indicates that their papers have been written together. Consequently, we store the graphs and the cubes in a graph databases. For each fact, its first graph with its cubes is stored using own graph database instances. We use two particular graph databases, one to store the graph for a fact, the other stores the cubes. Although a graph database supports a separate between subgraphs of the same database by using different label, we need a database per a graph of a fact because this can save time in order to answer user's requirements. Then, a first graph is built (we will show details after). The

network is the co-authorships network enriched by a cube for each edge in order to count the number of papers written by two authors according to the chosen dimensions, for instance, sessions, years and venues. For this fact, it has no sense to build a cube for each node (author) because the fact focuses on the papers returned between two authors. When the fact is the scientific production, the network is the authors network and cubes are created both for nodes and edges. It has a sense to count the scientific production of an author or between two authors. In order to view the constructed network from different perspectives, dimensions of cubes allow to perform multidimensional analysis over networks. For enriched graph computing, we propose a new algorithm (see Section 4.7.1 for more details). Finally, graphs with cubes are sent as the result to the OLAP analysis interface.

C **NAVIGATION**

The OLAP interface manages both the user's needs and interactions, the input and the output of graphs with cubes during analysis. The OLAP interface provides information (facts, measures and etc.) according to the meta data. The starting point is that a user selects a fact. This determines the graph. With the fact selected, the interface proposes the possible measures for a fact and the possible dimensions for cubes. After requirements defined, the interface uses these requirements to find the first graph and cubes from graph databases (see number 1). For this requirement, a result is returned to the OLAP interface (see number 2). Finally, the interface allows users to explore graphs and cubes from different views with OLAP operations. While a user navigates to the graph and cubes, the interface connected to graph databases to get the answer to user (see number 3).

4.3   A graph-based model

Most works about Graph OLAP focused on homogeneous networks. However, a heterogeneous representation is much richer. For example, a representation of a bibliographic network may contain nodes corresponding to different entities such as authors, papers and venues. There are different relationships among those nodes. Clearly, on one hand, a heterogeneous network makes its powerful; on the other hand it is also much challenging for many purposes and it allows users to extract several networks. In recent years, there has been an increasing interest for bibliographic networks. In this section, we will examine some graph model of bibliographic data. Then we present our graph models.

TABLE 4.1: Comparison about graph models for bibliographic data

Works and Properties	[SHZ$^+$09]	[QZY$^+$11]	[XY12]	[SMRBHRM12]	[YG14]	[YG14]	our
Information from a networked model							
- Author	×	×	×	×	×	×	×
- Paper	×	×	×	×	×	×	×
- Venue	×	×	×	×	×	×	×
- Keyword	×	×	×	×		×	×
- Time				×		×	×
- Institution		×		×		×	×
- Location			×	×		×	×
- Citation				×			×
Attributed graph				×	×		×
Bibliographic data problems							
- Several values in the same property	×	×	×	×	×	×	×
- A value changing over time						×	×

4.3.1 The existing models for bibliographic data

To sum up works related to bibliographic data modeling, we present a comparison between the different models in Table 4.1 p.49. The first criteria recalls the information which can be extracted from each model. We are interested in the works which designed the models as heterogeneous networks. The other criteria deals with how a network designed in the attributed graph. The last criteria shows what problems of bibliographic data (as explained in Section 2.2) can be solved by the model.

Most models deal with data as authors, papers and venues because they are the main information of bibliographic data. However, there are other kinds of data which is useful such as keywords, institutions, citations and etc.

As we said before, all models considered are designed as heterogeneous network. It is obvious, they are not attributed graph except Beheshti *et al.* [SMRBHRM12]. An attributed network is a network where nodes and edges are described by attributes. For example, a node stands for an author that contains attributes including author's name or age. An edge between authors and papers is described by attributes which can be the order of authors and institutions.

With the introduction of the problems of bibliographic data in Section 2.2, there are two problems: an entity concerns many different values in the same property; and a value of an entity is changing over time. All models can deal with an entity concerns many different values in the same property by creating a new node and a new edge. Only one model, that of Tao *et al.*, can solve the problem when a value is changing over time such as a change of institution. To do this, they defined an edge of institution with paper while others defined an edge of institution with author.

4.3.2   The proposed model

We present our graph model, which is for an attributed and heterogeneous network. The graph model contains four types of nodes (author, paper, venue, keyword) and four types of edges among these nodes. Each node and edge are described by attributes. Figure 4.2 p.51 shows the details of nodes and edges of our graph model. The attributes of nodes are: author (author's name); paper (year, title and abstract); venue (venue's name, year, research area) and keyword (keyword, category). The edges are constructed based on the relationships between nodes. The edges represent the writing relationship between authors and papers, the citation relationship between papers, the containing relationship between paper and keyword, and the publishing relationship between papers and venues. For instance, when the edge represents the writing relationship between authors and a paper, the edge has attributes like the order, institutions and countries. Considering institution and country attributes, there are close to a dimension concept in traditional data warehouse.

The attributes of a paper are the title, the year and the abstract. Year is an attributed dimension associated with time hierarchy. This model defines institution as an attribute on edge between author and paper to support query when authors change institutions. Our graph model allows users to extract different networks such as co-authorships network, institutions network and etc. Also this model can deal with two problems of bibliographic data. First, an entity concerns many different values in the same property. Secondly, a value is changing over time. Figure 4.3 p.51 illustrates an attributed graph capturing a bibliographic network.

4.4   Meta data

Basically, meta data is structured information that describes, explains, and makes it easier to retrieve, use, or manage data sources. Meta data is often called data about data or information about information. The term meta data is used differently in different works. Some works use meta data for machine understandable information. Some works use meta data for records which describe resources. An important reason for creating descriptive meta data is to facilitate discovery of relevant information. In addition to resource discovery, meta data can help to organize resources, to facilitate interoperability and legacy resource integration, to provide digital identification, and to support archiving and preservation.

FIGURE 4.2: Graph model for GreC on bibliographic data

FIGURE 4.3: An attributed graph for a bibliographic network

In this thesis, meta data has two aims. First, it is used to build graphs and cubes. To achieve this, meta data is a map which makes a correspondence to data that is stored in graph database. Second, meta data is important to determine the generic interface. Our meta data is designed according to information typically found in the data warehousing approach. Our approach is to present graphs with cubes, we use some elements which concerns to the elements of graph. Our meta data will be arranged according to an entity-relationship model. In the next step, we check the main structural components of our meta data and their relationships are identified.

Let's summarize and organize the information found in the meta data. The most important identified entities (objects of interest) are:

- facts

- measures

- dimensions

- hierarchies

- graphs

- levels

The identified relationships (and the entities linked by them) are:

- facts and measures

- facts and graphs

- facts and dimensions

- dimensions and hierarchies

- hierarchies and levels

- levels and levels

We use entity-relationship (ER) modeling to visualize the entities with their attributes and the relationships identified above. Conceptual ER models information gathered from the requirements. Entities and relationships modelled in such ER are defined around the work's need. The need of satisfying the database design is not considered yet. Conceptual ER is the simplest model among all. The complexity increases from conceptual

to logical. We understand at high level what are the different entities in our data and how they relate to one another in the conceptual model.

An entity-relationship model is an abstract representation of data typically used for data modeling. The entities are displayed as boxes, the attributes as ellipses and the relationships as diamonds while edges link the corresponding elements. The entities and relationships identified so far are displayed in Figure 4.4 p.55. Underneath the conceptual meta data, the logical model is presented in Figure 4.5 p.56.

The relationships are presented in the following:

- A fact has a many-to-many relationship with a graph because a fact can concern many graphs, and a graph can be referred to by many facts. Likewise, a fact has a many-to-many relationship with dimensions because a fact can have many dimensions.

- A fact also has a many-to-many relationship with a measure, a measure can be used to many facts.

- A graph refers to many dimensions because a graph gives more than one dimension, and a dimension can be referred to by many graphs.

- A dimension has a minimum of one hierarchy. A hierarchy belongs to only one dimension.

- A hierarchy has a minimum of one level and it has a maximum many levels. A level can be referred to by many hierarchies.

- Each level is related to a minimum of zero and a maximum of many levels.

The entities are described by attributes in the following:

- The FACTS table stores the name of facts (FName), the node type corresponding to the fact(NodeType), edge type corresponding to the fact (EdgeType) and a position for a cube corresponding to the fact (PosForCube). The possible values of a position for a cube can be edge, node and both node and edge. The facts are defined as homogeneous networks because we can go from a view to other views by using operators.

- The MEASURES table stores the name of measures (MName). It keeps a position in a graph to get a set of measure's value. This concerns to three attributes: name of a node or en edge (PosInGraphName), position in a graph (PosInGraph)

and attribute's name (AttributeName). This table keeps a computation function (Function) because there are different types of functions according to measures. Note that some measures do not need the position for getting a set of measure's value (we explain in the following)

- The GRAPHS table stores the graph name (GraphName).

- The DIMENSIONS table, we store the name of dimensions (DimName).

- The HIERARCHY table stores the name of hierarchies (HiName). It stores a position for getting the values. This concerns to three attributes: name of a node or en edge (PosInGraphName), position in a graph (PosInGraph) and attribute's name (AttributeName). This table stores a position for a cube (PosForCube) because a position for a cube may be changed if a user select a topological dimension. Note that some hierarchies do not need a position for a cube because they do not change the structure of a graph.

- The LEVELS table stores the name of level (LName). It stores a position for getting a set of values. This concerns to three attributes: name of a node or en edge (PosInGraphName), position in a graph (PosInGraph) and attribute's name (AttributeName).

After that we explain a small data example for our implementation in Figure 4.6 p.57. The details are presented as follows:

- If a fact is co-authorships, a facts table explains that this fact is co-authorships network which a node is author and an edge between two of them indicates they coauthored papers. It also defined this network has cubes on edges.

- With the fact is co-authorships, measures can be the number of papers, degree centrality and etc. If the measures are numeric, they have a position on the properties graph in order to get the values for computing a total number for a cube. For example, in order to compute the total number of papers, we have to get the different papers from paper node and title attribute.

- The fact is co-authorship, it is associated to $G1$ and $G2$. In $G1$, it gives time dimension, while $G2$ gives time and venue dimension.

- Basically, a dimension is structured according to hierarchy. For example, institution dimension is structured with institution hierarchy. The institution hierarchy is defined into levels: institution name and country.

- A level may have higher levels. For example, country is a higher level of institution.

FIGURE 4.4: ER model for meta data

FIGURE 4.5: Logical model of meta data

FIGURE 4.6: Data example of the implementation for the meta data

4.5 Definitions and notations

Classical data warehouses are usually created by designing a multidimensional model containing facts, dimensions and etc. Our proposal is graphs enriched by cubes, it is not the classical data warehouses and we do not build a conceptual model for multi-dimensional analysis. However, we focus on building cubes. They can be built on the fly and they can be computed in a preprocessing step. In this section, we consider an extending multidimensional structure for our approach. In order to explain clearly the later approach, we provide here the definitions and notations that allow us to present the principle and algorithms of our contributions. We present the notations and definitions for each element in our approach by implying data warehousing approach as follows.

Fact. In classical OLAP, a fact is the subject of the analysis, which is modelled as a fact table. In our concept, we propose to view these facts by a network in order to face different information and to describe the interconnection among information. Therefore, a fact is also the subject that we observe.

Definition 4.1. (Fact) A set of facts $\mathcal{F}$ is defined by $\{F_f\}$ where, F_f is a fact and $f \geq 1$.

Example:
$\mathcal{F} = \{F_1,\ F_2\}$
F_1 = Co-authorship.
F_2 = Production.

For example, interesting facts from bibliographic data can be the co-authorships or the production of authors.

According to the meta data, the choice of the fact determines which the characteristics of the graph are computed: the nodes, the nodes or/and edges valuated by cubes. Since we only consider one fact for each analysis, let us note F to precise the fact considered in the following notations.

Definition 4.2. (GreC) $GreC^F$ is the graph enriched by cubes for the fact F, which is defined by $(G^F,\ \mathcal{C}^F)$ where,

- $G^F = (V^F,\ E^F)$ is a graph of the fact F where V^F is a set of nodes and E^F is a set of edges.

- $\mathcal{C}^F = \{C_{V^F} \ \cup \ C_{E^F}\}$ is a set of cubes for the graph G^F. It corresponds to the union of C_{V^F} and C_{E^F}. C_{V^F} is a set of cubes which valuates the set of nodes V^F. C_{V^F} can be empty according to the fact chosen. C_{E^F} is a set of cubes which valuates the set of edges E^F. C_{E^F} can be empty according to the fact chosen.

Example:

- The fact is co-authorship.

  $GreC^F = GreC^{co-authorship}$

  $G^F = (\{\text{author}\}, \{\text{links between authors}\})$

  $\mathcal{C}^F = \{C_{links\ between\ authors}\}$

- The fact is production.

  $GreC^F = GreC^{production}$

  $G^F = (\{\text{author}\}, \{\text{links between authors}\})$

  $\mathcal{C}^F = \{C_{authors} \ \cup \ C_{links\ between\ authors}\}$

Table 4.2 p.60 shows two examples of facts including co-authorships and productions. To analyze co-authorships, the meta data determines this network where a node is an author and an edge is the collaboration (see FACTS table, line 1, Figure 4.6 p.57 according to the attributes NodeType and EdgeType). If the fact considered is co-authorships, cubes are provided only for edges (see FACTS table, line 1, in Figure 4.6 p.57 according to the attribute PosForCube). Note that these cubes will be fulfil by measures. There is no cube for nodes because we are focusing on the relationship among authors. On the contrary, if the fact is the production, the idea is to have a cube for each author representing the own publications and a cube for the relationship among authors. We define in the following the concepts of measure, dimension and cube.

TABLE 4.2: Examples

Facts	Co-authorship	Production
Example of measures	The number of papers written by two authors	The number of papers
Type of graphs		

Measures and cubes.

Definition 4.3. (Measure) A set of measures $\mathcal{M}$ corresponding to the fact F is denoted as $\{M_m, \ m \geq 1\}$. M_m is defined by $M_m = (M_m^{name}, \ M_m^g)$ where,

- M_m^{name} is the measure name.

- M_m^g could be a graph-specific function such the centrality algorithm or could be a function computing a numerical value.

