1

INTRODUCTION

LAWRENCE B. HOLDER AND DIANE J. COOK

School of Electrical Engineering and Computer Science
Washington State University, Pullman, Washington

The ability to mine data to extract useful knowledge has become one of the most important challenges in government, industry, and scientific communities. Much success has been achieved when the data to be mined represents a set of independent entities and their attributes, for example, customer transactions. However, in most domains, there is interesting knowledge to be mined from the relationships between entities. This relational knowledge may take many forms from periodic patterns of transactions to complicated structural patterns of interrelated transactions. Extracting such knowledge requires the data to be represented in a form that not only captures the relational information but supports efficient and effective mining of this data and comprehensibility of the resulting knowledge. Relational databases and first-order logic are two popular representations for relational data, but neither has sufficiently supported the data mining process.

The graph representation, that is, a collection of nodes and links between nodes, does support all aspects of the relational data mining process. As one of the most general forms of data representation, the graph easily represents entities, their attributes, and their relationships to other entities. Section 1.2 describes several diverse domains and how graphs can be used to represent the domain. Because one entity can be arbitrarily related to other entities, relational databases and logic have difficulty organizing the data to support efficient traversal of the relational links.

Mining Graph Data, Edited by Diane J. Cook and Lawrence B. Holder

Graph representations typically store each entity's relations with the entity. Finally, relational database and logic representations do not support direct visualization of data and knowledge. In fact, relational information stored in this way is typically converted to a graph form for visualization. Using a graph for representing the data and the mined knowledge supports direct visualization and increased comprehensibility of the knowledge. Therefore, mining graph data is one of the most promising approaches to extracting knowledge from relational data.

These factors have not gone unnoticed in the data mining research community. Over the past few years research on mining graph data has steadily increased. A brief survey of the major data mining conferences, such as the Conference on Knowledge Discovery and Data Mining (KDD), the SIAM Conference on Data Mining, and the IEEE Conference on Data Mining, has shown that the number of papers related to mining graph data has grown from 0 in the late 1990s to 40 in 2005. In addition, several annual workshops have been organized around this theme, including the KDD workshop on Link Analysis and Group Detection, the KDD workshop on Multi-Relational Data Mining, and the European Workshop on Mining Graphs, Trees and Sequences. This increasing focus has clearly indicated the importance of research on mining graph data.

Given the importance of the problem and the increased research activity in the field, a collection of representative work on mining graph data was needed to provide a single reference to this work and some organization and cross fertilization to the various topics within the field. In the remainder of this introduction we first provide some terminology from the field of mining graph data. We then discuss some of the representational issues by looking at actual representations in several important domains. Finally, we provide an overview of the remaining chapters in the book.

1.1 TERMINOLOGY

Data mining is the extraction of novel and useful knowledge from data. A *graph* is a set of nodes and links (or vertices and edges), where the nodes and/or links can have arbitrary labels, and the links can be directed or undirected (implying an ordered or unordered relation). Therefore, *mining graph data*, sometimes called *graph-based data mining*, is the extraction of novel and useful knowledge from a graph representation of data. In general, the data can take many forms from a single, time-varying real number to a complex interconnection of entities and relationships. While graphs can represent this entire spectrum of data, they are typically used only when relationships are crucial to the domain. The most natural form of knowledge that can be extracted from graphs is also a graph. Therefore, the *knowledge*, sometimes referred to as *patterns*, mined from the data are typically expressed as graphs, which may be subgraphs of the graphical data, or more abstract expressions of the trends reflected in the data. Chapter 2 provides more precise definitions of graphs and the typical operations performed by graph-based data mining algorithms.

