Algorithms and Models for Network Data and Link Analysis

Network data are produced automatically by everyday interactions – social networks, power grids, and citations between documents are a few examples. Such data capture social and economic behavior in a form that can be analyzed using powerful computational tools. This book is a guide to both basic and advanced techniques and algorithms for extracting useful information from network data. The content is organized around "tasks," grouping the algorithms needed to gather specific types of information and thus answer specific types of questions. Examples include similarity between nodes in a network, prestige or centrality of individual nodes, and dense regions or communities in a network. Algorithms are derived in detail and summarized in pseudo-code. The book is intended primarily for computer scientists, engineers, statisticians, and physicists, but is accessible to network scientists based in the social sciences. Matlab/Octave code illustrating some of the algorithms will gradually be available at:
http://www.cambridge.org/9781107125773.

François Fouss, Marco Saerens, and Masashi Shimbo received their Ph.D. degrees respectively from the Université catholique de Louvain, Belgium; the Université Libre de Bruxelles, Belgium; and Kyoto University, Japan. François Fouss and Marco Saerens are currently professors in computer science at the Université catholique de Louvain, Belgium, and Masashi Shimbo is associate professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. Their research and teaching interests include artificial intelligence, data mining, machine learning, pattern recognition, and natural language processing, with a focus on graph-based techniques.

Algorithms and Models for Network Data and Link Analysis

François Fouss

Université catholique de Louvain

Marco Saerens

Université catholique de Louvain

Masashi Shimbo

Nara Institute of Science and Technology

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107125773

First published 2016

Printed in the United States of America by Sheridan Books, Inc.

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloguing in Publication Data
Fouss, Franpcois, author. | Saerens, Marco, author. | Shimbo, Masashi, author.
Algorithms and models for network data and link analysis / Franpcois Fouss, Universitbe catholique de Louvain, Marco Saerens, Universitbe catholique de Louvain, Masashi Shimbo, Nara Institute of Science and Technology.
Cambridge, United Kingdom; New York: Cambridge University Press, 2016. | Includes bibliographical references and index.
LCCN 2016008448 | ISBN 9781107125773 (hardback : alk. paper)
LCSH: Network analysis (Planning) – Mathematics.
LCC T57.85 .F68 2016 | DDC 004.6/5–dc23
LC record available at https://lccn.loc.gov/2016008448

ISBN 978-1-107-12577-3 Hardback

Contents

List of Algorithms *page* xiii
List of Symbols and Notation xvii
Preface xxiii

1 Preliminaries and Notation **1**
1.1 Introduction 1
1.2 Content of the Book 3
1.3 Basic Definitions and Notation 6
1.3.1 Basic Graph Concepts 7
1.3.2 Standard Associated Matrices 11
1.3.3 Exploring the Graph and Cutting the Graph into Smaller Pieces 20
1.4 Building a Graph from Data 22
1.4.1 ϵ-Neighbor Graph 23
1.4.2 k-Nearest Neighbor Graph 24
1.4.3 Mutual k-NN Graph 24
1.5 Basic Markov Chain Concepts 25
1.5.1 Transition Matrix 25
1.5.2 Multistep Transition Matrix 26
1.5.3 Some Properties of Markov Chains and States 27
1.5.4 Defining a Random Walk Model on a Graph 29
1.5.5 Stationary Distribution of a Regular Markov Chain 30
1.5.6 Stationary Distribution of a Random Walk on an Undirected Graph 31
1.5.7 Fundamental Matrix of a Killed Random Walk 32
1.5.8 Stochastic Complementation 32
1.6 Average First Passage Time, Average Commute Time, and Related Quantities 34
1.6.1 A Generic Quantity: Expected Cost before Absorption 34
1.6.2 Average First Passage Cost 36
1.6.3 Average First Passage Time and Average Commute Time 36
1.6.4 Probabilities of Absorption 37
1.6.5 Expected Number of Visits 38
1.7 Basic Notions about Kernels on a Graph 38
1.7.1 Kernel Matrix 39
1.7.2 Kernels on a Graph 41
1.7.3 Useful Transformations of the Kernel Matrix 41