Example.
$\mathcal{M} = \{M_1, \ M_2\}$
$M_1 = (M_1^{number \ of \ papers}, M_1^{Numeric \ function})$
$M_2 = (M_2^{degree}, M_2^{Degree \ algorithm})$

$\mathcal{M}$ is a set of measures linked to a fact F. For example, there are two measures in $\mathcal{M}$. First, M_1 is the number of papers which is a numeric measure. It refers to numeric function in order to compute the total result (see MEASURES table, line 1, in Figure 4.6 p.57 according to the attributes Function). Second, M_2 is degree centrality. To compute this measure, it refers to the degree algorithm. Measures are computed according to a network and their functions are applied to a network. For example, from the authors network with M_1, if we does an OLAP operation like a roll up in order to see the institutions network, these measure are recomputed for the institutions network by using $M_1^{Numeric \ function}$. $M_1^{Numeric \ function}$ is a function which is used to authors network.

The value of a measure is placed in a cell of a cube, cells are structures determined by a set of dimensions. C_{VF} and C_{EF} are two sets of cubes that have the same structure. Let us note C^F this structure.

Definition 4.4. (Cube) A cube C^F is defined by $(M_m, \ \mathcal{L}_{D^{cube}}^{chosen})$ where,

- M_m is a measure.

- $\mathcal{L}_{D^{cube}}^{chosen}$ is the set of levels considered, for each dimension of $\mathcal{D}^{cube}$ that the set of dimensions considered for the cube.

Indeed each dimension can be organized according to a hierarchy composed of various levels granularity(the detailed notation will be given after). Let us mention that the concept of dimension is used into two different ways: the dimension for cubes and the dimension for graph. Dimensions provide the possible perspectives in a cube. As we said before, one type of dimension in Graph OLAP is a topological dimension coming

from the attributes of topological elements. In our approach, we adapt the definition of dimension for the graph analysis and for cubes.

So, $\mathcal{D}^{cube}$ defines the set of dimensions linked to the cube. Let us note $\mathcal{D}^{graph}$ for the set of dimensions for the graph.

Dimensions. The notations of dimensions are represented by the followings.

Definition 4.5. (Dimension) A set of dimensions $\mathcal{D}$ linked to the fact F is denoted as $\{D_d\}$, $d \geq 1$. A dimension D_d is defined by $D = (D_d^{name}, \mathcal{H}_{D_d})$ where,

- D_d^{name} is the dimension name.

- $\mathcal{H}_{D_d}$ is a set of hierarchies of D_d charectized.

Example of dimension:

- Example of D^{cube}:
  $D_1^{name} = \text{time}$
  $\mathcal{H}_{D_1} = \{\text{year}\}$

- Example of D^{graph}
  $D_2^{name} = \text{author}$
  $\mathcal{H}_{D_2} = \{\text{Person, Organization}\}$

D_1 is a time dimension and its type is for a cube. This dimension has one hierarchy. For D_2, author dimension which has a type for a graph has two hierarchies. One is person which is name of author and the other is organization as shown in Figure 4.7 p.63.

For each dimension, the set of associated attributes can be structured as a hierarchy. A hierarchy is usually structured into levels. We define as the following.

Definition 4.6. (Hierarchy) A set of hierarchies $\mathcal{H}_{D_d}$ of a dimension D_d is denoted as $\{H_{D_dh}, \ h \geq 1\}$. H_{D_dh} is defined by $H_{D_dh} = (H_{D_dh}^{name}, \{L_{D_dh}^l\})$ where,

- $H_{D_dh}^{name}$ is the hierarchy name.

FIGURE 4.7: Example of author dimension with two hierarchies (Person hierarchy and Organization hierarchy)

- $L^l_{D_{dh}}$ is a set of levels $L^l_{D_{dh}} = \{L^1_{D_{dh}}, \ L^2_{D_{dh}}, \ ...\}$ with $L^1_{D_{dh}} \prec \ L^2_{D_{dh}} \prec \ ...$ where $\prec$ expresses a total order on the levels, where $L^l_{D_{dh}}$ is a non-empty set of levels. Level names are unique.

Example of $H_{D_{time}h}$

$H^{name}_{D_{time}h} = Year$

$L^1_{D_{time}h} = Year$

Example of $H_{D_{organization}h}$

$H^{name}_{D_{organization}h} = Organization$

$L^1_{D_{organization}h} = Institution$

$L^2_{D_{organization}h} = Country$

$L^1_{D_{organization}h} \prec L^2_{D_{organization}h}$

Example of $H_{D_{venue}h}$

$H^{name}_{D_{venue}h} = Venue$

$L^1_{D_{venue}h} = Venue's \ name$

$L^2_{D_{venue}h} = Research \ area$

$L^1_{D_{venue}h} \prec L^2_{D_{venue}h}$

A fact can be examined through the dimensions. Let us consider co-authorships for example, the dimensions could be the time, the venue and the institution. Time and venue are defined to restrict the content of graph. Institutions concern an author. The institution is a topological dimension. The dimensions are defined with their respective levels: $\{year, \ all\}$; $\{venue's \ name, research \ area, \ all\}$; and $\{institution, \ country, \ all\}$, respectively. Note that each $L^l_{D_{dh}}$ comes from an attribute of node or of edge that belongs

to the same node or edge. For example, $L^1_{D_{venueh}}v$ and $L^2_{D_{venueh}}$ come from the attributes of venue node (see Figure 4.2 p.51)

To do analytics over graphs, multiple classification of graph measures were proposed in the literature. Here, we present a classification of graph measures, based on the type of the computation algorithm.

4.6 Types of measures

1. Numerical measures

   These measures are similar to the traditional measures such as the number of papers and number of authors.

2. Graph-based measures

   They can capture the properties of graphs and they are obtained by using graph algorithms. In this thesis, we are interesting in the centrality of nodes within a graph. The centrality of nodes, or the identification of which nodes are more "central" than others, has been a key issue in network analysis. It determines the qualified status of a node e.g., how important an author is within the co-authorships network. There are many types of the centrality concept such as degree, betweenness and closeness. We are going to details this measure.

   - **Degree Centrality** is the simplest concept, which is defined as the number of incident links upon a node ([Fre78]). It is the number of nodes adjacent to a given node:

   $$DC(i) = \sum_{j}^{N} x_{ij}$$

   where i is the given node, j represented all others nodes, N is the total number of nodes, and x is the adjacency matrix, in which x_{ij} is defined as 1 if the node i is connected to the node j, and .

   Let us see the co-authorships network as shown in Figure 4.8 p.65. Nodes in the network represent authors, and an edge between two of them indicates one or more publications written together. The value on the edge corresponds to the number of papers written together. For example, Jiawei Han has six

FIGURE 4.8: Example of co-authorships network

links. The result of the calculation of the degree scores is 6.

- **Betweenness Centrality** measures how often a given node sits between others. It relies on the identification of the shortest paths, and measures the number of them that passes through a given node [Fre78]. To faster calculation the betweenness scores of nodes, Brandes *et al.* proposed a new algorithm in order to reduce the time [Bra01]. This measure has been formalized as follows:

$$BC(i) = \sum_{i \neq j \neq k} \frac{g_{jk}(i)}{g_{jk}}$$

where i is the given node, g_{jk} is the number of shortest paths between two nodes j and k, and $g_{jk}(i)$ is the number those paths go through node i.

We give an example to calculate the betweenness score of J. Han ($i =$ J. Han) by using the co-authorships network in Figure 4.8. The first step is to compute the shortest paths between each pair of nodes (j, k) where $i \neq j \neq k$. The calculation finds multiple shortest paths if they have exactly the same distance. Table 4.3 p.66 shows the shortest paths of all pairs. For example, the shortest path between P.S. Yu and T. Wu is found over the direct tie with a path which goes through J. Han. For this pair (between P.S. Yu and T. Wu), a value of J. Han is equal:

$$\frac{g_{P.S.Yu,\ T.Wu}(J.Han)}{g_{P.S.Yu,\ T.Wu}} = \frac{1}{1}$$

TABLE 4.3: The shortest paths of all pairs by using the co-authorships network shown Figure 4.8

Node j	Node k	The shortest paths	$\frac{g_{jk}(J.Han)}{g_{jk}}$
Y. Yu	T. Wu	Y. Yu - J. Han - T. Wu	$1/2 = 0.5$
		Y. Yu - Y. Sun - T. Wu	
Y. Yu	Y. Sun	Y. Yu-Y. Sun	$0/1 = 0$
Y. Yu	H. Yin	Y. Yu -Y. Sun - H. Yin	$0/1 = 0$
Y. Yu	C. Li	Y. Yu - Y. Sun - C. Li	$1/2 = 0.5$
		Y. Yu - J. Han - C. Li	
Y. Yu	X. Yan	Y. Yu - Y. Sun - X. Yan	$1/2 = 0.5$
		Y. Yu - J. Han - X. Yan	
Y. Yu	P.S. Yu	Y. Yu - J. Han - P.S. Yu	$1/1 = 1$
T. Wu	Y. Sun	T. Wu - Y. Sun	$0/1 = 0$
T. Wu	H. Yin	T. Wu - Y. Sun - H. Yin	$0/1 = 0$
T. Wu	C. Li	T. Wu - C. Li	$0/1 = 0$
T. Wu	X. Yan	T. Wu - J. Han - X. Yan	$1/2 = 0.5$
		T. Wu - C. Li -X. Yan	
T. Wu	P.S. Yu	T. Wu - J. Han - T. Wu	$1/1 = 1$
Y. Sun	H. Yin	Y. Sun - H. Yin	$0/1 = 0$
Y. Sun	C. Li	Y. Sun - C. Li	$0/1 = 0$
Y. Sun	X. Yan	Y. Sun - X. Yan	$0/1 = 0$
Y. Sun	P.S. Yu	Y. Sun - P.S. Yu	$0/1 = 0$
H. Yin	C. Li	H. Yin - Y. Sun - C. Li	$0/1 =0$
H. Yin	X. Yan	H. Yin - Y. Sun - X. Yan	$0/1 =0$
H. Yin	P.S Yu	H. Yin - Y. Sun - P.S Yu	$0/1 = 0$
C. Li	X. Yan	C. Li - J. Han - X. Yan	$1/1 =0.5$
		C. Li - Y. Sun - X. Yan	
C. Li	P.S. Yu	C. Li - J. Han - P.S. Yu	$1/2 =0.5$
		C. Li - X. Yan - P.S. Yu	
X. Yan	P.S Yu	X. Yan - P.S Yu	$0/1 =0$

While the shortest path between Y. Yu and T. Wu is found two paths that has the same distance, but only path go through J. Han. Suppose that there are such two paths that go though J. Han. Therefore, a value of J. Han is equal $1/2 = 0.5$. The summarization of betweenness score of J. Han is the sum of the values of all pairs as show in table 4.3 p.66,

FIGURE 4.9: Co-authorships network: two sub-networks

$$BC(J.\ Han) = 0.5 + 0 + 0 + 0.5 + 0.5 + 1 + 0 + 0 + 0 + 0.5 + 1 + 0 + 0 + 0 + 0+$$
$$0 + 0 + 0 + 0.5 + 0.5 + 0$$
$$= 5$$

- **Closeness Centrality** measure for one given node how many steps away from others one are in the network. It relies on the length of the paths from a node to all other nodes in the network, and it is defined as the inverse total length [Fre78]. The original equation has been formalized as follows:

$$CC(i) = \frac{1}{\sum_j d_{ij}}$$

where i is the given node, j is another node in the network, and d_{ij} is the shortest distance between these two nodes. In this equation, the distances are inversed after they have been summed, and when summing an infinite number, the outcome is infinite. To overcome this issue while staying consistent with the existing measure of closeness, I took advantage of the fact that the limit of a number divided by infinity is zero. Although infinity is not an exact number, the inverse of a very high number is very close to 0. Table 4.4 p.69 shows The distance matrix for the nodes in the co-authorships network as show in Figure 4.9. The closeness score of all the nodes in the network is 0, it would be inaccurate to use this equation as a closeness measure for a disconnected network because the distance between nodes in a disconnected

network is infinite.

In our example, a bibliographic network may be a disconnected network which is composed of a set of sub-networks. For example, Figure 4.9 p.67 shows a co-authorships network, which contains two sub-networks. Freeman's algorithm is limited to compute closeness scores for disconnected network [Fre78]. Therefore, Opsahl *et al.* rewrite the closeness equation as the sum of the inversed distances to all other nodes instead of the inversed of the sum of distances to all other nodes [OAS10]. This measure has been formalized as follows:

$$CC(i) = \sum_{j \neq i} \frac{1}{d_{ij}}$$

where i is the given node, j is another node in the network, and d_{ij} is the shortest distance between these two nodes.

To exemplify this measure, table 4.5 p.69 shows closeness score of all nodes by using the co-authorships network in Figure 4.9 p.67. We give an example to calculate the closeness score of J. Han. The starting point is finding the shortest path from J. Han to others. The obtained results are 1, 1, 1, 2, 1, 1, 1, ∞, ∞ and ∞ (see table 4.3 p.66, line 2). To compute closeness score, it is calculated by the sum of each inversed distances except a number divided by infinity is zero. Therefore, his closeness score is:

$$CC(J.\ Han) = 1/1 + 1/1 + 1/1 + 1/2 + 1/1 + 1/1 + 1/1 + 1/\infty + 1/\infty + 1/\infty$$
$$= 6.5$$

TABLE 4.4: The distance matrix by using the co-authorships network shown Figure 4.9

	J.Han	Y. Yu	T. Wu	Y. Sun	H. Yin	C. Li	X. Yan	P.S. Yu	H. Lu	B. Yang	C.S. Jensen	Closeness
J.Han	-	1	1	1	2	1	1	1	∞	∞	∞	0
Y. Yu	1	-	2	1	2	2	2	2	∞	∞	∞	0
T. Wu	1	2	-	1	2	1	2	2	∞	∞	∞	0
Y. Sun	1	1	1	-	1	1	1	1	∞	∞	∞	0
H. Yin	2	2	2	1	-	2	2	2	∞	∞	∞	0
C. Li	1	2	1	1	2	-	2	2	∞	∞	∞	0
X. Yan	1	2	2	2	2	2	-	1	∞	∞	∞	0
P.S. Yu	1	2	2	2	2	2	1	-	∞	∞	∞	0
H. Lu	∞	∞	∞	∞	∞	∞	∞	∞	-	1	1	0
B. Yang	∞	∞	∞	∞	∞	∞	∞	∞	1	-	1	0
C.S. Jensen	∞	∞	∞	∞	∞	∞	∞	∞	1	1	-	0

TABLE 4.5: Closeness centrality by using Opsahl equation

	J.Han	Y. Yu	T. Wu	Y. Sun	H. Yin	C. Li	X. Yan	P.S. Yu	H. Lu	B. Yang	C.S. Jensen	Closeness
J.Han	-	1	1	1	0.5	1	1	1	0	0	0	6.5
Y. Yu	1	-	0.5	1	0.5	0.5	0.5	0.5	0	0	0	4.5
T. Wu	1	0.5	-	1	0.5	1	0.5	0.5	0	0	0	4.5
Y. Sun	1	1	1	-	1	1	1	1	0	0	0	7
H. Yin	0.5	0.5	0.5	1	-	0.5	0.5	0.5	0	0	0	4
C. Li	1	0.5	1	1	0.5	-	0.5	0.5	0	0	0	5
X. Yan	1	0.5	0.5	0.5	0.5	0.5	-	1	0	0	0	4.5
P.S. Yu	1	0.5	0.5	0.5	0.5	0.5	1	-	0	0	0	4.5
H. Lu	0	0	0	0	0	0	0	0	-	1	1	2
B. Yang	0	0	0	0	0	0	0	0	1	-	1	2
C.S. Jensen	0	0	0	0	0	0	0	0	1	1	-	2

4.7 Computing a graph enriched by cubes

We proposed a new way to analyze networks taking advantages from an OLAP technology. To achieve this, we first compute a graph for a fact. Then, we compute cubes for nodes and/or edges with respect to the fact. Finally, a graph and cubes are stored in a graph database (see Section 5.2.2 for more details).

