

Managing and Mining Graph Data

ADVANCES IN DATABASE SYSTEMS

Volume 40

Series Editors

Ahmed K. Elmagarmid

Purdue University
West Lafayette, IN 47907

Amit P. Sheth

Wright State University
Dayton, OH 45435

For other titles published in this series, please visit www.springer.com/series/5573

Managing and Mining Graph Data

by

Charu C. Aggarwal

*IBM T.J. Watson Research Center
Hawthorne, NY, USA*

Haixun Wang

*Microsoft Research Asia
Beijing, China*



Charu C. Aggarwal
IBM
Thomas J. Watson Research
Center
19 Skyline Drive
Hawthorne, NY10532
USA
charu@us.ibm.com

Haixun Wang
Microsoft Research Asia
49 Zhichun Road
100190 Beijing
5F Sigma Center
China, People's Republic
haixunw@microsoft.com

ISSN 1386-2944
ISBN 978-1-4419-6044-3 e-ISBN 978-1-4419-6045-0
DOI 10.1007/978-1-4419-6045-0
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010920842

© Springer Science+Business Media, LLC 2010

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

List of Figures	xv
List of Tables	xxi
Preface	xxiii
1	
An Introduction to Graph Data	1
<i>Charu C. Aggarwal and Haixun Wang</i>	
1. Introduction	1
2. Graph Management and Mining Applications	3
3. Summary	8
References	9
2	
Graph Data Management and Mining: A Survey of Algorithms and Applications	13
<i>Charu C. Aggarwal and Haixun Wang</i>	
1. Introduction	13
2. Graph Data Management Algorithms	16
2.1 Indexing and Query Processing Techniques	16
2.2 Reachability Queries	19
2.3 Graph Matching	21
2.4 Keyword Search	24
2.5 Synopsis Construction of Massive Graphs	27
3. Graph Mining Algorithms	29
3.1 Pattern Mining in Graphs	29
3.2 Clustering Algorithms for Graph Data	32
3.3 Classification Algorithms for Graph Data	37
3.4 The Dynamics of Time-Evolving Graphs	40
4. Graph Applications	43
4.1 Chemical and Biological Applications	43
4.2 Web Applications	45
4.3 Software Bug Localization	51
5. Conclusions and Future Research	55
References	55
3	
Graph Mining: Laws and Generators	69
<i>Deepayan Chakrabarti, Christos Faloutsos and Mary McGlohon</i>	
1. Introduction	70
2. Graph Patterns	71

2.1	Power Laws and Heavy-Tailed Distributions	72
2.2	Small Diameters	77
2.3	Other Static Graph Patterns	79
2.4	Patterns in Evolving Graphs	82
2.5	The Structure of Specific Graphs	84
3.	Graph Generators	86
3.1	Random Graph Models	88
3.2	Preferential Attachment and Variants	92
3.3	Optimization-based generators	101
3.4	Tensor-based	108
3.5	Generators for specific graphs	113
3.6	Graph Generators: A summary	115
4.	Conclusions	115
	References	117
4		
	Query Language and Access Methods for Graph Databases	125
	<i>Huahai He and Ambuj K. Singh</i>	
1.	Introduction	126
1.1	Graphs-at-a-time Queries	126
1.2	Graph Specific Optimizations	127
1.3	GraphQL	128
2.	Operations on Graph Structures	129
2.1	Concatenation	130
2.2	Disjunction	131
2.3	Repetition	131
3.	Graph Query Language	132
3.1	Data Model	132
3.2	Graph Patterns	133
3.3	Graph Algebra	134
3.4	FLWR Expressions	137
3.5	Expressive Power	138
4.	Implementation of the Selection Operator	140
4.1	Graph Pattern Matching	140
4.2	Local Pruning and Retrieval of Feasible Mates	142
4.3	Joint Reduction of Search Space	144
4.4	Optimization of Search Order	146
5.	Experimental Study	148
5.1	Biological Network	148
5.2	Synthetic Graphs	150
6.	Related Work	152
6.1	Graph Query Languages	152
6.2	Graph Indexing	155
7.	Future Research Directions	155
8.	Conclusion	156
	Appendix: Query Syntax of GraphQL	156
	References	157
5		
	Graph Indexing	161
	<i>Xifeng Yan and Jiawei Han</i>	
1.	Introduction	161