While data mining has become somewhat synonymous with finding frequent patterns in transactional data, the more general term of *knowledge discovery* encompasses this and other tasks as well. *Discovery* or *unsupervised learning* includes not only the task of finding patterns in a set of transactions but also the task of finding possibly overlapping patterns in one large graph. Discovery also encompasses the task of *clustering*, which attempts to describe all the data by identifying categories or clusters sharing common patterns of attributes and relationships. Clustering can also extract relationships between clusters, resulting in a hierarchical or taxonomic organization over the clusters found in the data. In contrast, *supervised learning* is the task of extracting patterns that distinguish one set of graphs from another. These sets are typically called the positive examples and negative examples. These sets of examples can contain several graph transactions or one large graph. The objective is to find a graphical pattern that appears often in the positive examples but not in the negative examples. Such a pattern can be used to predict the class (positive or negative) of new examples. The last graph mining task is the visualization of the discovered knowledge. *Graph visualization* is the rendering of the nodes, links, and labels of a graph in a way that promotes easier understanding by humans of the concepts represented by the graph.

All of the above graph mining tasks are described within the chapters of this book, and we provide an overview of the chapters in Section 1.3. However, an additional motivation for the work in this book is the important application domains and how their data is represented as a graph to support mining. In the next section we describe three domains whose data is naturally represented as a graph and in which graph mining has been successful.

1.2 GRAPH DATABASES

Three domains that epitomize the tasks of mining graph data are the Internet Movie Database, the Mutagenesis dataset, and the World Wide Web. We describe several graph representations for the data in these domains and survey work on mining graph data in these domains. These databases may also serve as a benchmark set of problems for comparing and contrasting different graph-based data mining methods.

1.2.1 The Internet Movie Database

The Internet Movie Database (IMDb) [41] maintains a large database of movie and television information. The information is freely available through online queries, and the database can also be downloaded for in-depth analysis. This database emerged from newsgroups in the early 1990s, such as rec.arts.movies, and has now become a commercial entity that serves approximately 65 million accesses each month.

Currently, the IMDb has information on 468,305 titles and 1,868,610 people in the business. The database includes filmographies for actors, directors, writers, composers, producers, and editors as well as movie information such as titles, release

dates, production companies and countries, plot summaries, reviews and ratings, alternative names, genres, and awards.

Given such filmography information, a number of mining tasks can be performed. Some of these mining tasks exploit the unstructured components of the data. For example, Chaovalit and Zhou [9] use text-based reviews to distinguish well-accepted from poorly accepted movies. Additional information can be used to provide recommendations to individuals of movies they will likely enjoy. Melville et al. [33] combine IMDb movie information (title, director, cast, genre, plot summary, keywords, user comments, reviews, awards) with movie ratings from EachMovie [14] to predict items that will be of interest to individuals. Vozalis and Margaritis [42] combine movie information, ratings from the GroupLens dataset [37], and demographic information to perform a similar recommendation task. In both of these cases, movie and user information is treated as a set of independent, unstructured attributes.

By representing movie information as a graph, relationships between movies, people, and attributes can be captured and included in the analysis. Figure 1.1(a) shows one possible representation of information related to a single movie. This hub topology represents each movie as a vertex, with links to attributes describing the movie. Similar graphs could be constructed for each person as well. With this representation, one task we can perform is to answer the following question:

What commonalities can we find among movies in the database?

Using a frequent subgraph discovery algorithm, subgraphs that appear in a large fraction of the movie graphs can be reported. These algorithms may report discoveries such as movies receiving awards often come from the same small set of studios [as shown in Fig. 1.1(b)] or certain director/composer pairs work together frequently [as shown in Fig. 1.1(c)].

By connecting people, movies, and other objects that have relationships to each other, a single connected graph can be constructed. For example, Figure 1.2 shows how different movies may have actors, directors, and studios in common. Similarly,

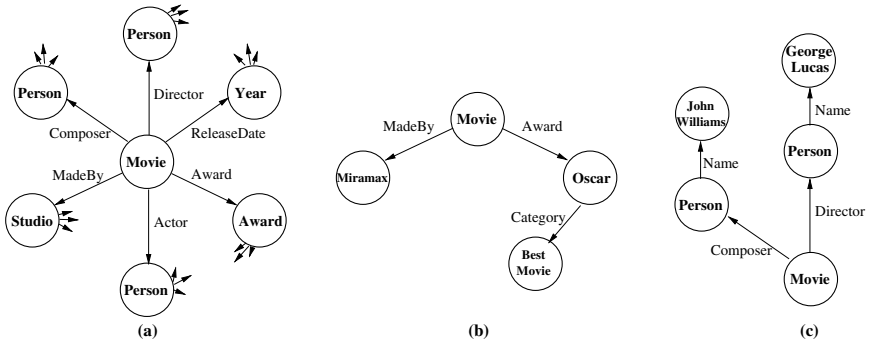

Figure 1.1. (a) Possible graph representation for information related to a single movie. (b) One possible frequent subgraph. (c) Another possible frequent subgraph.