In order to provide a running example of how data are organized in our approach, we refer to the bibliographic data as presented in Figure 2.1. Let us suppose that these papers have session and they are ordered by id as shown in table 4.6.

We provide different algorithms for building cubes according to measures. We present them in the following sections.

4.7.1 Graph computation

As we said before, bibliographic data has two problems: many values in the same property and changing value over time. In order to support these two problems, we use paths in algorithm for computing the aggregated graph. To build a first graph for analysis, we calculate a set of paths in the preprocessing step. We give the definition of path as follows:

Definition 4.7. (Path) A path P is defined on the heterogeneous network, and is denoted by $V_1 \xrightarrow{E_1} V_2 \xrightarrow{E_2} ... \xrightarrow{E_\lambda} V_q$. It defines a composite relation $E = E_1 \circ E_2 ... \circ E_\lambda$ between nodes V_1 and V_q, where $\circ$ denotes the composition operator on edges.

The structure of paths are defined from our graph model $G = (V, E, A_V, A_E)$ where V is the set of vertices, E the set of edges, A_V and A_E respectively the set of attributes describing nodes and edges. When a user defines a fact and a set of dimensions, we know a path structure according to meta data. If the fact is co-authorship and a dimension is time, a structure of path is *author* $\xrightarrow{write}$ *paper* (see the meta data in Figure 4.6) because a time is get from year attribute of paper node. If the fact is co-authorship and dimensions are time and venue, a structure of path is *author* $\xrightarrow{write}$ *paper* or *author* $\xrightarrow{write}$ *paper* $\xrightarrow{publish}$ *venue* because a time is get from year attribute of paper node and a venue is get from venue's name attribute of venue node.

To build a graph, we present an algorithm, BUILDGRAPH (Algorithm 1). It creates a graph $G' = (V', E')$ where $V' = \{(v_\alpha, P_\alpha)\}$, where $v_\alpha \in V$, $\alpha = 1, 2, ..., t$ and P_α is the set of paths of v_α and $E' = \{(e_{v_\beta - v_\gamma}, P_{\beta - \gamma})\}$, where $v_\beta \in V$, $v_\gamma \in V$ and $P_{\beta - \gamma}$ is the set of paths of the edge $v_\beta - v_\gamma$. We explain the algorithm followed by a running example. The steps of this algorithm are presented in the following.

TABLE 4.6: Example of papers listed in order of the id

Paper id	Details of papers	Session
Paper1	Bin Yang (Aalborg University and Fudan University), Hua Lu (Aalborg University) and Christian S. Jensen (Aalborg University), 'Probabilistic threshold k nearest neighbor queries over moving objects in symbolic indoor space', EDBT, 2010	Probabilistic and spatial database
Paper2	Jiawei Han (University of Illinois), Xifeng Yan (University of California) and Philip S. Yu (University of Illinois), 'Scalable OLAP and mining of information networks', EDBT, 2009	Data warehouse
Paper3	Tianyi Wu (University of Illinois), Yizhou Sun (University of Illinois), Cuiping Li (Remin University) and Jiawei Han (University of Illinois), 'Region-based online promotion analysis', EDBT, 2010	Data mining
Paper4	Yizhou Sun (University of Illinois), Yintao Yu (University of Illinois) and Jiawei Han (University of Illinois), 'Ranking-based clustering of heterogeneous information networks with star network schema', KDD, 2009	Data mining
Paper5	Yizhou Sun (University of Illinois), Jiawei Han (University of Illinois), Xifeng Yan (University of California) and Philip S. Yu (University of Illinois), 'Integrating meta-path selection with user-guided object clustering in heterogeneous information networks', KDD, 2012	Data mining
Paper6	Peixiang Zhao (University of Illinois), Jiawei Han (University of Illinois) and Yizhou Sun (University of Illinois), 'P-Rank: a comprehensive structural similarity measure over information networks', CIKM, 2009	Data mining
Paper7	Yizhou Sun (University of Illinois) and Jiawei Han (University of Illinois), 'RankClus: integrating clustering with ranking for heterogeneous information network analysis', EDBT, 2009	Data mining
Paper8	Bin Yang (Aalborg University and Fudan University) and Christian S. Jensen (Aalborg University), 'iPark: identifying parking spaces from trajectories', EDBT, 2013	Demonstration
Paper9	Hongzhi Yin (Peking University) and Yizhou Sun (Northeastern University), 'LCARS: a location-content-aware recommender system', KDD, 2013	Recommender system

---

Algorithm 1 BUILDGRAPH

Input: An heterogeneous multidimensional network $G = (V, E, A_V, A_E)$, a fact F, a measure M, a set of dimensions $\mathcal{D}$
Output: A graph $G' = (V', E')$ where $V' = \{(v_\alpha, P_\alpha)\}$ and $E' = \{(e_{v_\beta - v_\gamma}, P_{\beta-\gamma})\}$

1: Generate a set of paths (P) according to G, F, M and D
2: $V' = \varnothing$
3: **for** each $p \in P$ **do**
4:     **if** v_p not in V' **then**
5:         $V' = V' + (v_p, \{p\})$
6:     **else**
7:         add p at node v_p in V'
8:     **end if**
9: **end for**
10: $E' = \varnothing$
11: **for** each $s = 1$ to V'.size-1 **do**
12:     $list_s$ = get the values of object according to P {the considered objects depend on meta-data, for instance papers for the authors}
13:     **for** each $r = s + 1$ to V'.size **do**
14:         $list_r$ = get the values of object according to P
15:         **if** $list_s \cap list_r \neq \phi$ **then**
16:             $E' = E' + (e_{v_s - v_r}, \{P_s + P_r\})$
17:         **end if**
18:     **end for**
19: **end for**
20: Return G'

---

a. Concerning the input, the algorithm starts with the user's requirements defined through an interface that exploits meta data to present consistent possibilities for analysis to the user. Indeed the meta data is used in order to know the relationships between F, M, $\mathcal{D}$, etc. The selected requirements induce the specific structure of the path. Then the instances of the path are computed (in term of values).
For example, BUILDGRAPH takes as input the user's parameters. As previously, the fact can be the co-authorship, the measure is the number of papers written by two co-authors, the dimensions are the year, the venue and the session.

b. Then, a set of paths P is created from the structure of path at line 1.
A set of paths is generated with respect to the user's requirements. With the example of the parameters above, a set of paths is computed from *author* $\overset{write}{\rightarrow}$ *paper* $\overset{publish}{\rightarrow}$ *venue* which is get from the meta data. In our example, there are 26 paths as shown in table 4.7 p.74. These paths are listed in order, for example, $B.\,Ying \overset{write}{\rightarrow} paper1 \overset{publish}{\rightarrow} EDBT$ defined as path 1.

c. Subsequently, we explore the set of paths. For each path, we add a new node v_p with its path to V', if there is no such value (line 4-5). Otherwise, we simply

update a path id for the node v_p in V' (line 7).

This step is to compute a set of nodes. With co-authorships, author is a type of nodes (see FACTS table in the meta data, line 1). Refer to a set of paths in table 4.7 p.74, a list of nodes is computed from node's type of a fact which is author. For example, path 4 and 10 belong to the author named J. Han. A list of authors with their paths is created in table 4.8 p.75. We keep all paths of a node because they are used to compute edges.

d. After the loop, we compute E'. For each v_s in V', we compare the list of object's values with the adjacent v_s by using intersection operator.

To compute the edges, any two authors who wrote papers together, are added to a list of edges as shown in table 4.9 p.75. The edges are computed by using intersection operators. An example of an edge computation is shown as follows:

- To compute an edge of the fact, we get a type of an edge from meta data. The meta data shows a type of edge is "*paper/node/title*". This means that we can get a set of object's value from paper node where an attribute is title.

- For example, J. Han is concerned with a set of paths {4, 10, 13, 15, 19, 22} (see table 4.8 p.75). Associate to his paths, we compute a set of object's values from each path. Path 4 refers to paper2. Therefore, J. Han published paper2, paper3, paper4, paper5, paper6 and paper7. While Y. Sun concerns with a set of paths {8, 11, 14, 19, 20, 21, 26}, he published paper3, paper4, paper5, paper6, paper7 and paper9.

- After we compute a relationship among them. A set of papers of this relationship is computed in the followings.

  For J. Han and Y. Sun,
  := {*paper2, paper3, paper4, paper5, paper6, paper7*}∩
  {*paper3, paper4, paper5, paper6, paper7, paper9*}
  :={*paper3, paper4, paper5, paper6, paper7*}

e. The considered object's values depend on meta data. If the comparison result is not empty, we add a new edge $e_{v_s-v_r}$ with its paths to E'.

$e_{J.Han-Y.Sun}$:= {*paper3, paper4, paper5, paper6, paper7*}

TABLE 4.7: Example of a set of paths

Path id	author $\overset{write}{\to}$ paper $\overset{publish}{\to}$ venue
1	B. Ying $\overset{write}{\to}$ paper1 $\overset{publish}{\to}$ EDBT
2	H. Lu $\overset{write}{\to}$ paper1 $\overset{publish}{\to}$ EDBT
3	C.S. Jensen $\overset{write}{\to}$ paper1 $\overset{publish}{\to}$ EDBT
4	J. Han $\overset{write}{\to}$ paper2 $\overset{publish}{\to}$ EDBT
5	X. Yan $\overset{write}{\to}$ paper2 $\overset{publish}{\to}$ EDBT
6	P.S. Yu $\overset{write}{\to}$ paper2 $\overset{publish}{\to}$ EDBT
7	T. Wu $\overset{write}{\to}$ paper3 $\overset{publish}{\to}$ EDBT
8	Y. Sun $\overset{write}{\to}$ paper3 $\overset{publish}{\to}$ EDBT
9	C. Li $\overset{write}{\to}$ paper3 $\overset{publish}{\to}$ EDBT
10	J. Han $\overset{write}{\to}$ paper3 $\overset{publish}{\to}$ EDBT
11	Y. Sun $\overset{write}{\to}$ paper4 $\overset{publish}{\to}$ KDD
12	Y. Yu $\overset{write}{\to}$ paper4 $\overset{publish}{\to}$ KDD
13	J. Han $\overset{write}{\to}$ paper4 $\overset{publish}{\to}$ KDD
14	Y. Sun $\overset{write}{\to}$ paper5 $\overset{publish}{\to}$ KDD
15	J. Han $\overset{write}{\to}$ paper5 $\overset{publish}{\to}$ KDD
16	X. Yan $\overset{write}{\to}$ paper5 $\overset{publish}{\to}$ KDD
17	P.S. Yu $\overset{write}{\to}$ paper5 $\overset{publish}{\to}$ KDD
18	P. Zhao $\overset{write}{\to}$ paper6 $\overset{publish}{\to}$ CIKM
19	J. Han $\overset{write}{\to}$ paper6 $\overset{publish}{\to}$ CIKM
20	Y. Sun $\overset{write}{\to}$ paper6 $\overset{publish}{\to}$ CIKM
21	Y. Sun $\overset{write}{\to}$ paper7 $\overset{publish}{\to}$ EDBT
22	J. Han $\overset{write}{\to}$ paper7 $\overset{publish}{\to}$ EDBT
23	B. Yang $\overset{write}{\to}$ paper8 $\overset{publish}{\to}$ EDBT
24	C.S. Jensen $\overset{write}{\to}$ paper8 $\overset{publish}{\to}$ EDBT
25	H. Yin $\overset{write}{\to}$ paper9 $\overset{publish}{\to}$ KDD
26	Y. Sun $\overset{write}{\to}$ paper9 $\overset{publish}{\to}$ KDD

TABLE 4.8: Set of nodes

Nodes	set of path id
B. Ying	{1, 25}
H. Lu	{2}
C.S. Jensen	{3, 24}
P. Zhao	{18}
Y. Sun	{8, 11, 14, 20, 21, 26}
J. Han	{4, 10, 13, 15, 19, 22}
X. Yan	{5, 17}
P.S. Yu	{6}
T. Wu	{7}
C. Li	{9}
Y. Yu	{12}
H. Yin	{25}

TABLE 4.9: Set of edges

Edges	Set of object's values	Set of path id
P. Zhao, Y. Sun	{*paper*5}	{14, 16}
P. Zhao, J. Han	{*paper*5}	{15, 16}
Y. Sun, J. Han	{*paper*3, *paper*4, *paper*5, *paper*6, *paper*7}	{8, 10, 11, 13, 14, 15, 18, 19, 20, 21}
J. Han, P.S. Yu	{*paper*2}	{4, 6}
J. Han, X. Yan	{*paper*2}	{4, 5}
J. Han, Y. Yu	{*paper*4}	{12, 13}
J. Han, T. Wu	{*paper*3}	{7, 10}
J. Han, C. Li	{*paper*3}	{7, 9}
P.S. Yu, X. Yan	{*paper*2}	{5, 6}
P.S. Yu, Y. Sun	{*paper*5}	{14, 16}
Y. Sun, T. Wu	{*paper*3}	{7, 8}
Y. Sun, Y. Yu	{*paper*4}	{11, 12}
Y. Sun, H. Yin	{*paper*9}	{25, 26}
Y. Sun, C. Li	{*paper*3}	{8, 9}
Y. Sun, X. Yan	{*paper*5}	{14, 17}
C. Li, T. Wu	{*paper*3}	{7, 9}
H. Lu, B. Yang	{*paper*1}	{1, 2}
H. Lu, C.S. Jensen	{*paper*1}	{1, 3}
B. Yang, C.S. Jensen	{*paper*1, *paper*8}	{1, 3, 23, 24}

FIGURE 4.10: Graphs enriched by cubes: an example of co-authorships network

The first graph for analysis is stored in a graph database, we will explain the structure in the Section 5.2.2. The next step is to compute cubes for this graph. We describe in the following section.