2.	Feature-Based Graph Index	162
2.1	Paths	163
2.2	Frequent Structures	164
2.3	Discriminative Structures	166
2.4	Closed Frequent Structures	167
2.5	Trees	167
2.6	Hierarchical Indexing	168
3.	Structure Similarity Search	169
3.1	Feature-Based Structural Filtering	170
3.2	Feature Miss Estimation	171
3.3	Frequency Difference	172
3.4	Feature Set Selection	173
3.5	Structures with Gaps	174
4.	Reverse Substructure Search	175
5.	Conclusions	177
	References	178
6		
	Graph Reachability Queries: A Survey	181
	<i>Jeffrey Xu Yu and Jiefeng Cheng</i>	
1.	Introduction	181
2.	Traversal Approaches	186
2.1	Tree+SSPI	187
2.2	GRIPP	187
3.	Dual-Labeling	188
4.	Tree Cover	190
5.	Chain Cover	191
5.1	Computing the Optimal Chain Cover	193
6.	Path-Tree Cover	194
7.	2-HOP Cover	196
7.1	A Heuristic Ranking	197
7.2	A Geometrical-Based Approach	198
7.3	Graph Partitioning Approaches	199
7.4	2-Hop Cover Maintenance	202
8.	3-Hop Cover	204
9.	Distance-Aware 2-Hop Cover	205
10.	Graph Pattern Matching	207
10.1	A Special Case: $A \leftrightarrow D$	208
10.2	The General Cases	211
11.	Conclusions and Summary	212
	References	212
7		
	Exact and Inexact Graph Matching: Methodology and Applications	217
	<i>Kaspar Riesen, Xiaoyi Jiang and Horst Bunke</i>	
1.	Introduction	218
2.	Basic Notations	219
3.	Exact Graph Matching	221
4.	Inexact Graph Matching	226
4.1	Graph Edit Distance	227
4.2	Other Inexact Graph Matching Techniques	229
5.	Graph Matching for Data Mining and Information Retrieval	231

6.	Vector Space Embeddings of Graphs via Graph Matching	235
7.	Conclusions	239
	References	240
8		
	A Survey of Algorithms for Keyword Search on Graph Data	249
	<i>Haixun Wang and Charu C. Aggarwal</i>	
1.	Introduction	250
2.	Keyword Search on XML Data	252
	2.1 Query Semantics	253
	2.2 Answer Ranking	254
	2.3 Algorithms for LCA-based Keyword Search	258
3.	Keyword Search on Relational Data	260
	3.1 Query Semantics	260
	3.2 DBXplorer and DISCOVER	261
4.	Keyword Search on Schema-Free Graphs	263
	4.1 Query Semantics and Answer Ranking	263
	4.2 Graph Exploration by Backward Search	265
	4.3 Graph Exploration by Bidirectional Search	266
	4.4 Index-based Graph Exploration – the BLINKS Algorithm	267
	4.5 The ObjectRank Algorithm	269
5.	Conclusions and Future Research	271
	References	271
9		
	A Survey of Clustering Algorithms for Graph Data	275
	<i>Charu C. Aggarwal and Haixun Wang</i>	
1.	Introduction	275
2.	Node Clustering Algorithms	277
	2.1 The Minimum Cut Problem	277
	2.2 Multi-way Graph Partitioning	281
	2.3 Conventional Generalizations and Network Structure Indices	282
	2.4 The Girvan-Newman Algorithm	284
	2.5 The Spectral Clustering Method	285
	2.6 Determining Quasi-Cliques	288
	2.7 The Case of Massive Graphs	289
3.	Clustering Graphs as Objects	291
	3.1 Extending Classical Algorithms to Structural Data	291
	3.2 The XProj Approach	293
4.	Applications of Graph Clustering Algorithms	295
	4.1 Community Detection in Web Applications and Social Networks	296
	4.2 Telecommunication Networks	297
	4.3 Email Analysis	297
5.	Conclusions and Future Research	297
	References	299
10		
	A Survey of Algorithms for Dense Subgraph Discovery	303
	<i>Victor E. Lee, Ning Ruan, Ruoming Jin and Charu Aggarwal</i>	
1.	Introduction	304