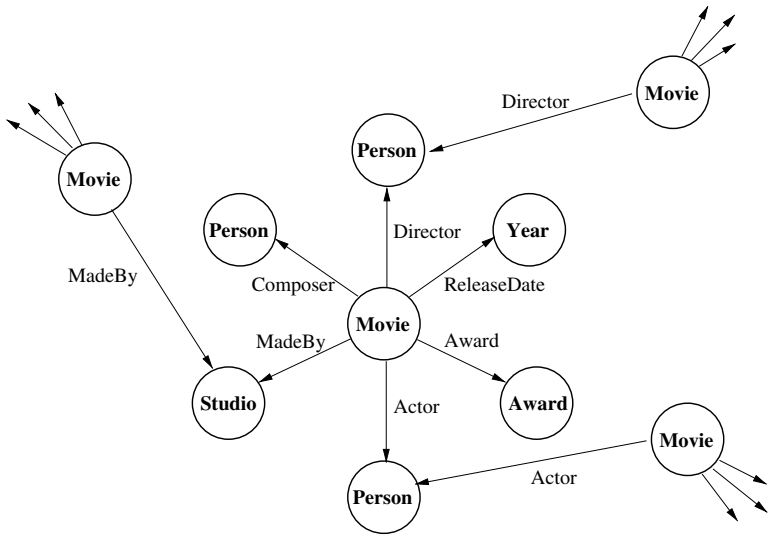

Figure 1.2. Second graph representation in which relationships between data points are represented using labeled edges.

different actors may appear in the same movie, forming a relationship between these people. Analysis of this connected graph may answer questions such as:

What common relationships can we find between objects in the database?

For the movie graph, a discovery algorithm may find a recurring pattern that movies made by the same studio frequently also have the same producer. Jensen and Neville [21] mention another type of discovery that can be made from a connected graph. In this case, an emerging film star may be characterized in the graph by a sequence of successful movies in which he or she stars and by winning one or more awards.

Other analyses can be made regarding the topology of such graphs. For example, Ravasz and Barabasi [36] analyzed a graph constructed by linking actors appearing in the same movie and found that the graph has a distinct hierarchical topology. Movie graphs can also be used to perform classification. As an example, Jensen and Neville [21] use information in a movie graph as shown in Figure 1.2 to predict whether a movie will make more than $2 million in its opening weekend. In a separate study, they use structure around nominated and nonnominated movies to predict which new movies will be nominated for awards [32].

These examples show that patterns can be learned from structural information that is explicitly provided. However, missing structure can also be inferred from this data. Getoor et al.'s [16] approach learns a graph linking actors and movies using IMDb information together with demographic information based on actor ZIP codes. Mining algorithms can be used to infer missing links in the movie graph. For example, given information about a collection of people who starred together

in a movie, link completion [17, 28] can be used to determine who the remaining individuals are who starred in the same movie. Such link completion algorithms can also be used to determine when one movie is a remake of another [21].

1.2.2 Mutagenesis Data

The Mutagenesis dataset is a chemical compound dataset where the task is explicitly defined as:

> Given the molecular structure of a compound, identify the compound as mutagenic or not mutagenic.

The Mutagenesis dataset was collected to identify mutagenic activity in chemical data [10]. Mutation is a structural alteration in DNA (deoxyribonucleic acid). Occasionally, a mutation improves an organism's chance of surviving or has no observable effect. In most cases, however, such DNA changes harm human health. A high correlation is also observed between mutagenicity and carcinogenicity. Some chemical compounds are known to cause frequent mutations. Mutagenicity cannot be practically determined for every compound using biological experiments, so accurate evaluation of mutagenic activity from chemical structure is very desirable.

