Chapter 1

AN INTRODUCTION TO GRAPH DATA

Charu C. Aggarwal
IBM T. J. Watson Research Center
Hawthorne, NY 10532
charu@us.ibm.com

Haixun Wang
Microsoft Research Asia
Beijing, China 100190
haixunw@microsoft.com

Abstract Graph mining and management has become an important topic of research recently because of numerous applications to a wide variety of data mining problems in computational biology, chemical data analysis, drug discovery and communication networking. Traditional data mining and management algorithms such as clustering, classification, frequent pattern mining and indexing have now been extended to the graph scenario. This book contains a number of chapters which are carefully chosen in order to discuss the broad research issues in graph management and mining. In addition, a number of important applications of graph mining are also covered in the book. The purpose of this chapter is to provide an overview of the different kinds of graph processing and mining techniques, and the coverage of these topics in this book.

Keywords: Graph Mining, Graph Management

1. Introduction

This chapter will provide an introduction of the topic of graph management and mining, and its relationship to the different chapters in the book. The problem of graph management finds numerous applications in a wide variety of application domains such as chemical data analysis, computational biology,

C.C. Aggarwal and H. Wang (eds.), *Managing and Mining Graph Data*,
Advances in Database Systems 40, DOI 10.1007/978-1-4419-6045-0_1,

social networking, web link analysis, and computer networks. Different applications result in different kinds of graphs, and the corresponding challenges are also quite different. For example, chemical data graphs are relatively small but the labels on different nodes (which are drawn from a limited set of elements) may be repeated many times in a single molecule (graph). This results in issues involving graph isomorphism in mining and management applications. On the other hand, in many large scale domains [12, 21, 22] such as the web, computer networks, and social networks, the node labels (eg. URLs) are distinct, but there are a very large number of them. Such graphs are also challenging because the degree distributions of these graphs are highly skewed [10], and this leads to difficulty in characterizing such graphs succinctly. The massive size of computer network graphs is a considerable challenge for mining algorithms. In some cases, the graphs may be *dynamic* and *time-evolving*. This means that the structure of the graph may change rapidly over time. In such cases, the *temporal aspect* of network analysis is extremely interesting.

A closely related field is that of XML data. Complex and semi-structured data is often represented in the form of XML documents because of its natural expressive power. XML data is naturally represented in graphical form, in which the attributes along with their values are expressed as nodes, and the relationships among them are expressed as edges. The expressive power of graphs and XML data comes at a cost, since it is much more difficult to design mining and management operations for structured data. The design of management and mining algorithms for XML data also helps in the design of methods for graph data, since the two fields are closely related to one another.

The book is designed to survey different aspects of graph mining and management, and provide a compendium for other researchers in the field. The broad thrust of this book is divided into three areas:

- **Managing Graph Data:** Since graphs form a complex and expressive data type, we need methods for representing graphs in databases, manipulating and querying them. We study the problem of designing query languages for graphs [14], and show how to use such languages in order to retrieve structures from the underlying graphs [26]. We also explore the design of indexing and retrieval structures for graph data. In addition, a number of specialized queries such as matching, keyword search and reachability queries [4–7, 24] are studied in the book. We will see that the design of the index is much more sensitive to the underlying application in the case of structured data than in the case of multi-dimensional data. The problem of managing graph data is related to the widely studied field of managing XML data. Where possible, we will draw on the field of XML data, and show how some of these techniques may be used in order to manage graphs in different domains. We will also present some of the recently designed techniques for graph data.

- **Mining Graph Data:** As in the case of other data types such as multidimensional or text data, we can design mining problems for graph data. This includes techniques such as frequent pattern mining, clustering and classification [1, 11, 16, 18, 23, 25, 26, 28]. We note that these methods are much more challenging in the graph domain, because the structural nature of the data makes the intermediate representation and interpretability of the mining results much more challenging. This is of course related to the cost of the greater expressive power associated with graphs.

- **Graph Applications:** Many of the techniques discussed above are for the case of generic graphs under a number of specific assumptions. However, graph domains are extremely diverse, and this may result in a large number of differences in the algorithms which are designed for such cases. For example, the algorithms which are designed for the web or social networks need to be constructed for graphs with very large size, but with distinct node labels. On the other hand, the algorithms which are designed for chemical data need to take into account repetitions in node labels. Similarly many graphs may have additional information associated with nodes and edges. Such variations make different applications much more challenging. Furthermore, the generic techniques discussed above may need to be applied differently for different application domains. Therefore, we have included different chapters to handle these different cases. We will study applications relating to the web, social networks, software bug localization, chemical and biological data.