2.	Types of Dense Components	305
2.1	Absolute vs. Relative Density	305
2.2	Graph Terminology	306
2.3	Definitions of Dense Components	307
2.4	Dense Component Selection	308
2.5	Relationship between Clusters and Dense Components	309
3.	Algorithms for Detecting Dense Components in a Single Graph	311
3.1	Exact Enumeration Approach	311
3.2	Heuristic Approach	314
3.3	Exact and Approximation Algorithms for Discovering Densest Components	322
4.	Frequent Dense Components	327
4.1	Frequent Patterns with Density Constraints	327
4.2	Dense Components with Frequency Constraint	328
4.3	Enumerating Cross-Graph Quasi-Cliques	328
5.	Applications of Dense Component Analysis	329
6.	Conclusions and Future Research	331
	References	333
11		
	Graph Classification	337
	<i>Koji Tsuda and Hiroto Saigo</i>	
1.	Introduction	337
2.	Graph Kernels	340
2.1	Random Walks on Graphs	341
2.2	Label Sequence Kernel	342
2.3	Efficient Computation of Label Sequence Kernels	343
2.4	Extensions	349
3.	Graph Boosting	349
3.1	Formulation of Graph Boosting	351
3.2	Optimal Pattern Search	353
3.3	Computational Experiments	354
3.4	Related Work	355
4.	Applications of Graph Classification	358
5.	Label Propagation	358
6.	Concluding Remarks	359
	References	359
12		
	Mining Graph Patterns	365
	<i>Hong Cheng, Xifeng Yan and Jiawei Han</i>	
1.	Introduction	366
2.	Frequent Subgraph Mining	366
2.1	Problem Definition	366
2.2	Apriori-based Approach	367
2.3	Pattern-Growth Approach	368
2.4	Closed and Maximal Subgraphs	369
2.5	Mining Subgraphs in a Single Graph	370
2.6	The Computational Bottleneck	371
3.	Mining Significant Graph Patterns	372
3.1	Problem Definition	372
3.2	gboost: A Branch-and-Bound Approach	373

3.3	gPLS: A Partial Least Squares Regression Approach	375
3.4	LEAP: A Structural Leap Search Approach	378
3.5	GraphSig: A Feature Representation Approach	382
4.	Mining Representative Orthogonal Graphs	385
4.1	Problem Definition	386
4.2	Randomized Maximal Subgraph Mining	387
4.3	Orthogonal Representative Set Generation	388
5.	Conclusions	389
	References	389
13		
	A Survey on Streaming Algorithms for Massive Graphs	393
	<i>Jian Zhang</i>	
1.	Introduction	393
2.	Streaming Model for Massive Graphs	395
3.	Statistics and Counting Triangles	397
4.	Graph Matching	400
4.1	Unweighted Matching	400
4.2	Weighted Matching	403
5.	Graph Distance	405
5.1	Distance Approximation using Multiple Passes	406
5.2	Distance Approximation in One Pass	411
6.	Random Walks on Graphs	412
7.	Conclusions	416
	References	417
14		
	A Survey of Privacy-Preservation of Graphs and Social Networks	421
	<i>Xintao Wu, Xiaowei Ying, Kun Liu and Lei Chen</i>	
1.	Introduction	422
1.1	Privacy in Publishing Social Networks	422
1.2	Background Knowledge	423
1.3	Utility Preservation	424
1.4	Anonymization Approaches	424
1.5	Notations	425
2.	Privacy Attacks on Naive Anonymized Networks	426
2.1	Active Attacks and Passive Attacks	426
2.2	Structural Queries	427
2.3	Other Attacks	428
3.	K -Anonymity Privacy Preservation via Edge Modification	428
3.1	K -Degree Generalization	429
3.2	K -Neighborhood Anonymity	430
3.3	K -Automorphism Anonymity	431
4.	Privacy Preservation via Randomization	433
4.1	Resilience to Structural Attacks	434
4.2	Link Disclosure Analysis	435
4.3	Reconstruction	437
4.4	Feature Preserving Randomization	438
5.	Privacy Preservation via Generalization	440
6.	Anonymizing Rich Graphs	441