4.7.2   Cubes computation

In this section, we present the cube computation algorithm when the measures are numerical or graph-based measures. To compute these cubes, there are different algorithms

according to the type of the measure.

Numerical measure

The cube computation with a numerical measure is given by BUILDCUBESNUMBER (cf. algorithm 2). Let us consider the first graph G with a fact, a measure and a set of dimensions $\mathcal{D}$ which refers to a set of dimension's levels, noted $\mathcal{DL}^{PARAM}$ corresponding to the dimension levels that has been defined by a user. $\mathcal{DL}^{PARAM}$ is defined as the set of dimension levels as parameters. This algorithm returns a graph G' with a specific set of cubes enriching the nodes or edges according to the fact.

As previously, the first graph is co-authorships network. Let us give an example of input as follows.

- F is the co-authorships.

- M is the number of papers.

- $\mathcal{D}$ contains time, venue and session.

- The values of $\mathcal{DL}^{PARAM}_{time}$ are 2009, 2010, 2012 and 2013.

- The values of $\mathcal{DL}^{PARAM}_{venue}$ are CIKM, EDBT, ICDE and KDD.

- The values of $\mathcal{DL}^{PARAM}_{session}$ are Data mining, Data warehouse, Demonstration, Probabilistic and Recommender system

Let us illustrate the algorithm BUILDCUBESNUMBER for the proof concept of SUM, followed by giving an example.

a. Consider the analysis need (determined by a fact), if the fact implies cube on nodes, the algorithm scans through a set of nodes V' (line 1). If fact implies cubes on edges, it scans through a set of edges E'.
   In our example, the fact is co-authorships. In order to know if the cubes are required on nodes or edges, the algorithm gets an answer from the meta data. This network needs cubes only on edges (see FACTS table in Figure 4.4 p.55)

b. For each node or edge, the structure of the cube is built according to a set of dimension values (line 3 for nodes and line 11 for edges).
   In our example, the size of the cube is defined as $4 \times 4 \times 5 = 80$ cells as shown Figure 4.11 p.79.

c. A set of paths is used to calculate the measure value for the cell. The algorithm access the values of paths which are to the measure value in order to count a total value for each cell.

To illustrate the computing of the measure value for each cell, we take the example of the cube between P. Zhao and Y. Sun. In table 4.9 p.75, path 18, 19 belong to this relationship. These paths refer to paper6 (see table 4.7 p.74). The dimension values of this path contain 2009 for time dimension, CIKM for venue dimension and data mining for session dimension. After that a cell at these dimension values is added one value (see Figure 4.12 p.79). If the current value in this cell is 1, the new value will be 2.

Algorithm 2 BUILDCUBESNUMBER

Input: A graph $G = (V,\ E,\ A_V,\ A_E)$ and $G' = (V',\ E')$, a fact F, a measure M, a set of dimensions $\mathcal{D}$

Output: An enriched graph $G' = (V', E', C_{V'}, C_{E'})$ where $C_{V'}$ and $C_{E'}$ are respectively the set of cubes enriching the nodes of V' and the edges of E'.

1: **if** F needs cubes on nodes {according to meta-data} **then**
2:     **for** each v in V' **do**
3:         Build the structure of C_v according to $\mathcal{D}$ corresponding to $\mathcal{DL}^{PARAM}$
4:         **for** each p in P_v **do**
5:             Update the measure value p in the corresponding cell(s) of C_v
6:         **end for**
7:     **end for**
8: **end if**
9: **if** F needs cubes on edges {according to meta-data} **then**
10:     **for** each e in E' **do**
11:         Build the structure of C_e according to $\mathcal{D}$ corresponding to $\mathcal{DL}^{PARAM}$
12:         **for** each p in P_e **do**
13:             Update the measure value p in the corresponding cell(s) of C_e
14:         **end for**
15:     **end for**
16: **end if**

Graph-based measures

If the measure is a graph-based measure, we need three algorithms in order to build the cubes when measures are the degree, the betweenness and the closeness. In social network analysis, graph-based measures are used to understand and explain social phenomena. Look at the co-authorships network in Figure 4.10a p.76, J. Han has 6 edges. In our proposal, the number of Han's edges are provided to a cube according to dimensions in order to answer the questions like what year is the best degree of J. Han? Or

FIGURE 4.11: A structure of a cube

Edges	Set of pid
P. Zhao, J. Han	{8, 10, 11, 13, 14, 15, 18, 19, 20, 21}

Paper6: Peixiang Zhao (University of Illinois), Jiawei Han (University of Illinois) and Yizhou Sun (University of Illinois), 'P-Rank: a comprehensive structural similarity measure over information networks', <u>CIKM</u>, <u>2009</u>, <u>Data Mining</u>.

FIGURE 4.12: Example for computing one measure value

what is the degree evolution between 2010 and 2011?

Algorithm 3 BUILDCUBESDEGREE

Input: Two graph $G = (V, E, A_V, A_E)$ and $G' = (V', E')$, and a set of dimensions $\mathcal{D}$
Output: $C_{V'}^{DC}$ the set of cubes of nodes with the centrality degree as a measure

---

 1: **for** each v in V' **do**
 2:     Build the structure of C_v^{DC} according to $\mathcal{D}$ corresponding to $\mathcal{DL}^{PARAM}$
 3:     **for** each cell c in C_v^{DC} **do**
 4:         $listadd_v = \phi$
 5:         Get adjacent nodes of v in $listadd_v$
 6:         Remove nodes from $listadd_v$ that are not concerned by the values of the complementary dimensions defining the cell
 7:         Put $listadd_v$.size in c
 8:     **end for**
 9: **end for**
10: Return $C_{V'}^{DC}$

Degree measure

The computation of the degree measure is given in BUILDCUBESDEGREE (cf. algorithm 3). Given the first graph G', a graph G and a set of dimensions $\mathcal{D}$, this algorithm returns cubes of nodes with the degree as a measure.

As previously, the first graph is co-authorships network. Let us give an example of input as follows.

- F is the co-authorships.

- M is the number of papers.

- $\mathcal{D}$ contains time, venue and session.

- The values of $\mathcal{DL}_{time}^{PARAM}$ are 2009, 2010, 2012 and 2013.

- The values of $\mathcal{DL}_{venue}^{PARAM}$ are CIKM, EDBT, ICDE and KDD.

- The values of $\mathcal{DL}_{session}^{PARAM}$ are Data mining, Data warehouse, Demonstration, Probabilistic and Recommender system.

Let us illustrate the algorithm BUILDCUBESCD and give its example as the followings.

a. For a node, the structure of a cube is built according to a set of dimension values (line 2).

In our example, the size of a cube is defined as $4 \times 4 \times 5 = 80$ cells as shown in Figure 4.11 p.79.

b. For a cell c of the cube,

(a) the algorithm gets a set of adjacent nodes of v from G' and they are kept into $listadd_v$ (line 4-5).

There are six adjacent nodes of J. Han (see Figure 4.13 p.82). So how many degree of J. Han are in a cell at EDBT 2009 in data mining session?

(b) To get the number of degree for c, nodes which are not concerned by the values of the complementary dimensions defining the cell c will be removed from $listadd_v$. Then the size of $listadd_v$ is added to the cell c (line 6-7).

Look at Figure 4.13 p.82, E1 is an edge between J. Han and Y. Sun. This edge concerns a set of object's value $\{paper3, paper4, paper5, paper6, ppaer7\}$ as shown in table 4.9 p.75. Paper4 and paper7 are published in EDBT 2009 in data mining. A result of this cell is count 1. After that we check every edge to compute the degree of this cell. Three nodes are removed because they do not concern this condition. Finally, J. Han has three degree in a cell at EDBT 2009 in data mining (see Figure 4.13 p.82).

(c) This process is repeated for each cell.

Betweenness measure

The cubes computation of betweenness measure is given in BUILDCUBESBETWEENESS (cf. algorithm 4). Given inputs as the first graph G', a graph G and a set of dimensions $\mathcal{D}$ which refers to a set of dimension's levels $\mathcal{DL}^{PARAM}$ corresponding to the dimensions and is defined by a user, this algorithm returns a set of degree centrality cubes for nodes. As previously, the first graph is co-authorships network. Let us give an example of input as follows.

- F is the co-authorships.

- M is the number of papers.

E1, E4 and E5 have a paper published in
EDBT 2009 in Data mining

How many degree of J. Han are in this cell?

FIGURE 4.13: Example for computing the degree measure of J. Han in the cell corresponding to EDBT 2009 according to data mining

- $\mathcal{D}$ contains time, venue and session.

- The values of $\mathcal{DL}_{time}^{PARAM}$ are 2009, 2010, 2012 and 2013.

- The values of $\mathcal{DL}_{venue}^{PARAM}$ are CIKM, EDBT, ICDE and KDD.

- The values of $\mathcal{DL}_{session}^{PARAM}$ are Data mining, Data warehouse, Demonstration, Probabilistic and Recommender system.

Let us illustrate the algorithm BUILDCUBESBC and give its example in the followings.

Algorithm 4 BUILDCUBESBETWEENESS

Input: Two graph $G = (V, E, A_V, A_E)$ and $G' = (V', E')$, and a set of dimensions $\mathcal{D}$
Output: $C_{V'}^{BC}$ Betweenness centrality cubes of nodes

1: **for** each v in V' **do**
2:     Build the structure C_v^{BC} according to $\mathcal{D}$ corresponding to $\mathcal{DL}^{PARAM}$
3:     **for** each cell c in C_v^{BC} **do**
4:        Get a sub graph G'_c of c according to $\mathcal{DL}^{PARAM}$
5:        Find all shortest paths $SP_{(-v)}$ between every pair of nodes PN in G'_c where both nodes in a pair are not equal to v
6:        Betweenness centrality $BC = 0$
7:        **for** each pn in PN **do**
8:           Extract SP_{pn} from $SP_{(-v)}$ into $listSP_{pn}$
9:           Extract $SP_{pn}(v)$ from $SP_{(-v)}$ into $listSP_{pn}(v)$
10:           $BC = BC + \frac{listSP_{pn}(v)}{listSP_{pn}}$
11:        **end for**
12:        Add BC to c
13:     **end for**
14: **end for**
15: Return $C_{V'}^{BC}$

---

    a. For a node v, the structure of a cube is built from its paths according to a set of dimension values (line 2).

       In our example, the size of a cube is defined as $4 \times 4 \times 5 = 80$ cells as shown in Figure 4.11 p.79.

    b. Then, we traverse each cell c of the cube. For each c,

       b1. we get a sub graph G'_c where $G'_c \subset G'$ and we compute the new shortest path between all pairs of nodes where a starting node and a ending node are not equal to v.

          We give an example to compute this measure of J. Han when a cell is EDBT conference in 2009. Figure 4.14 p.84 shows a graph in EDBT conference in 2009. To compute betweenness score, the first step is to compute the shortest paths between each pair of nodes as shown in Figure 4.14 p.84 when a node is not J. Han (see table 4.10 p.84).

       b2. For each pair of nodes pn,

          i. First is to extract the number of shortest paths SP_{pn} into $listSP_{pn}$ (line 8).

             For example, SP_{pn} of Y. Sun and P.S. Yu has one shortest path (see table 4.10 line 2).

          ii. Then we extract the number of shortest paths passed through v into $listSP_{pn}(v)$ (line 9).

FIGURE 4.14: Co-authorships network for EDBT conference in 2009

TABLE 4.10: The shortest path between all pairs of co-authorships network shown Figure 4.14 where a node is not J. Han

Node i	Node j	The shortest paths
Y. Sun	P.S. Yu	Y.Sun - J. Han - P.S. Yu
Y. Sun	X Yan	Y.Sun - J. Han - X. Yan
P.S.Yu	X. Yan	P.S.Yu - X. Yan

For example, the shortest path between Y. sun and P.S. Yu is $Y.Sun - J.Han - P.S.Yu$. This path pass through J. Han. Thus $listSP_{pn}(v)$ equals to 1.

iii. Finally, the algorithm computes betweenness centrality C_B of c (line 10). The betweenness value is the number of shortest paths SP_{pn} divided by the number of shortest paths passed through v.

In our example, BC of J. Han at $pn_{(Y.Sun, P.S.Yu)}$ is $1/1 = 1$.

iv. This process is repeated for each pn

b3. This process is repeated for each cell c.

Closeness measure

The cubes computation of closeness measure is given in BUILDCUBECLOSENESS (cf. algorithm 5). Given the first graph G', a graph G and a set of dimensions $\mathcal{D}$ which refers to a set of dimension's levels $\mathcal{DL}^{PARAM}$ corresponding to the dimensions and is defined by a user, this algorithm returns a set of degree centrality cubes for nodes.

TABLE 4.11: The shortest path distances between J. Han to others in the co-authorships network shown Figure 4.14

	Y. Sun	P.S. Yu	X. Yan	The shortest paths distance
J. Han	1	1	1	3

As previously, the first graph is co-authorships network. Let us give an example of input as follows.

- F is the co-authorships.

- M is the number of papers.

- $\mathcal{D}$ contains time, venue and session.

- The values of $\mathcal{DL}_{time}^{PARAM}$ are 2009, 2010, 2012 and 2013.

- The values of $\mathcal{DL}_{venue}^{PARAM}$ are CIKM, EDBT, ICDE and KDD.

- The values of $\mathcal{DL}_{session}^{PARAM}$ are Data mining, Data warehouse, Demonstration, Probabilistic and Recommender system.

Let us illustrate the algorithm BUILDCUBESCC and give its example in the followings.

a. For a node v, the structure of a cube is built from its paths according to a set of dimension values (line 2).

   In our example, the size of a cube is defined as $4 \times 4 \times 5 = 80$ cells as shown in Figure 4.11 p.79.

b. Subsequently, we travel each cell c of the cube. For each c, we get a sub graph G'_c where $G'_c \subset G'$ and we compute the distance of the shortest paths from v to others (line 5).