One of the goals of this book is to provide the reader with a comprehensive compendium of material in the area of graph management and mining. The book provides a number of introductory chapters in the beginning, and then discusses a variety of graph mining algorithms in more detail.

2. Graph Management and Mining Applications

In this section, we will discuss the organization of the different chapters in the book. We will discuss the different applications, and the chapters in which they are discussed. In the first two chapters, we provide an introduction to the area of graph mining an a general survey. This chapter (Chapter 1) provides a brief introduction to the area of graph mining and the organization of this book. Chapter 2 is a general survey which discusses the key problems and algorithms in each area. The aim of the first two chapters is to provide the reader with a general overview of the field without getting into too much detail. Subsequent chapters expand on the various areas of graph mining. We discuss these below.

Natural Properties of Real Graphs and Generators. In order to understand the various management and mining techniques discussed in the book, it is important to get a feel of what real graphs look like in practice. Graphs which arise in many large scale applications such as the web and social networks satisfy many properties such as the power law distribution [10], sparsity, and small diameters [19]. These properties play a key role in the design of effective management and mining algorithms for graphs. Therefore, we discuss these properties at an early stage of the book. Furthermore, the evolution of dynamic graphs such as social networks shows a number of interesting properties such as densification, and shrinking diameters [19]. Furthermore, since the study of graph mining algorithms requires the design of effective graph generators, it is useful to study methods for constructing realistic generators [3]. Clearly, the understanding that we obtain from the study of the natural properties of graphs in real domains can be leveraged in order to design models for effective generators. Chapter 3 studies the laws of real large-scale network graphs and a number of techniques for synthetic generation of graphs.

Query Languages and Indexing for Graphs. In order to effectively handle graph management applications, we need query languages which allow expressivity for management and manipulation of structural data. Furthermore, such query languages also need to be efficiently implementable. In chapter 4, a variety of query languages for graphs are presented.

A second issue is that of *efficient access* of the underlying information in order to resolve the queries. Therefore, it is useful to study the design of index structures for graphs. General techniques for efficiently indexing graphs are presented in chapter 5. While chapter 5 is focussed exclusively on the graph domain, we note that many of the indexing techniques for the XML domain can also be useful for graphs. Chapter 2 explores some of the connections between XML indexing and graph indexing. In addition to general queries such as similarity search, which are typically designed on *multi-graph data sets*, graph structures are naturally suited to the design of a number of different other kinds of queries for a single massive graph. In such cases, we may have a single graph, but we wish to determine important intra-node characteristics in the graph. Such queries often arise in the context of social networks and the web. Examples of such queries include reachability and distance based queries [2, 4–7, 24]. Such queries are based on the *intra-node distance behavior* in a large network structure, and are often extremely challenging because the underlying graph may be disk-resident. In chapter 6, the literature for reachability query processing is reviewed.

Graph Matching. Graph matching is a critical problem which arises in the context of a number of different kinds of applications such as schema match-

ing, graph embedding and other business applications [9]. In the problem of graph matching, we have a pair of graphs, and we attempt to determine a mapping of nodes between the two graphs such that edge and/or label correspondence is preserved. Graph matching has traditionally been studied in the theoretical literature in the context of the *graph isomorphism* problem. However, in the context of practical applications, precise matching between two graphs may not be possible. Furthermore, many practical variations of the problem allow for partial knowledge about the matching between different nodes. Therefore, we also need to study inexact matching techniques which allow edits on the nodes and edges during the matching process. Chapter 7 studies exact and inexact matching techniques for graphs.