6.1	Link Protection in Rich Graphs	442
6.2	Anonymizing Bipartite Graphs	443
6.3	Anonymizing Rich Interaction Graphs	444
6.4	Anonymizing Edge-Weighted Graphs	445
7.	Other Privacy Issues in Online Social Networks	446
7.1	Deriving Link Structure of the Entire Network	446
7.2	Deriving Personal Identifying Information from Social Net- working Sites	448
8.	Conclusion and Future Work	448
Acknowledgments		449
References		449
15		
A Survey of Graph Mining for Web Applications		455
<i>Debora Donato and Aristides Gionis</i>		
1.	Introduction	456
2.	Preliminaries	457
2.1	Link Analysis Ranking Algorithms	459
3.	Mining High-Quality Items	461
3.1	Prediction of Successful Items in a Co-citation Network	463
3.2	Finding High-Quality Content in Question-Answering Por- tals	465
4.	Mining Query Logs	469
4.1	Description of Query Logs	470
4.2	Query Log Graphs	470
4.3	Query Recommendations	477
5.	Conclusions	480
References		481
16		
Graph Mining Applications to Social Network Analysis		487
<i>Lei Tang and Huan Liu</i>		
1.	Introduction	487
2.	Graph Patterns in Large-Scale Networks	489
2.1	Scale-Free Networks	489
2.2	Small-World Effect	491
2.3	Community Structures	492
2.4	Graph Generators	494
3.	Community Detection	494
3.1	Node-Centric Community Detection	495
3.2	Group-Centric Community Detection	498
3.3	Network-Centric Community Detection	499
3.4	Hierarchy-Centric Community Detection	504
4.	Community Structure Evaluation	505
5.	Research Issues	507
References		508
17		
Software-Bug Localization with Graph Mining		515
<i>Frank Eichinger and Klemens Böhm</i>		
1.	Introduction	516
2.	Basics of Call Graph Based Bug Localization	517

2.1	Dynamic Call Graphs	517
2.2	Bugs in Software	518
2.3	Bug Localization with Call Graphs	519
2.4	Graph and Tree Mining	520
3.	Related Work	521
4.	Call-Graph Reduction	525
4.1	Total Reduction	525
4.2	Iterations	526
4.3	Temporal Order	528
4.4	Recursion	529
4.5	Comparison	531
5.	Call Graph Based Bug Localization	532
5.1	Structural Approaches	532
5.2	Frequency-based Approach	535
5.3	Combined Approaches	538
5.4	Comparison	539
6.	Conclusions and Future Directions	542
	Acknowledgments	543
	References	543
18		
	A Survey of Graph Mining Techniques for Biological Datasets	547
	<i>S. Parthasarathy, S. Tatikonda and D. Ucar</i>	
1.	Introduction	548
2.	Mining Trees	549
2.1	Frequent Subtree Mining	550
2.2	Tree Alignment and Comparison	552
2.3	Statistical Models	554
3.	Mining Graphs for the Discovery of Frequent Substructures	555
3.1	Frequent Subgraph Mining	555
3.2	Motif Discovery in Biological Networks	560
4.	Mining Graphs for the Discovery of Modules	562
4.1	Extracting Communities	564
4.2	Clustering	566
5.	Discussion	569
	References	571
19		
	Trends in Chemical Graph Data Mining	581
	<i>Nikil Wale, Xia Ning and George Karypis</i>	
1.	Introduction	582
2.	Topological Descriptors for Chemical Compounds	583
2.1	Hashed Fingerprints (FP)	584
2.2	Maccs Keys (MK)	584
2.3	Extended Connectivity Fingerprints (ECFP)	584
2.4	Frequent Subgraphs (FS)	585
2.5	Bounded-Size Graph Fragments (GF)	585
2.6	Comparison of Descriptors	585
3.	Classification Algorithms for Chemical Compounds	588
3.1	Approaches based on Descriptors	588
3.2	Approaches based on Graph Kernels	589
4.	Searching Compound Libraries	590

<i>Contents</i>	xiii
4.1 Methods Based on Direct Similarity	591
4.2 Methods Based on Indirect Similarity	592
4.3 Performance of Indirect Similarity Methods	594
5. Identifying Potential Targets for Compounds	595
5.1 Model-based Methods For Target Fishing	596
5.2 Performance of Target Fishing Strategies	600
6. Future Research Directions	600
References	602
Index	607