   We give an example to compute this measure of J. Han when a cell is EDBT conference in 2009. Figure 4.14 p.84 shows a graph in EDBT conference in 2009. To compute closeness score, the first step is to compute the shortest paths distance between J. Han to others (see table 4.11 p.85). For example, the distance shortest path from J. Han to Y. Sun is 1.

c. After that the algorithm computes closeness centrality CC of c (line 6-8).

   For example, there are three paths from J. Han to others (see table 4.11 p.85). Thus the summarization of closeness score of J. Han is the sum of the values $1/1 + 1/1 + 1/1 = 3$.

Algorithm 5 BUILDCUBESCLOSENESS

Input: Two graph $G = (V, E, A_V, A_E)$ and $G' = (V', E')$, and a set of dimensions $\mathcal{D}$
Output: $C_{V'}^{CC}$ the set of cubes of nodes with the centrality betweenness as a measure

1: **for** each v in V' **do**
2:    Build the structure C_v^{CC} according to D corresponding to $\mathcal{DL}^{PARAM}$
3:    **for** each cell c in C_v^{CC} **do**
4:       Get a sub graph G'_c of c arcading to $\mathcal{DL}^{PARAM}$
5:       Find the shortest path SP_v from v to others
6:       Closeness centrality $CC = 0$
7:       **for** each sp in SP_v **do**
8:          $CC = CC + \frac{1}{length\ of\ sp}$
9:       **end for**
10:       Add CC to c
11:    **end for**
12: **end for**
13: Return $C_{V'}^{CC}$

4.8 OLAP operations on graphs enriched by cubes

In classical OLAP, operations like roll up, drill down, slice and dice support to explore different multidimensional views and allow interactive querying and analysis of the underlying data. We extend them to analyze graphs enriched by cubes. In our approach, we categorize dimensions into two classes: dimensions for cubes ($\mathcal{D}^{cube}$) and dimensions for a graph ($\mathcal{D}^{graph}$). With two classes of dimensions, we divide OLAP operations into two categories in the followings.

1. **OLAP operations on cubes**

   When a user navigates on a graph, these operations focus on the cubes on nodes and/or edges. These operations do not change the structure of the network. They are close to the informational operations as proposed in Graph OLAP [CYZ+08]. Operations can be divided as follows:

   - Roll up/drill down

     The roll up operation decreases the granularity for the specified dimension $D_d \in \mathcal{D}$ of cubes by grouping measure value into the higher level (where $L_{D_dh}^l = \{L_{D_dh}^1, L_{D_dh}^2, ..., L_{D_dh}^l\}$ and D_d^{type} are defined for the constraints on the content of a graph). The drill down operation increases the granularity by switching to the next lower level of the dimension. Derived granularities are defined as follows:

$$Rollup^{cube}(G', L^l_{D_d h}) := (G', L^{l+1}_{D_d h})$$

$$DrillDown^{cube}(G', L^l_{D_d h}) := (G', L^{l-1}_{D_d h})$$

Figure 4.15 p.88 shows an example of roll up and drill down on a cube of co-authorships network. The cube has number of papers as measures and time, venue and session as dimensions. The cube is aggregated along the session dimension. These operations will not change the structure of a network.

Figure 4.15 p.88 displays an example of a cube for a relationship between J. Han and Y. Sun when the fact is co-authorship and the measure is the number of papers. For instance, there is one paper of their collaboration in EDBT 2009 in data mining.

- Slice and dice

  The slice ($Slice^{cube}$) operation reduces the number of cube dimensions after setting one of the dimensions to specific value. The dice ($Dice^{cube}$) is an operation that reduces the set of data being analyzed by a selection criterion. Derived granularities are defined as follows:

$$Slice^{cube}(G', L^l_{D_d h}) := (G', L^l_{D_d h})$$

$$Dice^{cube}(G', L^l_{D_d h}) := (G', L^l_{D_d h})$$

Figure 4.16 p.89 shows an example of slice on a cube on an edge of J. Han and Y. Sun. The cube is sliced based on the session "data mining". Figure 4.17 p.90 are selected on KDD conference and session is data mining.

Furthermore, there is another possibility of slice and dice operations. Not only the number of cube dimensions are reduced but also they implies a change in a graph. The aim is to analyze a specific graph according to selected values. Derived granularities are defined as follows:

$$Slice^{cube}(G', L^l_{D_d h}) := (G^{slice}, L^l_{D_d h})$$

$$Dice^{cube}(G', L^l_{D_d h}) := (G^{dice}, L^l_{D_d h})$$

FIGURE 4.15: Informational Roll up/Drill down on a cube for the edge between J. Han and Y. Sun

For example, this obtains a new graph as shown in Figure 4.18b. If the cubes in a new graph is changed as shown in Figure 4.18c.

2. **OLAP operations on a graph**

These operations change the structure of a graph and the cubes are recalculated accroding to a new graph. They are close to the topological Graph OLAP [CYZ+08]. In our approach, the structure of a graph can be changed into two ways. First, a

FIGURE 4.16: Slicing based on session = Data mining for a cube on the edge between
J. Han and Y. Sun

type of nodes is changed by another. Second, a type of nodes does not change to
another type but the graph is changed to another graph. We describe them in the
followings:

- Roll up/Drill down

  The roll up ($Rollup^{graph}$) operation generates the network at a higher level.
  The drill down ($Drilldown^{graph}$) operation generates the network at a lower
  level. Derived granularities are defined as follows:

$$Rollup^{graph}(G', D_d) := (G^{rollup}, D_d)$$

$$Drilldown^{graph}(G', D_d) := (G^{drilldown}, D_d)$$

  where G^{rollup} is a higher level network of G' and $G^{drilldown}$ is a lower level
  network of G'.

FIGURE 4.17: Dicing based on session = Data mining and venue = KDD for a cube on the edge between J. Han and Y. Sun

It is more difficult if we take into account the slowly changing dimension over time. A higher level of network cannot be computed from a lower level without accessing raw data. Networked data is often non-summarizable. For example, an author, Y. Sun, published a paper when he was at Northeastern University then he published another paper when he was at university of Illinois. There are two publications of Y. Sun, one for each university. But from the author network, if the user does an OLAP operation like a roll up in order to see the institutions network, these two papers will be counted for both universities, and it is an incorrect answer. The idea of keeping a set of paths into nodes in the previous algorithm allows us to solve this problem.

Figure 4.19b p.93 shows an example of a roll up of the co-authorships network to the institutions network. While all authors of a same institution are merged into one node, edges are created when any two institutions published papers together. In case of many institutions of an author in the same time, the author is counted into all his institutions. After the roll-up, in the more

(a) Co-authorships network with three dimension cubes for each edges

(b) Slicing based on session = 'Data mining'

(c) Slicing based on session = 'Data mining' and
venue = 'KDD'

FIGURE 4.18: Example of slice and dice on co-authorships network

generalized network, new cubes have to be computed. In our example, co-authorships network involves edge cubes, whereas institutions network needs both node and edge cubes.

To build the institutions network, we use both BUILDGRAPH and BUILCUBES algorithms. Before computing a set of nodes (line 2 in algorithm 1), we need to filter paths instead of generating a set of paths (line 1 in algorithm 1). We have to filter paths because all nodes of data set are collected in V', but some nodes may not be in co-authorships network (because some papers are written by only one author). The path filter step is called when the previous network needs edge cubes. Then we compute a new set of nodes from line 2 in algorithm 1. Refer to the example of Figure 4.19, nodes are grouped into institutions. For example, university of Illinois contains path6 and path7 because J. Han and P.S. Yu belong to this university. Figure 4.19 p.93 shows a roll up from co-authorships network to institutions network. Cubes are described both nodes and edges. For instance, Northeastern university is valuated by cube 1 and cube 2 valuates an edge between University of Illinois and Remin University.

- Slice
  The slice operation filters the specified graph $g' \in G'$. It is defined as follows:

$$Slice(G', D_d) := (G'^{slice}, D_d)$$

where G'^{slice} is a sub graph of G'.

Traditional slice operation selects one particular dimension from a given cube and provides a new sub-cube. In our context, slice operation can not be like the classical one. It should be adapted to graphs. The slice operation selects a part of the graph and provides a new sub-graph. For example, if a whole co-authorships network is too big to be comprehensive, the user can focus on a smaller subgraph more interesting to analyze information clearly. Figure 4.20 p.94 shows an example of slice by selecting a sub-graph from the whole co-authorships network.

FIGURE 4.19: Roll up from the co-authorships network to the institutions network

TABLE 4.12: The comparison between the basics of graph OLAP and GreC approach

	Basic of graph OLAP concepts ([CYZ$^+$08])	GreC
Main idea	A cube with graphs.	A graph with cubes.
Fact	Subject of analysis is viewed as a cube	Subject of analysis is viewed as a graph
Measure	Aggregated graph	Numerical measures Graph-based measures
Dimension	Informational and topological	Informational and topological
Aggregation function	Specific aggregation functions	Specific aggregation functions and supporting the slowly changing dimension
Informational roll up OLAP operation	Overlay a set of graphs into a summarized graph	Perform on cubes
Topological roll up OLAP operation	A new cube with aggregated graphs	A new graph with smaller recalculated cubes

FIGURE 4.20: Slice a sub-graph of co-authorships network

4.9 Conclusion

In the Graph OLAP literature, Chen *et al.* [CYZ$^+$08] introduced the principle concept of Graph OLAP. Table 4.12 p.93 shows the comparison between Chen's context and our approach.

Chen *et al.* presented a cubes with graphs. Building on that, a cube contains a set of graphs. On the contrary, GreC presents a subject of analysis as a graph. Each node or edge is weighted by cube. Both these concepts support informational and topological dimension. There are specific aggregation functions. However, GreC supports slowly

changing dimension. There are different ways for roll up on these dimensions. When a roll up is made on an informational dimension in Chen *et al.*, a set of networks is explored to a summarized graph. In our proposal, a roll up is provided on cubes. It has en effect on the structure of graph. In contrast, a roll up on a topological dimension reorganizes the individual networks for a more generlized view for Chen *et al.*. GreC can perform this operation on a graph but not in the individual networks.

This chapter has introduced GreC approach aiming to investigate and navigate the networks by OLAP analysis. Each node and edge are described by a cube. The GreC approach performs multidimensional views of an heterogeneous network rather than a set of graphs. The user can see the global view of a graph. Moreover, our approach keeps the evolution of network as explained in Chapter 2. It allows user to take time dimension on cubes in order to see a history of graph. To achieve these, we first illustrated the process of GreC We described the parts of the process and their respective components. The process consists of three layers: the pre-processing, graphs enriched by cubes computing and the navigation by OLAP analysis. The preprocessing integrates data from different databases and load to a graph database. The part of computation creates a graph and their cubes for a fact and stores them to a graph database. The navigation allows a user to explore graphs and cubes from different views with OLAP operations. The literature reviews as presented in Section 2.2 provided our running example. Therefore, we introduced a graph model for bibliographic data. This model is a multidimensional heterogeneous network that allows to extract different networks and also solves the problems of bibliographic data as explained in Section 2.2.

Consequently, we presented definitions and notations for graphs enriched by cubes by mapping the concepts of fact, dimensions and measure from the multidimensional model. We proposed algorithms which, in addition, solve the slowly changing dimension problem in OLAP analysis in order to compute a graph and cubes. Finally, OLAP operations are adapted to GreC. It takes into account the structure of network in order to do topological OLAP operations and not only classical or informational OLAP operations. We proposed both operations on a graph and cubes. We first take the operations directly on the cubes to see the observe on the graph from a level to another levels. Secondly, operations can take on the graphs. For example, we go from authors network to institutions network. In this case, we recomputed the cubes with the same measures and information in order to have a good data according to a new level. In the future, we can investigate the third operation that an aggregation function is considered. From the authors network, we do operation like a roll up in order to see the institutions network. Instead of proposing the indicator measure, we can apply also an aggregation function. For instance, instead of the degree of an institution, we can have the average degree for the institution according to the degree of the authors. The operations allows to navigate

within the graph. This approach allows to deal with the evolution of network. As we said before, the dynamic networks are usually the different screenshots. In GreC approach, a graph are characterized with multiple cubes with time dimension. This allows to have information about the dynamic of the graph. The next chapter will demonstrate our approach on the real datasets and the performance of algorithms will be studied.

Chapter 5

Implementation and Experiments

5.1 Introduction

In this chapter, we explain the tools for implementing a prototype based on our approach proposed in Chapter 4, called graphs enriched by cubes (GreC). Prototyping helps to prove the interest and the feasibility of our approach in a real-data scenario. As our case study, we chose to build a prototype that implemented the approach to navigate within the world of academic publications.

In this chapter, we first describe data that we use in our experiments and we present storing GreC approach into graph database as a NoSQL database in Section 5.2. In Section 5.3, we give the overview of the architecture and the implementation of our prototype. We show the possibilities for navigation of our prototype. Afterwards, we address the complexity of the algorithms in Section 5.4. We experimentally compare our graph construction with a state of the art algorithm (Beheshti et al.'s approach [SMRBHRM12]). Finally, we summarise the chapter in Section 5.5.

5.2 Data considered and storing graph NoSQL for GreC

5.2.1 Data

In our experiments, we use the bibliographic data, which is extracted from three bibliographic databases. First, DBLP[1] is the well known database, providing bibliographic information on major computer science journals and proceedings. Information is collected into XML files. However, DBLP doesn't provide the institutions of authors But

---

[1] http://dblp.uni-trier.de/

we can see them in ACM. ACM contains a comprehensive bibliographic database focused exclusively on the field of computing. It also provides a richly interlinked set of connections among authors, works, and institutions. Furthermore, we need to get the research area for the venues from Microsoft Research Area. In theses three sources, we keep only three research areas (data mining, databases and information retrieval) and we pick only a few representative conferences for the three areas (PODS, EDBT, KDD, DOLAP, ASONAM, SIGIR and CIKM). At the end, we build a data set which contains 4,727 papers and 8,238 authors since 2009.

5.2.2   Storing graph NoSQL for GreC

In the last decade, the nature of data stored has changed in a number of ways. First of all, the volume of data produced, stored, and processed is growing very quickly. Second, data has been becoming more complex. Finally, data has been becoming increasingly interconnected. These give rise to a new category of database management systems (DBMS) called NoSQL[2]. NoSQL mostly refers to an open-source database and it does not using SQL. NoSQL databases can be categorized in four types: key-value, document, column-family, and graph.

Graph databases are well-suited for for graph applications such as in chemistry, biology, social networks, etc. Board [T.T13] have shown that graph databases present good performances, much better than classical relational databases for representing and querying such large graphs, especially for connected data.