Structure–activity relationships (SARs) relate biological activity with molecular structure. The importance of SARs to drug design is well established. The Mutagenesis problem focuses on obtaining SARs that describe the mutagenicity of nitroaromatic compounds or organic compounds composed of NO or NO_2 groups attached to rings of carbon atoms. Analyzing relationships between mutagenic activity and molecular structure is of great interest because highly mutagenic nitroaromatic compounds are carcinogenic.

The Mutagenesis dataset collected by Debnath et al. [10] consists of the molecular structure of 230 compounds, such as the one shown in Figure 1.3. Of these compounds, 138 are labeled as mutagenic and 92 are labeled nonmutagenic. Each compound is described by its constituent atoms, bonds, atom and bond types, and partial charges on atoms. In addition, the hydrophobicity of the compound (log P), the energy level of the compound's lowest unoccupied molecular orbital (LUMO), a Boolean attribute identifying compounds with three or more benzyl rings (I1), and a

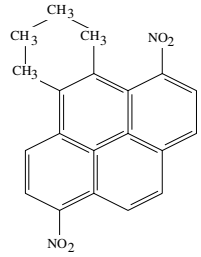

Figure 1.3. 1,6,-Dinitro-9,10,11,12-tetrahydrobenzo[*e*]pyrene.

Boolean attribute identifying compounds that are acenthryles (Ia). The mutagenicity of the compounds has been determined using the Ames test [1]. While alternative datasets are being considered by the community as challenges for structural data mining [29], the Mutagenesis dataset provides both a representative case for graph representations of chemical data and an ongoing challenge for researchers in the data mining community.

Some work has focused on analyzing these chemical compounds using global, nonstructural descriptors such as molecular weight, ionization potential, and various physiocochemical properties [2, 19]. More recently, researchers have used inductive logic programming (ILP) techniques to encode additional relational information about the compounds and to infuse the discovery process with background knowledge and high-level chemical concepts such as the definitions of methyl groups and nitro groups [23, 40]. In fact, Srinivasan and King in a separate study [38] show that traditional classification approaches such as linear regression improve dramatically in classification accuracy when enhanced with structural descriptors identified by ILP techniques.

Inductive logic programming methods face some limitations because of the explicit encoding of structural information and the prohibitive size of the search space [27]. Graphs provide a natural representation for the structural information contained in chemical compounds. A common mining task for the Mutagenesis data, therefore, is to represent each compound as a separate graph and look for frequent substructures in these graphs. Analysis of these graphs may answer the following question:

> What commonalities exist in mutagenic or non-mutagenic compounds that will help us to understand the data?

This question has been addressed by researchers with notable success [5, 20]. A related question has been addressed as well [11, 22]:

> What commonalities exist in mutagenic or nonmutagenic compounds that will help us to learn concepts to distinguish the two classes?

An interesting twist on this task has been offered by Deshpande et al. [12], who do not use the substructure discovery algorithm to perform classification but instead use frequency of discovered subgraphs in the compounds to form feature vectors that are then fed to a Support Vector Machine classifier.

Many of the graph templates used for the Mutagenesis and other chemical structure datasets employ a similar representation. Vertices correspond to atoms and edges represent bonds. The vertex label is the atom type and the edge label is the bond type. Alternatively, separate vertices can be used to represent attributes of the atoms and the bonds, as shown in Figure 1.4. In this case information about the atom's chemical element, charge, and type (whether it is part of an aromatic ring) is given along with attributes of the bond such as type (single, double, triple) and relative three-dimensional (3D) orientation. Compound attributes including log

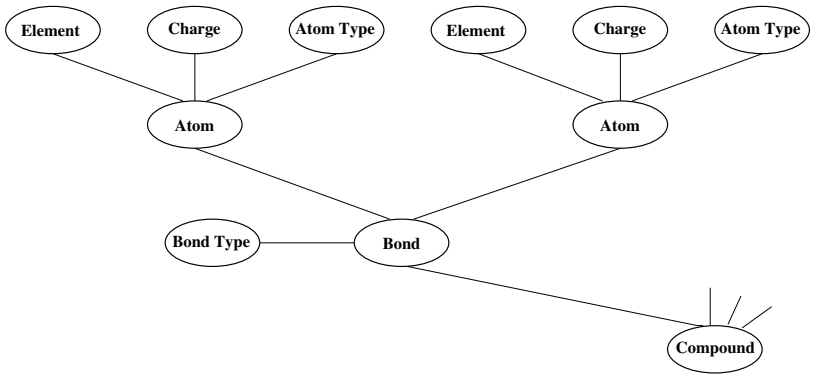

Figure 1.4. Graph representation for a chemical compound.