List of Figures

3.1	Power laws and deviations	73
3.2	Hop-plot and effective diameter	78
3.3	Weight properties of the campaign donations graph: (a) shows all weight properties, including the densification power law and WPL. (b) and (c) show the Snapshot Power Law for in- and out-degrees. Both have slopes > 1 (“fortification effect”), that is, that the more campaigns an organization supports, the superlinearly-more money it donates, and similarly, the more donations a candidate gets, the more average amount-per-donation is received. Inset plots on (c) and (d) show iw and ow versus time. Note they are very stable over time.	82
3.4	<i>The Densification Power Law</i> The number of edges $E(t)$ is plotted against the number of nodes $N(t)$ on log-log scales for (a) the arXiv citation graph, (b) the patents citation graph, and (c) the Internet Autonomous Systems graph. All of these grow over time, and the growth follows a power law in all three cases 58.	83
3.5	Connected component properties of Postnet network, a network of blog posts. Notice that we experience an early gelling point at (a), where the diameter peaks. Note in (b), a log-linear plot of component size vs. time, that at this same point in time the giant connected component takes off, while the sizes of the second and third-largest connected components (CC2 and CC3) stabilize. We focus on these next-largest connected components in (c).	84

3.6	Timing patterns for a network of blog posts. (a) shows the entropy plot of edge additions, showing burstiness. The inset shows the addition of edges over time. (b) describes the decay of post popularity. The horizontal axis indicates time since a post's appearance (aggregated over all posts), while the vertical axis shows the number of links acquired on that day.	84
3.7	The Internet as a "Jellyfish"	85
3.8	The "Bowtie" structure of the Web	87
3.9	The Erdős-Rényi model	88
3.10	The Barabási-Albert model	93
3.11	The edge copying model	96
3.12	The Heuristically Optimized Tradeoffs model	103
3.13	The small-world model	105
3.14	The Waxman model	106
3.15	The R-MAT model	109
3.16	<i>Example of Kronecker multiplication</i> Top: a "3-chain" and its Kronecker product with itself; each of the X_i nodes gets expanded into 3 nodes, which are then linked together. Bottom row: the corresponding adjacency matrices, along with matrix for the fourth Kronecker power G_4 .	112
4.1	A sample graph query and a graph in the database	128
4.2	SQL-based implementation	128
4.3	A simple graph motif	130
4.4	(a) Concatenation by edges, (b) Concatenation by unification	131
4.5	Disjunction	131
4.6	(a) Path and cycle, (b) Repetition of motif G_1	132
4.7	A sample graph with attributes	132
4.8	A sample graph pattern	133
4.9	A mapping between the graph pattern in Figure 4.8 and the graph in Figure 4.7	134
4.10	An example of valued join	135
4.11	(a) A graph template with a single parameter \mathcal{P} , (b) A graph instantiated from the graph template. \mathcal{P} and G are shown in Figure 4.8 and Figure 4.7.	136
4.12	A graph query that generates a co-authorship graph from the DBLP dataset	137
4.13	A possible execution of the Figure 4.12 query	138
4.14	The translation of a graph into facts of Datalog	139

4.15	The translation of a graph pattern into a rule of Datalog	139
4.16	A sample graph pattern and graph	143
4.17	Feasible mates using neighborhood subgraphs and profiles. The resulting search spaces are also shown for different pruning techniques.	143
4.18	Refinement of the search space	146
4.19	Two examples of search orders	147
4.20	Search space for clique queries	149
4.21	Running time for clique queries (low hits)	149
4.22	Search space and running time for individual steps (synthetic graphs, low hits)	151
4.23	Running time (synthetic graphs, low hits)	151
5.1	Size-increasing Support Functions	165
5.2	Query and Features	170
5.3	Edge-Feature Matrix	171
5.4	Frequency Difference	172
5.5	cIndex	177
6.1	A Simple Graph G (left) and Its Index (right) (Figure 1 in 32)	187
6.2	Tree Codes Used in Dual-Labeling (Figure 2 in 34)	189
6.3	Tree Cover (based on Figure 3.1 in 1)	190
6.4	Resolving a virtual node	194
6.5	A Directed Graph, and its Two DAGs, G_{\downarrow} and G_{\uparrow} (Figure 2 in 13)	197
6.6	Reachability Map	198
6.7	Balanced/Unbalanced $S(A_w, w, D_w)$	200
6.8	Bisect G into G_A and G_D (Figure 6 in 14)	201
6.9	Two Maintenance Approaches	203
6.10	Transitive Closure Matrix	204
6.11	The 2-hop Distance Aware Cover (Figure 2 in 10)	206
6.12	The Algorithm Steps (Figure 3 in 10)	207
6.13	Data Graph (Figure 1(a) in 12)	209
6.14	A Graph Database for G_D (Figure 2 in 12)	210
7.1	Different kinds of graphs: (a) undirected and unlabeled, (b) directed and unlabeled, (c) undirected with labeled nodes (different shades of gray refer to different labels), (d) directed with labeled nodes and edges.	220
7.2	Graph (b) is an induced subgraph of (a), and graph (c) is a non-induced subgraph of (a).	221