Our proposal of GreC relies on the modelisation of a graph and cubes. With the advantages above gave, graph databases are well-suited for our approach. We illustrate the storage required for GreC approach in the followings.

- **Graph storing**

  To build a graph considered, we need to access the initial graph as presented in Section 4.3. It is an attributed and heterogenous network and It stores a whole data of bibliographic data. To navigate with GreC approach, a graph considered which is enriched by cubes is required, for example, co-authorships network, institutions network, etc. The graphs considered are different from the initial graph, they are stored in a graph database with different structure. In this section, we present the structure of a graph considered.

---

[2]http://nosql-database.org/

FIGURE 5.1: An example of the modelling a graph in NoSQL graph database

In property graph, it introduced the concept of labels. Labels are a way of attaching one or more simple types to nodes and relationships. Figure 5.1 displays an example of a representative of a graph. Type of nodes (Label) is defined according to a graph considered. For example, a graph considered is co-authorships network. Label of nodes is author and label of edges is co-author. A node concerns two properties: a value of node (Name) and a set of paths that is associated to this node (PathID). For example, a node in a red circle names Y. Sun and it has a set of paths {8, 11, 14, 20, 21, 26}. A set of paths contains all paths which belongs to Y. Sun. An edge concerns two properties: a name of edge that refers to two nodes (Name) and a set of paths that is associated with this edge (PathID). For example, E1 is an edge between Y. Sun and J. Han (Name). This edge has co-author as label. It has a set of paths {8, 10, 11, 13, 14, 15, 18, 19, 20, 21}. This set of paths contains all paths which belong to Y. Sun and J. Han. A set of path id is kept in the graph structure because it is useful to build a higher level network as presented in Section 4.7.1 and it is also used to build cubes for nodes and edges.

- **Storing cubes**

Usually, data warehousing allows to create and store cubes, for instance, in relational databases. However, NoSQL databases can help for retrieving relevant information from data using the OLAP paradigms. In this thesis, we use the representation of data cubes through NoSQL databases that is presented in [CL14]. Their model relies on the modelisation of dimensions and facts with typed nodes. Nodes are linked by relations describing:

- links of type hierarchy (:HIER) in the case of dimensions form a lower level to a higher level,

- links of type fact (:FACT) in the case of a link between a dimension and a fact.

Since we build different cubes for nodes and edges in our approach, we define the properties for each node in the structure of modelling cubes. These properties are used to define a unique cube for each node and edge. For example, a cube valuates J. Han node or Y. Sun when a measure is the number of papers. Likewise, if a measure is degree centrality, another cubes of J. Han and Y. Sun are created. Figure 5.2 p.101 shows a representation of data cubes. There are two types of nodes. First, a cell node represents a value in each cell of a cube. With cubes for nodes and edges, we define a cell node with two levels: FactEdge or FactNode. On the one hand, if a cell node has FactNode label, this cube is built for a node. On the other hand, if a cell node has FactEdge, this cube is built for an edge. This node type is described by three properties. They are Measure (the name of measure), For (name of a node or an edge) and Value (a value in a cell). Second, a node is for a level of dimension. This node are described by two properties. They are Type (a dimension name) and Value (a value of this level). A cell node is linked to dimension nodes through relations of type :FACT. A lower level dimension nodes is linked to a higher level dimension node through a relation of type :HIER.

It should be noted that the value in the cell node is built only if it is not empty. Figure 5.3 p.101 shows an example cube for production of J.Han. Two papers are published by J. Han, these papers are described with two dimensions values: 2009 for time dimension and EDBT conference in DB research area for venue dimension. Figure 5.4 p.102 shows an example cube for an edge between J. Han and Y. Sun. There have been one paper issued from a collaboration between J. Han and Y. Sun. This value might be associated to two dimensions: 2009 for time dimension and EDBT conference in DB research area for venue dimension.

FIGURE 5.2: Modelling cubes and dimensions in a NoSQL graph

FIGURE 5.3: An example of cube in a NoSQL graph for J. Han

FIGURE 5.4: An example of cube in a NoSQL graph for an edge between J. Han and Y. Sun

5.3 Implementation of GreC

This section explains about the implementation of prototype. We present tools which are used to our development. After that we show the analysis through our prototype.

5.3.1 Tools

In order to develop our prototype, we use tools as follows:

- A new type of NoSQL database called graph database is used to implement our graph model. We chose Neo4j[3] version 2.0.1 as a graph database because it is an open-source software, it supports the properties of our graph model and it provides a framework for graphs with massive scalability. Neo4j is a graph database running on the Java Virtual Machine (JVM) that processes and stores data natively as property graphs. Nodes, edges, and properties can be created completely arbitrarily. The edges must have a start node and an end node at all times, which is enforced by the database.

---

[3]https://neo4j.com/

- The interface analysis needs meta data which are implemented under Oracle 11g as a relational database.

- Finally, an OLAP interface analysis is developed on NetBeans IDE 7.4 and uses Java version 1.7.0_75. For graph visualisation, we use GraphStream library because it is a library to model and analyze the dynamic graphs and it is an open source library. Although, GraphStream provides the algorithms for network centrality. However, its algorithms do not support a disconnected network. In our case, a network is composed of sub-networks. To overcome this limit, we modified the algorithms to support the disconnected network.

5.3.2   Overview of the architecture

The architecture shown in Figure 5.5 provides an overview of the implementation of GreC. There is an integrated set of three modules for GreC: building GreC data, defining GreC's content interfaces and navigation with GreC. The data comes from various data sources in order to bring it into a form suitable and complete data. The complete basic data is stored in one instance of Neo4j. This instance stores information about bibliographic data. To create the generic modules, a meta data is used to refer the structure required for building GreC data and the interfaces. In this architecture, the meta data is stored under Oracle.

Building GreC data preprocesses the different possible graphs for each facts and the various possible cubes for these graphs. This first builds a graph and cubes. To achieve this, this module needs to access meta data in order to know the structure of a graph and cubes and then get contents from the complete basic data. For each fact, we obtain a graph and all possible cubes for this fact. Consequently, the graph and cubes for each fact are stored using own Neo4j instances. It means that one instance for one graph and one instance for the cubes corresponding to this graph because this can save time to answer user's requirements.

Defining GreC's content interfaces prepares the various interfaces for the user's needs and interactions. This is a generic interface because its structure is developed by using meta data.

Navigation with GreC is to select, visualize and analyse the underlying graphs with cubes. According to user's requirements selected, this module finds the graph and its cubes from Neo4j. The results are visualized on the interface. Two classes of data analysis operations are provided: operations for graph and operations for cubes.

Let us show in the following section in order to see the example of analysis.

FIGURE 5.5: GreC implementation architecture

5.3.3 Examples of analysis

In this section, we present the analysis possibilities of the prototype. Figure 5.6 shows the OLAP interface that contains three parts. The first part provides input components for the user's requirements. A user can define a fact, a measure and a set of dimensions. The second part shows a result of the graph considered with measure and dimensions. The last part is to perform OLAP operations on cubes. We give examples in the followings.

The starting point is that a user defines the input data (see Figure 5.7 p.106). For example, the user selects the co-authorship as a fact (see number 1.1 in Figure 5.7 p.106). After the fact selected, the first filter appears in the interface (see F1 in Figure

FIGURE 5.6: User Interface

5.7 p.106). In our example, the first filter is a list of author's name because a type's node of co-authorship is author. More, measures and dimensions appear according to this fact. Next, the number of papers is selected as a measure (see number 1.2 in Figure 5.7 p.106). Look at number 1.3.1 in Figure 5.7, year is selected as a level of time dimension. As this result, the second filter appears (see F2 in Figure 5.7 p.106). After a user defines a level of venue, two filters are shown: venue's name and area (see F3 and F4 in Figure 5.7 p.106). There are two filters for venue dimension because it has two levels (venue's name and research area). For the filter, a user can define data by limiting the values of the graph. Finally, the user presses the button named ≪compute≫ (see number 1.4 in Figure 5.7 p.106) in order to see the graph and cubes.

As a result, the co-authorships network is shown in the second part of the interface (see Figure 5.8 p.107). There are two sub-parts. First, the graph visualization is the present the network considered (see number 2.1 in Figure 5.8 p.107). The graph may be too big depending on the data considered. Therefore, a user can expend this graph by using two buttons under the graph. The first button is to zoom out the graph (see number 2.1.1 in Figure 5.8). It makes the graph clearly as shown in Figure 5.9 p.108. The second button is to make the graph smaller in order to see overall of the graph (see number 2.1.2 in Figure 5.8). Second, dimensions are shown in the right part (see number 2.2 in Figure

FIGURE 5.7: Example of defining the requirements

5.8). These dimensions are a graph dimension and cubes dimensions. In our example, a graph dimension is author which has two hierarchies: name and institution. This dimension changes the structure of graph where type of nodes and edges are changed. Dimension for cubes are time and venue. These dimensions control the size of the graph. We give more example in the followings.

Figure 5.10 p.109 shows an example to navigate on cubes. In our example, a list of cubes is the relationships between two authors. To navigate within the cubes, a user has to define a level of dimensions (see number 3.1 in Figure 5.10 p.109). For example, we define a level of time dimension. Then a user can sort cubes by clicking on the header of each column (see number 3.2 in Figure 5.10 p.109). In the figure, cubes are ordered by the total number of papers. To see more details of each cube, a user can click at that cube (see number 3.3 in Figure 5.10 p.109). For example, look at this edge between Iadh Ounis and Craig Macdonald; these authors published 29 papers together. These papers are presented according to year of publications (see number 3.4 in Figure 5.10 p.109). If we select a level of time and venue (see big A in Figure 5.11 p.110), it could be interesting to have two ways of visualization. The first way is to focus on time, having the count of papers per year (see big C in Figure 5.11 p.110). Each year has the count of papers by venue. The second way is to focus on the venue, having the count of papers per venue's name (see big D in Figure 5.11 p.110). In this case, each venue has the count

FIGURE 5.8: The graph of co-authorships network

of papers by year. Iadh Ounis and Craig Macdonald wrote 9 papers together in 2013: 3 papers published in SIGIR and 6 papers published in CIKM.

Then, we do a topological roll up on this co-authorships network. Its next higher level is institution, we obtain the result as shown in Figure 5.12 p.111. We define a level of dimensions (see A in Figure 5.13 p.111). A level of time dimension is year and a level of venue dimension is venue's name (*abb*). For example, Microsoft Research Asia published 132 papers in three areas from 2009 to 2013 (see B in Figure 5.13 p.111). These papers are considered like a cube with two dimensions. The first way is to focus on time, having the count of papers per year (see C in Figure 5.13 p.111). The institutions network contains a cube for a node and an edge. Therefore, the visualization of this network is different from the visualization of co-authorships network as shown in Figure 5.10 p.109. Look at big E in Figure 5.13 p.111, Microsoft Research Asia has one collaboration with Microsoft Research Beijing in 2009 by publishing in the CIKM conference. There are 10 papers written by several authors but all belonging to the same institution (Microsoft research Asia) in 2012 (see big F in Figure 5.13 p.111). There are 16 papers of the total

FIGURE 5.9: The graph after using zooming in button

of numbers of papers written by Microsoft Research Asia in 2013 (see big G in Figure 5.13 p.111).

If a user want to see a sub-graph of co-authorships network, a user can do by selecting a specific on dimensions. For example, we want to see a co-authorships network in ASONAM in 2013. Therefore, time and venue dimension are defined (see number 1 in Figure 5.14 p.112). Then, we click ≪explore≫ button (see number 2 in Figure 5.14 p.112). We obtain a sub-graph (see number 3 in Figure 5.14 p.112). Likewise, a set of cubes is changed with respect to this sub-graph.

Furthermore, in the interface, it is an easy way to slice on a graph by using mouse click. Figure 5.15 p.113 shows a way to slice a subgraph from the co-authorships network for the ASONAM conference in 2013. There are several groups of authors. Suppose that we need to consider only the group in the red circle; with a slice operation, the user can select the sub co-authorships network by dragging a mouse. After that a sub-graph selected is changed to green color (see number 1 in Figure 5.15 p.113). To show a sub-graph selected, a user can do a right click on the space of graph visualization and a user

FIGURE 5.10: A set of cubes for co-authorships network (on three areas and all years) according to time dimension

select the first option which is ≪ select graph ≫ (see number 2 in Figure 5.15 p.113). Finally, a user obtains a sub-graph selected (see number 3 in Figure 5.15 p.113).

Consequently, we define degree centrality as a measure. Figure 5.16 p.113 and 5.17 p.114 shows a list of cubes in co-authorships network in three areas since 2009. Jiawei Han has the highest degree. He appears relatively central. We can see his degree centrality according to dimensions selected. First, year is selected as a level of time dimension (see number 1 in Figure 5.16 p.113). We click at a row of Jiawei Han (see number 2 in Figure 5.16 p.113). His degree centrality is shown according to year (see number 3 in Figure 5.16 p.113). For example, he has 47 degree in 2013. This means that he has 47 collaborators in 2013. Second, year is selected as a level of time dimension and venue's name (*abb*) is defined as a level of venue dimension (see number 1 in Figure 5.17 p.114). To see degree of Jiawei Han according to these dimensions, we have to click his row (see number 2 in Figure 5.17 p.114). Look at number 3 in Figure 5.17 p.114, his degree is visualised into two ways. The first one is to focus on time, having the degree per year

FIGURE 5.11: A set of cubes for co-authorships network (on three areas and all years)according to time and venue dimension

and each year has the degree per venue. The second way is to focus on the venue, having the degree per venue's name and each venue has the degree per year.

5.4 Performances study

In this section, we experimentally study the performance of our approach.

5.4.1 Set up

Our experiments are conducted with Java 1.7.0_75 on a laptop with an Intel core i5 2.4 GHz processor with 8 GB of RAM on Mac OS X version 10.9.2 a machine at the user's side. To measure the performance, we use bibliographic data, which is extracted in Section 5.2.1 p.97. We provided such data into four datasets as shown in table 5.1.

FIGURE 5.12: The institutions network (on three areas and all years) with a number of papers

FIGURE 5.13: A set of cubes for institutions network (on three areas and all years)

FIGURE 5.14: Co-authorships network in ASONAM 2013

TABLE 5.1: Four Data Sets

Datasets	Number of Publications	Network of co-authorship		Network of institution	
		Number of nodes	Number of edges	Number of nodes	Number of edges
D1	1,000	2,216	4,322	696	959
D2	2,000	3,790	8,094	1,157	1,820
D3	3,000	5,335	12,150	1,573	2,711
D4	4,000	7,038	16,107	2,051	3,575

These datasets have different size of volume in order to measure the performance of all experiments with respect to data volume.