Keyword Search in Graphs. In the problem of keyword search, we would like to determine small groups of link-connected nodes which are related to a particular keyword [15]. For example, a web graph or a social network may be considered a massive graph [21, 22], in which each node may contain a large amount of text data. Even though keyword search is defined with respect to the text inside the nodes, we note that the linkage structure also plays an important role in determining the appropriate set of nodes. The information in the text and linkage structure re-enforce each other, and this leads to higher quality results. Keyword search provides a simple but user-friendly interface for information retrieval on the web. It also proves to be an effective method for searching data of complex structures. Since many real life data sets are structured as tables, trees and graphs, keyword search over such data has become increasingly important and has attracted much research interest in both the database and the IR communities. It is important to design keyword search techniques which maintain query semantics, ranking accuracy, and query efficiency. Chapter 8 provides an exhaustive survey of keyword search techniques in graphs.

Graph Clustering and Dense Subgraph Extraction. The problem of graph clustering arises in two different contexts:

- In the first case, we wish to determine dense node clusters in a *single large graph*. This problem arises in the context of a number of applications such as graph-partitioning and the minimum cut problem. The determination of dense regions in the graph is a critical problem from the perspective of a number of different applications in social networks, web graph clustering and summarization. In particular, most forms of graph summarization require the determination of dense regions in the underlying graphs. A number of techniques [11, 12, 23] have been designed in the literature for dense graph clustering.

- In the second case, we have multiple graphs, each of which may possibly be of modest size. In this case, we wish to cluster graphs as objects. The distance between graphs is defined based on a structural similarity function such as the edit distance. Alternatively, it may be based on other aggregate characteristics such as the membership of frequent patterns in graphs. Such techniques are particularly useful for graphs in the XML domain, which are naturally expressed as objects. A method for XML data clustering is discussed in [1].

In chapter 9, both the above methods for clustering graphs have been studied. A particularly closely related problem to clustering is of dense subgraph extraction. Whereas the problem of clustering is traditionally defined as a *strict partitioning of the nodes*, the problem of dense subgraph extraction is a relaxed variation of this problem in which dense subgraphs may have overlaps. Furthermore, many nodes may not be included in any dense component. The dense subgraph problem is often studied in the context of frequent pattern mining of multi-graph data sets. Other variations include the issue of repeated presence of subgraphs in a single graph or in multiple graphs. These problems are studied in chapter 10. The topics discussed in chapters 9 and 10 are closely related, and provide a good overview of the area.

Graph Classification. As in the case of graph clustering, the problem of graph classification arises in two different contexts. The first context is that of vertex classification in which we attempt to label the nodes of a single graph based on training data. Such problems are based on that of determining *desired properties of nodes* with the use of training data. Examples of such methods may be found in [16, 18]. The second context is one in which we attempt to label entire graphs as objects. The first case arise in the context of massive graphs such as social networks, whereas the second case arises in many different contexts such as chemical or biological compound classification, or XML data [28]. Chapter 11 studies a number of different algorithms for graph classification.

Frequent Pattern Mining in Graphs. The problem of frequent pattern mining is much more challenging in the case of graphs than in the case of standard transaction data. This is because not all frequent patterns are equally relevant in the case of graphs. In particular, patterns which are highly connected are much more relevant. As in the case of transactional data, a number of different measures may be defined in order to determine which graphs are the most significant. In the case of graphs, the structural constraints make the problem even more interesting. As in the case of the transactional data, many variations of graph pattern mining such as that of determining closed patterns or significant patterns [25, 26], provide different kinds of insights to the field.

The frequent pattern mining problem is particularly important for the graph domain, because the end-results of the algorithms provide an overview of the important structures in the underlying data set, which may be used for other applications such as indexing [27]. Chapter 12 provides an exhaustive survey of the different algorithms for frequent pattern mining in graphs.

Streaming Algorithms for Graphs. Many graph applications such as those in telecommunications and social networks create continuous streams of edges. Such applications create unique challenges, because the entire graph cannot be held either in main memory or on disk. This creates tremendous constraints for the underlying algorithms, since the standard one-pass constraint of streaming algorithms applies to this case. Furthermore, it is extremely difficult to explore the structural characteristics of the underlying graph, because a global view of the graph is hard to construct in the streaming case. Chapter 13 discusses a number of streaming applications for such edge streams. The chapter discusses how graph streams can be summarized in an application-specific way, so that important structural characteristics of the graph can be explored.