P, LUMO, I1, and Ia can be attached to the vertex representing the entire chemical compound.

When performing a more in-depth analysis of the data, researchers often augment the graph representation with additional features. The types of features that are added are reflective of the type of discoveries that are desired. Ketkar et al. [22], for example, add inequality relationships between atom charge values with the goal of identifying value ranges in the concept description. In Chapter 14 of this book, Okada provides many more descriptive features that can be considered.

1.2.3 Web Data

The World Wide Web is a valuable information resource that is complex, dynamically evolving, and rich in structure. Mining the Web is a research area that is almost as old as the Web itself. Although Etzioni coined the term "Web mining" [15] to refer to extracting information from Web documents and services, the types of information that can be extracted are so varied that this has been refined to three classes of mining tasks: Web content mining, Web structure mining, and Web usage mining [26].

Web content mining algorithms attempt to answer the following question:

What patterns can I find in the content of Web pages?

The most common approach to answering this question is to perform mining of the content that is found within each page on the Web. This content typically consists of text occasionally supplemented with HTML tags [8, 43]. Using text mining techniques, the discovered patterns facilitate classification of Web pages and Web querying [4, 34, 44].

When structure is added to Web data in the form of hyperlinks, analysts can then perform Web structure mining. In a Web graph, vertices represent Web pages and edges represent links between the Web pages. The vertices can optionally be

labeled by the domain name (as Kuramochi and Karypis describe in Chapter 6), and edges are typically unlabeled or labeled with a uniform tag. Additional vertices can be attached to the Web page nodes that are labeled with keywords or other textual information found in the Web page content. Figure 1.5 shows a sample graph of this type for a collection of three Web pages. With the inclusion of this hypertext information, Web page classification can be performed based on structure alone (Gartner and co-workers describe this in Chapter 11 of this book) or together with Web content information [18]. Algorithms that analyze Web pages based on more than textual content can also potentially glean more complex patterns, such as "there is a prevalence of data mining web pages that have links to job pages and links to publication pages."

Other researchers focus on insights that can be drawn using structural information alone. Chakrabarti and Faloutsos. (Chapter 4) and others [7, 24] have studied the unique attributes of graphs created from Web hyperlink information. Such hyperlink graphs can also be used to answer the following question:

What patterns can I find in the Web structure?

In Chapter 6, Kuramochi and Karypis discover frequent subgraphs in these topology graphs. In Chapter 16, Tomkins and Kumar show how new or emerging communities of Web pages can be identified from such a graph. Analysis of this graph leads to identification of topic hubs and authorities [25]. Authorities in this case are highly ranked pages on a given topic, and hubs represent overview sites with links to strong authority pages. The PageRank program [6] precomputes page ranks based on the number of links to the page from other sites together with the probability that a Web surfer will visit the page directly, without going through intermediary sites. In Chapter 12, Shimbo and Ito also demonstrate how the relatedness of Web pages can be determined from link structure information. Finally,

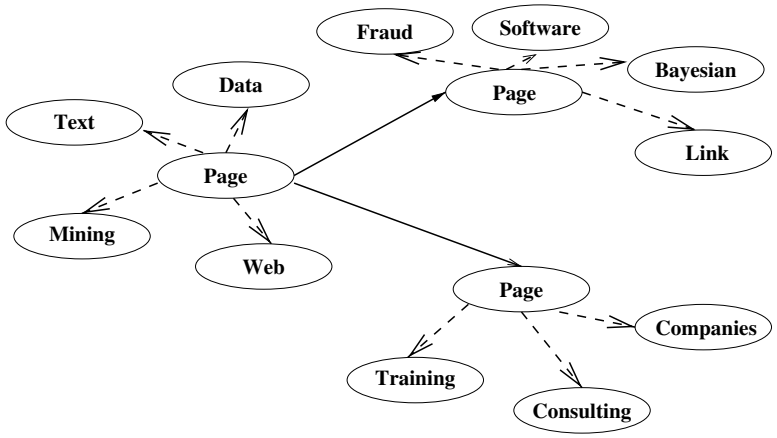

Figure 1.5. Graph representation for Web text and structure data. Solid arrows represent edges labeled ''hyperlink'' and dashed arrows represent edges labeled ''keyword.''