FIGURE 5.15: Slice a sub-graph on Co-authorships network (on ASONAM in 2013)

FIGURE 5.16: Example of a Cube for the co-authorships network (on three areas and all years) when a measure is degree centrality according to time dimension

FIGURE 5.17: Example of a Cube for the co-authorships network (on three areas and all years) when a measure is degree centrality according to time and venue dimension

5.4.2 Performance results

5.4.2.1 Complexity of building graphs

First, we compare our algorithm for building aggregated graphs with that of Beheshti's approach [SMRBHRM12] because it is the most similar. They proposed a graph data model extending decision support on multidimensional networks and considering both objects and links. They used the concepts of folder and path nodes to support multidimensional and multi- level views and to provide network semantics. To build a network considered, their algorithm starts by scanning all paths to compute nodes. As a result, each node will be stored with its measures. Next, to compute edges, the algorithm first groups nodes according to their measure values. Each measure value contains its name and a set of nodes that associated with it. After that the algorithm travels each measure value to access a set of nodes. An edge is built by grouping any two nodes.

Regarding the complexity for building a graph considered by our approach and by Beheshti's approach, it can be split into two steps: the computation of nodes and the computation of edges.

The complexity of nodes computation for both approaches is $O(|P|)$ because both approaches have to scan all paths to get the different nodes. On the contrary, there is a difference for the edge computation. Our approach uses $O(|V'_f|^2)$, where V'_f is the number of generalized nodes. Whereas Beheshti approach uses $O(|P| + (|V_M| * |v'_f|^2))$, where P is the number of paths, V_M is the number of measure values and $v'_f \in V'_f$ is the number of generalized nodes in each measure value.

With the same complexity of nodes computation, we experimented the running time for the edge computation with two queries as follows:

- Query1 builds the co-authorship networks with the number of papers. This query refers to a structure path $author - write - paper$.

- Query2 creates the institutions network with the number of papers. This query refers to a structure path $author - write - paper - publish - venue$.

To better see the running time of the edge computation, we divide the data set into four data sets as shown in table 5.1 p.112. Figure 5.18 p.116 and Figure 5.19 p.116 compare the running time of query 1 and query 2 in four data sets and for both approaches. Our approach increased the running time when the number of nodes is higher. It scales linearly with respect to the number of nodes (V'_f). In Query 1 as shown in Figure 5.18, Beheshti's approach required less time for dataset2 and more time for the following dataset; our approach requires less time. Although Beheshti's approach has better time for dataset 1 of Query 2, the performance of our approach is better for other datasets. Note that there is 696 nodes of dataset 1 of Query 2. Therefore Beheshti's approach is better performance when number of nodes is less than 1000.

5.4.2.2 Running time of building cubes

We study the performance of algorithms for creating cubes according to different measures. We take an example of the computation of cubes for co-authorships network. Let us suppose that a numeric measure is the number of papers and dimensions are time and venue. In our experiments, we created cubes for nodes. Figure 5.20 p.117 shows the running time for the cubes computation when the number of nodes is higher. For a classical measure and degree centrality, the running time linearly increases with number of cubes. On the contrary, the running time of betweenness centrality and closeness centrality quickly increase with number of cubes. They take much time because they rely on the shortest paths and we have to build a new graph for each cell.

FIGURE 5.18: Running time of edges computation for Query 1

FIGURE 5.19: Running time of edges computation for Query 2

FIGURE 5.20: Running time of cubes computation with different measures

5.4.2.3 Running time of the queries

To demonstrate the usability of GreC approach, we compare our approach with a cube of graphs [CYZ⁺08]. GreC approach is different concept from cube of graphs. GreC presents a subject of analysis as a graph where each node and/or edge is enriched by cube. On the contrary, a cube of graphs is that a cube contains a set of graphs. To validate the performances, we apply two such approaches on four queries according to four datasets as show in Table 5.1. These queries are defined to navigate on a cube because they support both two approaches. In this experiment, we are interested in analyzing co-authorship production according to year and name of venues. Therefore, four queries are defined as follows.

- Query1 includes restrictions on 2010 and EDBT conference in order to see a specific co-authorships network.

- Query2 operates on dimension time to get co-authorships network in 2011.

- Query3 shows a co-authorships network in all venues and all years.

- Query4 studies the relationships between Iadh Ounis and Craig Macdonald in all years and all conferences.

FIGURE 5.21: Running time for Query1

To answer the queries, we assume that all the data structures needed by evaluation algorithms can fit in the working memory. Figure 5.21 to 5.24 plot the running time for four queries., i.e., Query1 (Figure 5.21), Query2 (Figure 5.22 p.119), Query3 (Figure 5.23 p.119) and Query4 (Figure 5.24 p.120).

Figure 5.21 clearly shows that a cube of graphs performs better time because each graph is prepared for each cell of a cube. We can directly access to a graph that we want. Its complexity depends on the number of cells in a cube. Figure 5.22 p.119 to 5.24 p.120 shows running time for Query2, Query3 and Query4 respectively. It is clearly that graphs enriched by cubes performs better time. The best time of graphs enriched by cubes is in Figure 5.23 p.119 because the query wants to find a global view of a graph which is already prepared. A cube of graphs performs the worst time because it needs to summarize graphs for Query2 and Query3. For Query4, it has to travel every graph in order to know an edge between Iadh Ounis and Craig Macdonald. In contrast, GreC has prepared a cube for each edge.

5.5   Conclusion

In this chapter, we provided information about the implementation of GreC approach and the conducted experiments.

FIGURE 5.22: Running time for Query2

FIGURE 5.23: Running time for Query3

FIGURE 5.24: Running time for Query4

In the particular of implementation, our approach needs graphs and cubes. We used Neo4j as a NoSQL database in order to store the graphs and the cubes. Basically, the classical cubes can be created in relational databases. In this thesis, we kept all data in Neo4j instances. To store the cubes to Neo4j, we used the structure of data cubes as presented in [CL14]. The interface and all the process of GreC are implemented in a generic way through the use of meta data because meta data contains information about the structure. Meta data is designed in Oracle as a RDBMS system. Thanks to the illustration of GreC that this approach is particularly interesting in the context of bibliographic data manipulating. We think that this approach is more complementary than the proposed by classical Graph OLAP (cubes of graphs).

Furthermore, we studied the performances. First, our graph construction is compared with Beheshti's approach [SMRBHRM12] because it is the most similar. As the result, our algorithm is better when the number of nodes slightly increases. This is very interesting in the context of an increasing volume of data. However, it scales linearly because it depends on the number of nodes. Second, we studied the performance of cube construction algorithms with respect to different measures. When measures are betweenness centrality and closeness centrality, algorithms took much time if the number of cubes increases because they rely on the shortest paths. Last, the usability of GreC

is achieved with a graph considered and its cubes. The performance of queries depends on number of cubes. With the four kind of queries used in our experiment, GreC did better performance rather than a cube of graph's approach [CYZ$^+$08] for three queries.

Chapter 6

Conclusion and Perspectives

6.1 Conclusion

OLAP which is a well-known technology proposed in the context of data warehousing, is widely used in different application domains. Its ability to provide a user-centred navigation within the data both a multidimensional view of the data and aggregation process makes it very useful. The development of big information networks induces the question of providing consistent way to analyze them. Various challenges are open such as the specific data type, the volume, the dynamic and etc. In this thesis, we addressed the issue of OLAP analysis with graphs to enable multidimensional analysis on informational networks. This contribution has been proposed through Graph OLAP approache that proposed to build cubes of graphs. We proposed a complementary way that is building graphs enriched by cubes (GreC). To achieve this, we detailed a completed process, a framework and algorithms that allowed to make this idea real. We have implemented the graphs enriched by cubes approach in a research scenario, specifically in the context of bibliographic data even if our approach could be applied to other dimensions. It allowed the user to navigate within the network considered where nodes and edges were valued by cubes that provided rich useful information (among them: temporal information). All the process took into account the user's analysis needs. This approach solved in a certain way the slowly changing dimension problem.

This research has achieved the following contributions:

- **Graphs enriched by cubes approach**
  Our proposal is a new approach for online analytical processing on graphs, which consists in enriching graphs with cubes. The nodes and/or edges of the network considered are described by cubes. Two types of measures were introduced to our

approach. First, they are graphs enriched by cubes with classical measures. More-over, we proposed to add centrality measures (degree, betweenness, and closeness) in order to explore the role of nodes in each network. The presented framework is well adapted for analysing bibliographic data. Thus a graph model for this kind of data was required since the graph model is the basic step for our approach. This is our second contribution.

- **The graph model for bibliographic data**

  In bibliographic data, there are objects (authors, papers and etc.) which come from multiple bibliographic databases. In our approach we need to build several different networks such as co-authorships, institutions of authors and etc. In real-ity, their contents may have two problems: an entity concerns many different value in the same properties and a property values is changing over time. To handle all problems, we used the properties of graph theory and we presented a graph model for bibliographic networks. Our graph model is an attributed and heterogeneous network. That is used after like a generator of networks, thanks to same meta data.

- **Definitions and notations for graphs enriched by cubes**

  We developed a formal model for GreC by proposing definitions and notations that extend the concepts used for OLAP and Graph OLAP. First, a fact was a subject of analysis which was viewed as a network in order to face different data and to depict the interconnection among data. The fact defined some characteristics of a network where nodes and edges were valued. Second, the concept of measure was presented. There were classical ones (numerical) but also graph-based ones. Third, dimensions were organized according to different levels representing hierarchy. In our approach, dimensions were divided into two types: dimensions for cubes and dimensions for a graph. Finally, a concept of hierarchy was presented because it organized the dimension attributes and this implied different operations on graph or cubes for the analysis.

- **Reinforcing OLAP operations to graphs enriched by cubes**

  The use of OLAP operations to do multidimensional analysis of information net-works can potentially provide answers to the users like scientists i.e., for questions such as *who is the leader in KDD conference?*. To provide a rich analysis frame-work, we considered different types of operations. First OLAP on the cubes of the graph allows navigate within the information data describing nodes and/or edges. In this case the structure of the graph does not change. This refers to the informational operations in "classical" Graph OLAP. Furthermore, we decide to enriched this vision to the graph itself. In this case, OLAP operations could

take into account the structure of the network in order to go from one view of this network to another one. To do this, we had to recompute cubes for this new network view. This refers to topological OLAP operations as proposed in Graph OLAP. In both cases, we had to redefine the operations in the context of GreC.

- **The algorithms building graphs enriched by cubes**
  To implement the GreC approach, we needed to imagine two types of algorithms. The first one was to build the graph for analysis. The second one dealt with computing the cubes. There were four different algorithms according to the type of the measure considered: cube computation with the numerical measured and cube computation with three centrality measures (degree, betweenness and closeness). These algorithms were used at the beginning to build the first graph enriched by cubes, but also during the analysis navigational process when it was required.

  In Chapter 5, we also tested the performance of our graph computation algorithm with Beheshti's approach [SMRBHRM12] because it was closed to our. The result showed that our algorithm had a better performance. In addition, we studied the performances of cube computation. The results from the experiment showed that betweenness centrality and closeness centrality took much time if the number of cubes increases because they relied on the shortest paths. Finally, we also tested our approach by conducting usability queries. We compared our approach with the first Graph OLAP approach[CYZ$^+$08]. It was clear that GreC approach performed better time when a query wanted to find a global view of a graph which is already prepared. Grec approach also had a better time when a query needed to summarize graphs, although our algorithm traveled to cubes. However, Graph OLAP had better time when a user wanted to see a specific graph because Graph OLAP prepared a graph for each granularity. On the contrary, our approach needed to travel all cubes to get the answer.

- **Implementation of the graphs enriched by cubes approach based on the prototype**
  We developed a prototype in order to show how our approach can be used for helping users to analyze information networks with OLAP technology. The objective of the development was to ensure that our approach can help a user to analyze and navigate a network that is enriched by cubes. The prototype worked as a user-centered prototype in which users can analyze and navigate with in the network considered with cubes. As a case study, we focused on academic publications; specifically the publications that were provided by DBLP, ACM and Microsoft Research Area. This implementation constituted a prof of concept for GreC approach.

6.2 Perspectives

In this thesis, we show that GreC constitutes a good navigational approach to analyze information networks. This work opens various issues that could be addressed to improve GreC. Here are a couple of possible research directions:

- The first short-term perspectives concerns the analysis possibilities by extending the envisaged measures. In this work, measures were only numeric measures (the number of papers) and centrality measures (degree, betweenness and closeness) in order to explore the role of nodes in each network. Centrality is important because it indicates which node occupies critical positions in the network. However, the importance of the centrality could be adapted to explore the role of edges. For example, each edge in the co-authorships network can be associated with an edge betweenness centrality value. An edge which has a high edge betweenness centrality score represents a bridge-like connector between two parts of a sub co-authorships network, and its removal may affect the communication between many pairs of nodes through the shortest paths between them. It means that the removal of that edge will result in a partition of the co-authorships network into two densely connected sub-networks. In the context of bibliographic data, this refers to people that are at the junction between two communities on sub-communities, depending the data considered. Therefore, we plan to apply the centrality to explore the role of edges in each network. We also plan to add other graph-based measures, i.e., diameter, similarity and etc.

- In addition, one interesting and challenging extension is to consider text mining tools in order to enrich the model and the network by more attributes. Text mining tools can be useful for information extraction. So we will combine Graph OLAP and Text OLAP in order to handle all networked data. In this thesis, we consider two types of measures: numerical measures and graph-based measures. However, the importance of incorporating text-rich document data can be analyze through textual measures in graphs enriched by cubes. For example, one measure can provide the analysis of the keywords of a specific author or the analysis of the keywords of a relationships among two authors, in order to get an overview of keywords contents or the evolution of keywords [RTTZ08].

- Regarding operations for graph in this thesis, the evolution of the network could be analysed by taking into account the time dimension in the cubes that are valuated nodes and/or edges. In Grec, for now, we have just considered unary operations where one graph is the input, such as drill down, roll up, etc. Another issue that could be interesting to explore the dynamic in the graph would be to consider

binary operations. In this case, the idea would consist on focusing on two graphs as inputs. The two graphs could be a snapshot at two different moments. And the operations could be the difference, the intersection between these two graphs. This perspective induces considering how these operations could be applied in GreC approach and answering to this question: what does it mean to envisage the difference or the intersection of two graphs enriched by cubes, and particularly in terms of the cubes that are valuating nodes and/or edges? This perspective would bring new analysis possibilities in terms of graph evolution.