Privacy-Preserving Data Mining of Graphs. In many applications such as social networks, it is critical to preserve the privacy of the nodes in the underlying network. Simple de-identification of the nodes during the release of a network structure is not sufficient, because an adversary may use background information about known nodes in order to re-identify the other nodes [17]. Graph privacy is especially challenging, because background information about many structural characteristics such as the node degrees or structural distances can be used in order to mount identity-attacks on the nodes [17, 13]. A number of techniques have recently been proposed in the literature, which use node addition, deletion, or swapping in order to hide such structural characteristics for privacy-preservation purposes [20, 29]. The key in these techniques is to hide identifying structural characteristics, without losing the overall structural utility of the graph. Chapter 14 discusses the challenges of graph privacy, and a variety of algorithms which can be used for private processing of such graphs.

Web Applications. Since the web is naturally structured as a graph, numerous such applications require graph mining and management algorithms. A classic example is the case of social networks in which the linkage structure is defined in the form of a graph. Typical social networking applications require the determination of interesting regions in the graph such as the dense communities. Community detection is a direct application of the problem of clustering, since it requires the determination of dense regions of the underlying graph. Many other applications such as blog analysis, web graph analysis,

and page rank analysis for search require the use of graph mining algorithms. Chapter 15 provides a comprehensive overview of graph mining techniques for web applications. Since social networking is an important area, which cannot be easily covered within the context of the single chapter on web applications, we devote a special chapter on social networking. Graph mining applications for social networking are discussed in chapter 16.

Software Bug Localization. Software programs can be represented as graphs, in which the control flow is represented in the form of a graph. In many cases, the software bugs arise as a result of "typical" distortions in the underlying control flow. Such distortions can also be understood in the context of the graphical structure which represents this control flow. Therefore, software bug localization is a natural application is graph mining algorithms in which the structure of the control flow graph is studied in order to determine and isolate bugs in the underlying program. Chapter 17 provides a comprehensive survey of techniques for software bug localization.

Chemical and Biological Data. Chemical compounds can be represented as graph structures in which the atoms represent the nodes, and the bonds represents the links. If desired, a higher level of representation can be used in which sub-units of the molecules represent the nodes and the bonds between them represent the links. For example, in the case of biological data, the amino-acids are represented as nodes, and the bonds between them are the links. Chemical and biological data are inherently different in the sense that the graphs corresponding to biological data are much larger and require different techniques which are more suitable to massive graphs. Therefore, we have devoted two separate chapters to the topic. In chapter 18, methods for mining biological compounds are presented. Techniques for mining chemical compounds are presented in chapter 19.

3. Summary

This book provides an introduction to the problem of managing and mining graph data. We will present the key techniques for both management and mining of graph data sets. We will show that these techniques can be very useful in a wide variety of applications such as the web, social networks, biological data, chemical data and software bug localization. . The book also presents some of the latest trends for mining massive graphs and their applicability across different domains. A number of trends in graph mining are fertile areas of research for future applications:

- Scalability is the new frontier in graph mining applications. Applications such as the web and social networks are defined on *massive graphs*

in which it is impossible to explicitly store the underlying edges in main memory and sometimes even on disk. While graph-theoretic algorithms have been studied extensively in the literature, these techniques implicitly assume that the graphs can be held in main memory and are therefore not very useful for the case of disk-resident. This is because disk access may result in random access to the underlying edges which is extremely inefficient in practice. This also leads to a lack of scalability of the underlying algorithms.

- Many communication and social networking applications create large sets of edges which arrive continuously over time. Such dynamic applications require quick responses to queries to a number of traditional applications such as the shortest path problem or connectivity queries. Such queries are an enormous challenge, since it is impossible to prestore the massive volume of the data for future analysis. Therefore, effective techniques need to be designed to compress and store the graphical structures for future analysis.

- A number of recent data mining applications and advances such as privacy-preserving data mining and uncertain data need to be studied in the context of the graph domain. For example, social networks are structured as graphs, and privacy applications are particularly important in this context. Such applications are also very challenging since they are defined on a massive domain of nodes.

This book studies a number of important problems in the graph domain in the context of important graph and networking applications. We also introduce some of the recent trends for massive graph mining applications.

References

[1] C. Aggarwal, N. Ta, J. Feng, J. Wang, M. J. Zaki. XProj: A Framework for Projected Structural Clustering of XML Documents, *KDD Conference*, 2007.