Desikan and Srivastava [13] have been investigating methods of finding patterns in dynamically evolving graphs, which can provide insights on trends as well as potential intrusions.

The third type of question that is commonly addressed in mining the web is:

What commonalities can we find in Web navigation patterns?

Answering this question is the problem of Web usage mining. Although mining clickstream data on the client side has been investigated [30], data is most easily collected and mined from Web servers [39]. Didimo and Liotta (Chapter 3) provide some graph representations and visualizations of navigation patterns. As Berendt points out [3], a graph representation of navigation allows the individual's website roadmap to be constructed. From the graph one can determine which pages act as starting points for the site, which collection of pages are typically navigated sequentially, and how easily (or often) are pages within the site accessed. Navigation graphs can be used to categorize Web surfers and can ultimately assist in organizing websites and ranking Web pages [3, 31, 35, 45].

1.3 BOOK OVERVIEW

The intention of this book is to provide an overview of the state of the art in mining graph data. To this end, we have gathered writings from colleagues that contribute to varied aspects of the problem. The aspects we focus on here are basic graph tools, techniques for mining graph data, and noteworthy graph mining applications.

Chapter 2 kicks off the *graph tools* part of the book by providing working definitions of key terms including graphs, subgraphs, and the operation that underlies much of graph mining, graph isomorphism. Here, Bunke and Neuhaus examine graph isomorphism techniques in detail and evaluate their merits based on the type of data that is available and the task that must be performed. Didimo and Liotta provide a thorough overview of graph visualization techniques in Chapter 3. They show how graph drawing algorithms assist and enhance the mining process and show how many of the techniques are customized for particular mining and other graph-based tasks. In Chapter 4, Chakrabarti and Faloutsos describe how the R-MAT algorithm can be used to generate graphs that exhibit properties found in real-world graphs. The ability to generate such graphs is useful for developing new mining algorithms, for testing existing algorithms, and for performing mining tasks such as anomaly detection.

In Part II, we highlight some of the most popular *mining techniques* that are currently developed for graph data. Chapters 5 through 7 focus on methods of discovering subgraph patterns from graph data. Yan and Han (Chapter 5) and Kuramochi and Karypis (Chapter 6) investigate efficient methods for extracting frequent substructures from graph data. Cook, Holder, and Ketkar (Chapter 7) evaluate subgraph patterns based on their ability to compress the input graph. Discovered subgraphs can be used to generate a graph grammar that is descriptive of the data, as shown by

Jonyer in Chapter 8. In contrast, Ohara et al. (Chapter 9) allow discovered subgraphs to represent features in a supervised learning problem. These subgraphs represent the attributes in a decision tree that can be learned from graph data. In Chapter 10, Liquière presents an alternative method for inducing concepts from graph data. By defining a partial order over the graph-based examples, the classification space can be viewed as a lattice and classical algorithms can be used to construct the concept definition from this lattice. In Chapter 11, Gärtner et al. define kernels on structural data that can be represented as a graph. The result can be applied to graph classification, making this problem tractable.

The next two chapters focus on properties of portions of the graph (individual edges, nodes, or neighborhoods around nodes), rather than on the graph as a whole, to perform the mining task. Shimbo and Ito, in Chapter 12, define an inner product of nodes in a graph. The resulting kernel can be used to analyze Web pages based on a combination of two factors: importance of the page and the degree to which two pages are related. Bhattacharya and Getoor use this location information in Chapter 13 to perform graph-based entity resolution. Edge attributes and constructed clusters of nodes can be used to identify the unique (nonduplicated) set of entities in the graph and to induce the corresponding entity graph.