- Moreover, we could think about representing co-authorships links in a more complex way to match with the reality of links. Indeed, in GreC the edges concern at each time two authors. However, in a copublication, authors are more than two and this information is difficult to rebuild with our approach. Thus we loose a part of information. To overhead this limit, the detection of "cliques" in the graph could be an issue. In this case, that poses the problem of multiplying the cubes for difficult graphs of authors. Another issue to be investigated would be to use of hypergraph. This perspective induces explaining how GreC could be adapted to this new representation.

- With GreC, we were focusing on proposing a new way to visualize a graph with a new kind of cubes. The possible extensions consist also on focusing on the user to provide him/her a useful help for explaining the data. A first issue could be the community detection. This is of extraordinary importance in the domain of information networks analysis to understand the organization , the structure. This implies to reconsider community detection according to the type information we propose through the cubes in GreC. In this context, that means considering different of communities depending on co-authorships, but also on topics, publication behaviour, and also considering the temporal dimension.

- In order to help the user, since the data could be huge, we could improve the step for filtering the data to "select the good graph". It consists in developing more filtering possibilities but we can also think about a recommendation process [NRTT13] as in the domain of information retrieval. Recommendation has been also the subject of different works in the domain of OLAP analysis. Thus this perspective could focus on how to do recommendation in the context of GreC: for graph, but also for cubes, in terms of user profile or in terms of a collaborative use of the GreC platform.

- A parallel perspective addressed to the user would be to focus on the visualization of the data. It is interesting to improve the visualization to 3D interfaces for OLAP. Lafon *et al.* studied how 3D and VR can be used in OLAP interfaces [LBGV13]. A

new 3D interface for OLAP with several extensions like the possibility contains two measures (DB-Miner), images, and 3D widgets that represent the OLAP operators to be triggered. This perspective is very important in the context of the growth of the volume of data. Particularity, if we want to consider a real-time approach that would induce to focus on performances, considering the scalabilities of our approach.

- And last, but not least, this is to organize a concrete user evaluation. We can target two types. First of all, the users of the domain covered by the considered data. For now, we applied our approach on data for computer science. So the users could be the researcher in computer science. It will be interesting to consider after other domains. The second types of users could be sociologists that are specialized in sociology of science. They are interesting in tool for Scientometrics and GreC could constitute in the analysis of how the research is done.

Bibliography

[AGZ15]   Sabri Skhiri Alejandro Vaisman Amine Ghrab, Oscar Romero and Este-
             ban Zimányi. A framework for building OLAP cubes on graphs. In *19th
             East-European Conference on Advances in Databases and Information
             Systems (ADBIS'15)*, pages 92–105, 2015.

[BBH$^+$08] Akanksha Baid, Andrey Balmin, Heasoo Hwang, Erik Nijkamp, Jun
             Rao, Berthold Reinwald, Alkis Simitsis, Yannis Sismanis, and Frank
             van Ham. Dbpubs: multidimensional exploration of database publica-
             tions. *Proceedings of the VLDB Endowment*, 1(2):1456–1459, 2008.

[Bra01]   Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal
             of mathematical sociology*, 25(2):163–177, 2001.

[Cab11]   Guillaume Cabanac. Accuracy of inter-researcher similarity measures
             based on topical and social clues. *Scientometrics*, 87(3):597–620, 2011.

[CD97]   Surajit Chaudhuri and Umeshwar Dayal. An overview of data ware-
             housing and OLAP technology. *ACM Sigmod record*, 26(1):65–74, 1997.

[CGP09]   Michele Coscia, Fosca Giannotti, and Ruggero Pensa. Social network
             analysis as knowledge discovery process: a case study on digital bibli-
             ography. In *International Conference on Advances in Social Networks
             Analysis and Mining (ASONAM'09)*, pages 279–283, 2009.

[CL14]   Arnaud Castelltort and Anne Laurent. NoSQL graph-based OLAP
             analysis. In *6th International Conference on Knowledge Discovery and
             Information Retrieval (IC3K'14)*, pages 217–224, 2014.

[CYZ$^+$08] Chen Chen, Xifeng Yan, Feida Zhu, Jiawei Han, and Philip S Yu. Graph
             OLAP: Towards online analytical processing on graphs. In *8th IEEE
             International Conference on Data Mining (ICDM'08)*, pages 103–112,
             2008.

[Def96]   OLAP and OLAP server definitions: OLAP glossary, 1996. Accessed: 2016-20-03.

[Die00]   Reinhard Diestel. *Graph theory (Graduate Texts in Mathematics).* Springer-Verlag Berlin and Heidelberg GmbH, 2000.

[DKL08]   Hongbo Deng, Irwin King, and Michael R Lyu. Formal models for expert finding on dblp bibliography data. In *8th IEEE International Conference on Data Mining (ICDM'08)*, pages 163–172, 2008.

[FPSS⁺96]   Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. Knowledge discovery and data mining: Towards a unifying framework. In *2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, volume 96, pages 82–88, 1996.

[Fre78]   Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[GAHS11]   Manish Gupta, Charu C Aggarwal, Jiawei Han, and Yizhou Sun. Evolutionary clustering and analysis of bibliographic networks. In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM'11)*, pages 63–70, 2011.

[GT11]   Tsvetanka Georgieva-Trifonova. Warehousing and OLAP analysis of bibliographic data. *Intelligent Information Management*, 3:190–197, 2011.

[Hul23]   Edward Wyndham Hulme. *Statistical bibliography in relation to the growth of modern civilization.* Printed for the author by Butler & Tanner; Grafton & Co., 1923.

[HV03]   Emil Hudomalj and Gaj Vidmar. OLAP and bibliographic databases. *Scientometrics*, 58(3):609–622, 2003.

[HYQQ09]   Zhixing Huang, Yan Yan, Yuhui Qiu, and Shuqiong Qiao. Exploring emergent semantic communities from dblp bibliography database. In *International Conference on Advances in Social Network Analysis and Mining (ASONAM'09)*, pages 219–224, 2009.

[Inm92]   William H Inmon. *Build the Data Warehouse.* 1992.

[JFL13]   Wararat Jakawat, Cécile Favre, and Sabine Loudcher. OLAP on information networks: A new framework for dealing with bibliographic data. In *1st International Workshop on Social Business Intelligence*

(SoBI'13), collocated with the East-European Conference on Advances in Databases and Information Systems (ADBIS'13), pages 361–370, 2013.

[JFL15]   Wararat Jakawat, Cécile Favre, and Sabine Loudcher. OLAP cube-based graph approach for bibliographic data. In *42nd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'16), Student Research Forum*, 2015.

[JFL16]   Wararat Jakawat, Cécile Favre, and Sabine Loudcher. Graphs enriched by cubes for OLAP on bibliographic networks. *International Journal of Business Intelligence and Data Mining (IJBIDM'16)*, 11(1):85–107, 2016.

[JHC⁺10]   Xin Jin, Jiawei Han, Liangliang Cao, Jiebo Luo, Bolin Ding, and Cindy Xide Lin. Visual cube and on-line analytical processing of images. In *19th ACM international conference on Information and knowledge management (CIKM'10)*, pages 849–858, 2010.

[KA14]   Mehmet Kaya and Reda Alhajj. Development of multidimensional academic information networks with a novel data cube based modeling method. *Information Sciences*, 265:211–224, 2014.

[KH11]   Benedikt Kämpgen and Andreas Harth. Transforming statistical linked data for use in OLAP systems. In *7th International Conference on Semantic Systems (I-SEMANTICS'11)*, pages 33–40, 2011.

[KLR⁺04]   Stefan Klink, Michael Ley, Emma Rabbidge, Patrick Reuther, Bernd Walter, and Alexander Weber. Visualising and mining digital bibliographic data. In *GI Jahrestagung (2)*, pages 193–197, 2004.

[KR02]   Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley and Sons, Inc., 2002.

[KR11]   Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley and Sons, Inc., 2011.

[KRW⁺06]   Stefan Klink, Patrick Reuther, Alexander Weber, Bernd Walter, and Michael Ley. Analysing social networks within bibliographical data. In *17th International Conference on Database and Expert Systems Applications (DEXA'06)*, pages 234–243, 2006.

[LBGV13]   Sébastien Lafon, Fatma Bouali, Christiane Guinot, and Gilles Venturini. On studying a 3D user interface for OLAP. *Data Mining and Knowledge Discovery*, 27(1):4–21, 2013.

[LFJ13]   Sabine Loudcher, Cécile Favre, and Wararat Jakawat. Que peut apporter l'OLAP à l'analyse de réseaux d'informations bibliographiques ? In *4ème conférence sur les modèles et l'analyse des réseaux : approches mathématiques et informatiques (MARAMI'13)*, 2013.

[LJMF15]   Sabine Loudcher, Wararat Jakawat, Edmundo Pavel Soriano Morales, and Cécile Favre. Combining OLAP and information networks for bibliographic data analysis: a survey. *Scientometrics*, 103(2):471–487, 2015.

[MK08]   Konstantinos Morfonios and Georgia Koutrika. OLAP cubes for social searches: Standing on the shoulders of giants? In *11th International Workshop on the Web and Databases (WebDB'08)*, 2008.

[ML10]   Fabrice Muhlenbach and Stéphane Lallich. Discovering research communities by clustering bibliographical data. In *IEEE WIC ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'10)*, volume 1, pages 500–507, 2010.

[MMC13]   Wassim Derguech Muntazir Mehdi, Ratnesh Sahay and Edward Curry. On-the-fly generation of multidimensional data cubes for web of things or semantic sensor networks. In *17th International Database Engineering & Applications Symposium (IDEAS'13)*, pages 28–37, 2013.

[New03]   Mark EJ Newman. The structure and function of complex networks. volume 45, pages 167–256, 2003.

[NRTT13]   Elsa Negre, Franck Ravat, Olivier Teste, and Ronan Tournier. Cold-start recommender system problem within a multidimensional data warehouse. In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS'13)*, pages 1–8, 2013.

[OAS10]   Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, 2010.

[PK10]   Manh Cuong Pham and Ralf Klamma. The structure of the computer science knowledge network. In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM'10)*, pages 17–24, 2010.

[Pri69]   Alan Pritchard. *Statistical bibliography; an interim bibliography.* 1969.

[QZY⁺11]   Qiang Qu, Feida Zhu, Xifeng Yan, Jiawei Han, S Yu Philip, and Hongyan Li. Efficient topological OLAP on information networks. In *16th International Conference on Database Systems for Advanced Applications (DASFAA'11)*, pages 389–403, 2011.

[RTTZ08]   Franck Ravat, Olivier Teste, Ronan Tournier, and Gilles Zurfluh. Top_keyword: an aggregation function for textual document olap. In *International Conference on Data Warehousing and Knowledge Discovery (DaWaK'08)*, pages 55–64, 2008.

[SHZ⁺09]   Yizhou Sun, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, and Tianyi Wu. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT'09)*, pages 565–576, 2009.

[SMRBHRM12] Beheshti Seyed-Mehdi-Reza, Benatallah Boualem, Motahari-Nezhad Hamid Reza, and Allahbakhsh Mohammad. A framework and a language for on-line analytical processing on graphs. In *13th International Conference on Web Information Systems Engineering (WISE'12)*, pages 213–227, 2012.

[SQU10]   Kazuhiro Seki, Huawei Qin, and Kuniaki Uehara. Impact and prospect of social bookmarks for bibliographic information retrieval. In *10th annual joint conference on Digital libraries (JCDL'10)*, pages 357–360, 2010.

[Tho02]   Erik Thomsen. *OLAP Solutions: Building Multidimensional Information Systems.* Wiley. com, 2002.

[THP08]   Yuanyuan Tian, Richard A Hankins, and Jignesh M Patel. Efficient aggregation for graph summarization. In *ACM SIGMOD international conference on Management of data (SIGMOD'08)*, pages 567–580, 2008.

[T.T13]   Board T.T.A. Technology radar, 2013.

[Van97]   Raan A Van. Scientometrics: State-of-the-art. *Scientometrics*, 38(1):205–218, 1997.

[VT11]   Iraklis Varlamis and George Tsatsaronis. Visualizing bibliographic databases as graphs and mining potential research synergies. In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM'11)*, pages 53–60, 2011.

[WER15]   Ahmed Waqas, Zimányi Esteban, and Wrembel Robert. Temporal datawarehouses: Logical models and querying. volume XIe journées francophones sur les Entrepôts de Données et l'Analyse en Ligne, RNTI-B-11 (EDA'15), pages 33–48, 2015.

[WFW⁺14]   Zhengkui Wang, Qi Fan, Huiju Wang, Kian-Lee Tan, Deepak Agrawal, and Amr El Abbadi. Pagrol: Parallel graph OLAP over large-scale attributed graphs. In *IEEE 30th International Conference on Data Engineering (ICDE'14)*, pages 496–507, 2014.

[WSR⁺12]   Lili Wu, Roshan Sumbaly, Chris Riccomini, Gordon Koo, Hyung Jin Kim, Jay Kreps, and Sam Shah. Avatara: OLAP for webscale analytics products. *Proceedings of the VLDB Endowment*, 5(12):1874–1877, 2012.

[XY12]   Peixiang Zhao Jiawei Han Xiao Yu, Yizhou Sun. Query-driven discovery of semantically similar substructures in heterogeneous networks. In *18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1500–1503, 2012.

[YG14]   Dan Yin and Hong Gao. Iceberg cube query on heterogeneous information networks. In *9th International Conference on Wireless Algorithms, Systems, and Applications (WASA'14)*, pages 740–749. 2014.

[YWZ12]   Mu Yin, Bin Wu, and Zengfeng Zeng. HMGraph OLAP: a novel framework for multi-dimensional heterogeneous network analysis. In *15th International Workshop on Data warehousing and OLAP (DOLAP'12)*, pages 137–144, 2012.

[ZCG09]   Osmar R Zaïane, Jiyang Chen, and Randy Goebel. Mining research communities in bibliographical data. In *Advances in Web Mining and Web Usage Analysis*, pages 59–76. 2009.

[ZHPL12]   Jing Zhang, Xiaoguang Hong, Zhaohui Peng, and Qingzhong Li. Nestedcube: Towards online analytical processing on information-enhanced multidimensional network. In *Web-Age Information Management*, pages 128–139. 2012.

[ZLXH11]   Peixiang Zhao, Xiaolei Li, Dong Xin, and Jiawei Han. Graph cube: on warehousing and OLAP multidimensional networks. In *ACM SIGMOD International Conference on Management of Data (SIGMOD'11)*, pages 853–864, 2011.