[2] R. Agrawal, A. Borgida, H.V. Jagadish. Efficient Maintenance of transitive relationships in large data and knowledge bases, *ACM SIGMOD Conference*, 1989.

[3] D. Chakrabarti, Y. Zhan, C. Faloutsos R-MAT: A Recursive Model for Graph Mining. *SDM Conference*, 2004.

[4] J. Cheng, J. Xu Yu, X. Lin, H. Wang, and P. S. Yu, Fast Computing Reachability Labelings for Large Graphs with High Compression Rate, *EDBT Conference*, 2008.

[5] J. Cheng, J. Xu Yu, X. Lin, H. Wang, and P. S. Yu, Fast Computation of Reachability Labelings in Large Graphs, *EDBT Conference*, 2006.

[6] E. Cohen. Size-estimation framework with applications to transitive closure and reachability, *Journal of Computer and System Sciences*, v.55 n.3, p.441-453, Dec. 1997.

[7] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick, Reachability and distance queries via 2-hop labels, *ACM Symposium on Discrete Algorithms*, 2002.

[8] D. Cook, L. Holder, Mining Graph Data, *John Wiley & Sons Inc*, 2007.

[9] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.

[10] M. Faloutsos, P. Faloutsos, C. Faloutsos, On Power Law Relationships of the Internet Topology. *SIGCOMM Conference*, 1999.

[11] G. Flake, R. Tarjan, M. Tsioutsiouliklis. Graph Clustering and Minimum Cut Trees, *Internet Mathematics*, 1(4), 385–408, 2003.

[12] D. Gibson, R. Kumar, A. Tomkins, Discovering Large Dense Subgraphs in Massive Graphs, *VLDB Conference*, 2005.

[13] M. Hay, G. Miklau, D. Jensen, D. Towsley, P. Weis. Resisting Structural Re-identification in Social Networks, *VLDB Conference*, 2008.

[14] H. He, A. K. Singh. Graphs-at-a-time: Query Language and Access Methods for Graph Databases. In *Proc. of SIGMOD '08*, pages 405–418, Vancouver, Canada, 2008.

[15] H. He, H. Wang, J. Yang, P. S. Yu. BLINKS: Ranked keyword searches on graphs. In *SIGMOD*, 2007.

[16] H. Kashima, K. Tsuda, A. Inokuchi. Marginalized Kernels between Labeled Graphs, *ICML*, 2003.

[17] L. Backstrom, C. Dwork, J. Kleinberg. Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography. *WWW Conference*, 2007.

[18] T. Kudo, E. Maeda, Y. Matsumoto. An Application of Boosting to Graph Classification, *NIPS Conf.* 2004.

[19] J. Leskovec, J. Kleinberg, C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters. *ACM Transactions on Knowledge Discovery from Data (ACM TKDD)*, 1(1), 2007.

[20] K. Liu and E. Terzi. *Towards identity anonymization on graphs.* ACM SIGMOD Conference 2008.

[21] R. Kumar, P Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, E. Upfal. The Web as a Graph. *ACM PODS Conference*, 2000.

[22] S. Raghavan, H. Garcia-Molina. Representing web graphs. *ICDE Conference*, pages 405-416, 2003.

[23] M. Rattigan, M. Maier, D. Jensen: Graph Clustering with Network Sructure Indices. *ICML*, 2007.

[24] H. Wang, H. He, J. Yang, J. Xu-Yu, P. Yu. Dual Labeling: Answering Graph Reachability Queries in Constant Time. *ICDE Conference*, 2006.

[25] X. Yan, J. Han. CloseGraph: Mining Closed Frequent Graph Patterns, *ACM KDD Conference*, 2003.

[26] X. Yan, H. Cheng, J. Han, and P. S. Yu, Mining Significant Graph Patterns by Scalable Leap Search, *SIGMOD Conference*, 2008.

[27] X. Yan, P. S. Yu, and J. Han, Graph Indexing: A Frequent Structure-based Approach, *SIGMOD Conference*, 2004.

[28] M. J. Zaki, C. C. Aggarwal. XRules: An Effective Structural Classifier for XML Data, *KDD Conference*, 2003.

[29] B. Zhou, J. Pei. Preserving Privacy in Social Networks Against Neighborhood Attacks. *ICDE Conference*, pp. 506-515, 2008.