7.3	Graph (b) is isomorphic to (a), and graph (c) is isomorphic to a subgraph of (a). Node attributes are indicated by different shades of gray.	222
7.4	Graph (c) is a maximum common subgraph of graph (a) and (b).	224
7.5	Graph (a) is a minimum common supergraph of graph (b) and (c).	225
7.6	A possible edit path between graph g_1 and graph g_2 (node labels are represented by different shades of gray).	227
7.7	Query and database graphs.	232
8.1	Query Semantics for Keyword Search $Q = \{x, y\}$ on XML Data	253
8.2	Schema Graph	261
8.3	The size of the join tree is only bounded by the data Size	261
8.4	Keyword matching and join trees enumeration	262
8.5	Distance-balanced expansion across clusters may perform poorly.	266
9.1	The Sub-structural Clustering Algorithm (High Level Description)	294
10.1	Example Graph to Illustrate Component Types	309
10.2	Simple example of web graph	316
10.3	Illustrative example of shingles	316
10.4	Recursive Shingling Step	317
10.5	Example of CSV Plot	320
10.6	The Set Enumeration Tree for $\{x, y, z\}$	329
11.1	Graph classification and label propagation.	338
11.2	Prediction rules of kernel methods.	339
11.3	(a) An example of labeled graphs. Vertices and edges are labeled by uppercase and lowercase letters, respectively. By traversing along the bold edges, the label sequence (2.1) is produced. (b) By repeating random walks, one can construct a list of probabilities.	341
11.4	A topologically sorted directed acyclic graph. The label sequence kernel can be efficiently computed by dynamic programming running from right to left.	346
11.5	Recursion for computing $r(x_1, x'_1)$ using recursive equation (2.11). $r(x_1, x'_1)$ can be computed based on the pre-computed values of $r(x_2, x'_2)$, $x_2 > x_1$, $x'_2 > x'_1$.	346
11.6	Feature space based on subgraph patterns. The feature vector consists of binary pattern indicators.	350

11.7	Schematic figure of the tree-shaped search space of graph patterns (i.e., the DFS code tree). To find the optimal pattern efficiently, the tree is systematically expanded by rightmost extensions.	353
11.8	Top 20 discriminative subgraphs from the CPDB dataset. Each subgraph is shown with the corresponding weight, and ordered by the absolute value from the top left to the bottom right. H atom is omitted, and C atom is represented as a dot for simplicity. Aromatic bonds appeared in an open form are displayed by the combination of dashed and solid lines.	356
11.9	Patterns obtained by gPLS. Each column corresponds to the patterns of a PLS component.	357
12.1	AGM: Two candidate patterns formed by two chains	368
12.2	Graph Pattern Application Pipeline	371
12.3	Branch-and-Bound Search	375
12.4	Structural Proximity	379
12.5	Frequency vs. G-test score	381
13.1	Layered Auxiliary Graph. Left, a graph with a matching (solid edges); Right, a layered auxiliary graph. (An illustration, not constructed from the graph on the left. The solid edges show potential augmenting paths.)	402
13.2	Example of clusters in covers.	410
14.1	Resilient to subgraph attacks	434
14.2	The interaction graph example and its generalization results	444
15.1	Relation Models for Single Item, Double Item and Multiple Items	462
15.2	Types of Features Available for Inferring the Quality of Questions and Answers	466
16.1	Different Distributions. A dashed curve shows the true distribution and a solid curve is the estimation based on 100 samples generated from the true distribution. (a) Normal distribution with $\mu = 1$, $\sigma = 1$; (b) Power law distribution with $x_{min} = 1$, $\alpha = 2.3$; (c) Loglog plot, generated via the toolkit in 17.	490
16.2	A toy example to compute clustering coefficient: $C_1 = 3/10$, $C_2 = C_3 = C_4 = 1$, $C_5 = 2/3$, $C_6 = 3/6$, $C_7 = 1$. The global clustering coefficient following Eqs. (2.5) and (2.6) are 0.7810 and 0.5217, respectively.	492
16.3	A toy example (reproduced from 61)	496
16.4	Equivalence for Social Position	500