The final part of the book features a collection of graph mining *applications*. These applications cover a diverse set of fields that are challenging and relevant, and for which data can be naturally represented as a graph. In Chapter 14, Okada provides an overview of chemical structure mining, including graph representations of the data and graph-based algorithms for analyzing the data. Zaki uses tree mining techniques to analyze bioinformatics data in Chapter 15. Specifically, the Sleuth algorithm is used to mine subtrees and can be applied to bioinformatics data such as RNA (ribonucleic acid) structures and phylogenetic subtrees. Tomkins and Kumar apply graph algorithms to Web data in Chapter 16 in which dense subgraphs are extracted that may represented communities of websites. Finally, Greenblatt and coworkers introduce a variety of graph mining tools in Chapter 17 that are effective for analyzing social network graphs. These applications are by no means comprehensive but illustrate the types of fields for which graph mining techniques are needed, and define the challenges that continue to drive this growing field of study.

REFERENCES

1. B. N. Ames, J. Mccann, and E. Yamasaki. Methods for detecting carcinogens and mutagens with the salmonella/mammalian-microsome mutagenicity test. *Mutation Research*, 31(6):347–364, 1975.
2. J. M. Barnard, G. M. Downsa, and P. Willet. Descriptor-based similarity measures for screening chemical databases. In H. J. Bohm and G. Schneider, eds. *Virtual Screening for Bioactive Molecules*, Wiley, New York, 2000.
3. B. Berendt. The semantics of frequent subgraphs: Mining and navigation pattern analysis. In Proceedings of WebKDD, Chicago, Illinois, 2005.
4. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic Web. *Scientific American* 279(5):34–43, 2001.

5. C. Borgelt and M. R. Berthold. Mining molecular fragments: Finding relevant substructures of molecules. In Proceedings of the IEEE International Conference on Data Mining, Maebashi City, Japan, pp. 51–58 2002.
6. S. Brin and L. Page. The anatomy of a large-scale hypertextual (web) search engine. *Computer Network and ISDN Systems* 30:107–117, 1998.
7. A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stat, and A. Tomkins. Graph Structure in the Web: Experiments and models. In Proceedings of the World Wide Web Conference, Amsterdam, The Netherlands, 2000.
8. S. Chakrabarti. Data mining for hypertext: A tutorial survey. *ACM SIGKDD Explorations* 1(2):1–11, 2000.
9. P. Chaovalit and L. Zhou. Movie Review Mining: A comparison between supervised and unsupervised classification approaches. In Proceedings of the Thirty-Eighth Annual Hawaii International Conference on System Sciences, Waikoloa, Hawaii, 2005.
10. A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds: Correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry* 34(2):786–797, 1991.
11. M. Deshpande, M. Kuramochi, and G. Karypis. Automated approaches for classifying structures. In Proceedings of the Workshop on Data Mining in Bioinformatics, Edmonton, Alberta, Canada, 2002.
12. M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering* 17(18):1036–1050, 2005.
13. P. Desikan and J. Srivastava. Mining Temporally Evolving Graphs. In Proceedings of WebKDD, Seattle, Washington, 2004.
14. EachMovie. http://research.compaq.com/SRC/eachmovie.
15. O. Etzioni. The World wide web: Quagmire or gold mine? *Communications of the ACM* 39(11):65–68, 1996.
16. L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In Proceedings of the International Conference on Machine Learning, Williamstown, Massachusetts, 2001.
17. A. Goldenberg and A. Moore. Tractable learning of large bayes net structures from sparse data. In Proceedings of the International Conference on Machine Learning, 2004.
18. J. Gonzalez, L. B. Holder, and D. J. Cook. Graph-based relational concept learning. In Proceedings of the International Machine Learning Conference, 2002.
19. C. Hansch, R. M. Muir, T. Fujita, C. F. Maloney, and M. Streich. The correlation of biological activity of plant growth-regulators and chloromycetin derivatives with hammett constants and partition coefficients. *Journal of the American Chemical Society* 85:2817–2824, 1963.
20. A. Inokuchi, T. Washio, T. Okada, and H. Motoda. Applying the apriori-based graph mining method to mutagenesis data analysis. *Journal of Computer Aided Chemistry* 2:87–92, 2001.
21. D. Jensen and J. Neville. Data mining in social networks. In Workshop on Dynamic Social Network Modeling and Analysis, Washington, DC, 2002.
22. N. Ketkar, L. B. Holder, and D. J. Cook. Qualitative comparison of graph-based and logic-based multi-relational data mining: a case study. In Proceedings of the KDD Workshop on Multi-Relational Data Mining, 2005.
23. R. D. King, S. H. Muggleton, A. Srinivasan, and M. J. E. Sternberg. Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities

to predict mutagenicity by inductive logic programming. In Proceedings of the National Academy of Sciences, Vol. 93, pp. 438–442, National Academy of Sciences, Washington, DC, 1996.
24. J. Kleinberg and S. Lawrence. The structure of the web. *Science* 294, 2001.
25. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46:604–632, 1999.
26. P. Kolari and A. Joshi. Web mining: Research and practice. *IEEE Computing in Science and Engineering* 6(4):49–53, 2004.
27. S. Kramer, B. Pfahringer, and C. Helma. Mining for causes of cancer: Machine learning experiments at various levels of details. In Proceedings of the Conference on Knowledge Discovery and Data Mining, pp. 233–226, Newport Beach, California, 1997.
28. J. Kubica, A. Goldenberg, P. Komarek, and A. Moore. A comparison of statistical and machine learning algorithms on the task of link completion. In Proceedings of the KDD Workshop on Link Analysis for Detecting Complex Behavior, 2003.
29. H. Lodhi and S. H. Muggleton. Is Mutagenesis Still Challenging? In Proceedings of the International Conference on Inductive Logic Programming, pp. 35–40, 2005.
30. A. Maniam. Graph-based click-stream mining for categorizing browsing activity in the world wide web. Master's thesis, University of Texas at Arlington, 2004.
31. J. E. McEneaney. Graphic and numerical methods to assess navigation in hypertext. *International Journal of Human-Computer Studies* 55:761–786, 2001.
32. A. McGovern and D. Jensen. Identifying predictive structures in relational data using multiple instance learning. In Proceedings of the International Conference on Machine Learning, 2003.
33. P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In Proceedings of the National Conference on Artificial Intelligence, pp. 187–192, 2002.
34. A. Mendelzon, G. Michaila, and T. Milo. Querying the world wide web. In Proceedings of the International Conference on Parallel and Distributed Information Systems, pp. 80–91, 1996.
35. R. Meo, P. L. Lanzi, M. Matera, and R. Esposito. Integrating web conceptual modeling and web usage mining. In Proceedings of WebKDD, 2004.
36. E. Ravasz and A.-L. Barabasi. Hierarchical organization in complex networks. *Physical Review E*, 67, 2003.
37. P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In Proceedings of the ACM Conference on Computed Supported Cooperative Work, pp. 175–186, 1994.
38. A. Srinivasan and R. D. King. Feature construction with inductive logic programming: A study of quantitative predictions of biological activity aided by structural attributes. *Data Mining and Knowledge Discovery* 3(1):37–57, 1999.
39. J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations* 1(2):1–12, 2000.
40. M. J. E. Sternberg and S. H. Muggleton. Structure activity relationships (SAR) and pharmacophore discovery using inductive logic programming (ILP). *QSAR and Combinatorial Science* 22, 2003.
41. The Internet Movie Database. http://www.imdb.com.
42. E. Vozalis and K. Margaritis. Recommender systems: An experimental comparison of two filtering algorithms. In Proceedings of the Ninth Panhellenic Conference in Informatics, 2003.

43. R. Weiss, B. Velez, and M. Sheldon. HyPursuit: A hierarchical network search engine that exploits context-link hypertext clustering. In Proceedings of the Conference on Hypertext and Hypermedia, pp. 180–193, 1996.
44. O. R. Zaiane and J. Han. Resource and knowledge discovery in global information systems: A preliminary design and experiment. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 331–336, 1995.
45. M. J. Zaki. Efficiently mining frequent trees in a forest. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, 2002.