17.1	An unreduced call graph, a call graph with a structure affecting bug, and a call graph with a frequency affecting bug.	518
17.2	An example PDG, a subgraph and a topological graph minor.	524
17.3	Total reduction techniques.	526
17.4	Reduction techniques based on iterations.	527
17.5	A raw call tree, its first and second transformation step.	527
17.6	Temporal information in call graph reductions.	529
17.7	Examples for reduction based on recursion.	530
17.8	Follow-up bugs.	537
18.1	Structural alignment of two FHA domains. FHA1 of Rad53 (left) and FHA of Chk2 (right)	559
18.2	Frequent Topological Structures Discovered by TSMiner	560
18.3	Benefits of Ensemble Strategy for Community Discovery in PPI networks in comparison to community detection algorithm MCODE and clustering algorithm MCL. The Y-axis represents $-\log(p\text{-value})$.	568
18.4	Soft Ensemble Clustering improves the quality of extracted clusters. The Y-axis represents $-\log(p\text{-value})$.	569
19.1	Performance of indirect similarity measures (MG) as compared to similarity searching using the Tanimoto coefficient (TM).	595
19.2	Cascaded SVM Classifiers.	598
19.3	Precision and Recall results	599

List of Tables

3.1	Table of symbols	71
4.1	Comparison of different query languages	154
6.1	The Time/Space Complexity of Different Approaches 25	183
6.2	A Reachability Table for G_{\downarrow} and G_{\uparrow}	198
10.1	Graph Terminology	306
10.2	Types of Dense Components	308
10.3	Overview of Dense Component Algorithms	311
17.1	Examples for the effect of call graph reduction techniques.	531
17.2	Example table used as input for feature-selection algorithms.	536
17.3	Experimental results.	540
19.1	Design choices made by the descriptor spaces.	586
19.2	SAR performance of different descriptors.	587

Preface

The field of graph mining has seen a rapid explosion in recent years because of new applications in computational biology, software bug localization, and social and communication networking. This book is designed for studying various applications in the context of managing and mining graphs. Graph mining has been studied by the theoretical community extensively in the context of numerous problems such as graph partitioning, node clustering, matching, and connectivity analysis. However the traditional work in the theoretical community cannot be directly used in practical applications because of the following reasons:

- The definitions of problems such as graph partitioning, matching and dimensionality reduction are too “clean” to be used with real applications. In real applications, the problem may have different variations such as a disk-resident case, a multi-graph case, or other constraints associated with the graphs. In many cases, problems such as frequent sub-graph mining and dense graph mining may have a variety of different flavors for different scenarios.
- The size of the applications in real scenarios are often very large. In such cases, the graphs may not be stored in main memory, but may be available only on disk. A classic example of this is the case of web and social network graphs, which may contain millions of nodes. As a result, it is often necessary to design specialized algorithms which are sensitive to disk access efficiency constraints. In some cases, the entire graph may not be available at one time, but may be available in the form of a continuous stream. This is the case in many applications such as social and telecommunication networks in which edges are received continuously.

The book will study the problem of managing and mining graphs from an applied point of view. It is assumed that the underlying graphs are massive and cannot be held in main memory. This change in assumption has a critical impact on the algorithms which are required to process such graphs. The problems studied in the book include algorithms for frequent pattern mining, graph

matching, indexing, classification, clustering, and dense graph mining. In many cases, the problem of graph management and mining has been studied from the perspective of structured and XML data. Where possible, we have clarified the connections with the methods and algorithms designed by the XML data management community. We also provide a detailed discussion of the application of graph mining algorithms in a number of recent applications such as graph privacy, web and social networks.

Many of the graph algorithms are sensitive to the application scenario in which they are encountered. Therefore, we will study the usage of many of these techniques in real scenarios such as the web, social networks, and biological data. This provides a better understanding of how the algorithms in the book apply to different scenarios. Thus, the book provides a comprehensive summary both from an algorithmic and applied perspective.