1.7.4 Computing a Euclidean Distance Matrix from a Kernel Matrix, and Vice Versa 44
1.7.5 An Illustrative Example 45
1.8 Useful Properties and Applications of the Laplacian Matrix and Its Pseudoinverse 46
1.8.1 Basic Properties of L, L^+, and L^+ 46
1.8.2 Application to the Computation of Random Walk–Based Quantities 49
1.9 Expectation-Maximization in a Nutshell 50
1.9.1 Majorization Technique 50
1.9.2 Jensen's Inequality 51
1.9.3 Expectation-Maximization Algorithm 51
1.10 Shortest-Path or Geodesic Distance 54
1.10.1 Floyd-Warshall All-Pairs Shortest-Path Algorithm 54
1.10.2 Matrix Form of the Floyd-Warshall Algorithm 56
1.10.3 Computing Connected Components from the Distance Matrix 57
1.11 Basic Standard Assumptions Used in This Book 58

2 Similarity/Proximity Measures between Nodes 59
2.1 Introduction 59
2.2 An Illustrative Example 60
2.3 A Quick Reminder about Similarities and Dissimilarities 61
2.3.1 General Conditions for Dissimilarity Measures 61
2.3.2 General Conditions for Similarity Measures 62
2.4 Local Similarity Measures 62
2.5 Global Similarity and Distance Measures 67
2.5.1 Katz Index 68
2.5.2 Resistance Distance 69
2.5.3 Commute-Time Distance and Euclidean Commute-Time Distance 74
2.5.4 SimRank and an Extension for Comparing Two Graphs 83
2.6 Kernel-Based Similarity Measures 86
2.6.1 Exponential Diffusion Kernel 87
2.6.2 Laplacian Exponential Diffusion Kernel 87
2.6.3 Regularized Laplacian Kernel and Variants 89
2.6.4 Commute-Time or Resistance-Distance Kernel 92
2.6.5 Similarities Based on Diffusion Models: Regularized Commute-Time Kernel and Random Walk with Restart Similarity 94
2.6.6 Markov Diffusion Distance and Kernel 97
2.7 Further Reading 101

3* Families of Dissimilarity between Nodes 102
3.1 Introduction 102
3.2* Logarithmic Forest and Walk Distances 103
3.2.1 Logarithmic Forest Distance 103
3.2.2 Walk Distance 104

3.3* The p-Resistance Distance 105
3.3.1 Definition of p-Resistance 105
3.3.2 An Alternative Definition of p-Resistance 107
3.4* Bag-of-Paths Framework 108
3.4.1 General Idea 108
3.4.2 Background and Notation 108
3.4.3 A Gibbs-Boltzmann Distribution on the Set of Paths 109
3.4.4 Computing the Probability of Sampling a Path Starting in i and Ending in j 111
3.4.5 Bag-of-Hitting-Paths Model 114
3.5* Three Distance Measures Based on the Bag-of-Hitting-Paths Probabilities 120
3.5.1 First Distance Based on the Associated Surprisal Measure 121
3.5.2 Second Distance Based on the Bag of Hitting Paths 123
3.5.3 Third, Simplified Distance Based on the Bag of Hitting Paths 126
3.6* Randomized Shortest-Path Dissimilarity and the Free Energy Distance 126
3.6.1 Randomized Shortest-Path Dissimilarity 126
3.6.2 Free Energy, or Potential, Distance 128
3.7* Bag-of-Paths Absorption Probabilities 130
3.7.1 Computing the Bag-of-Paths Absorption Probabilities 130
3.7.2 Computing Absorption Probabilities in Function of $\mathbf{L}^+$ 132
3.8* Bag-of-Paths Covariance Measure between Nodes 135
3.8.1 Definition of the Bag-of-Paths Covariance Measure 135
3.8.2 Computation of the Covariance Measure 138

4 Centrality Measures on Nodes and Edges 143
4.1 Introduction 143
4.2 Standard Centrality Measures 144
4.2.1 Closeness Centrality 144
4.2.2 Shortest-Path Eccentricity 146
4.2.3 Shortest-Path Betweenness Centrality 146
4.2.4 Load Betweenness Centrality 154
4.2.5 Shortest-Path Likelihood Betweenness 155
4.3 Some Closeness Centrality Measures Based on Node Similarity 156
4.3.1 Katz and Total Communicability Centrality 156
4.3.2 Subgraph Centralities 157
4.4 Random Eccentricity Measure 158
4.5 An Electrical and Random Walk–Based Betweenness Centrality 160
4.5.1 Current-Flow Node Betweenness 160
4.5.2 Random Walk Interpretation of Current-Flow Betweenness 163
4.5.3 Group Betweenness 166
4.6 Markov and Current-Flow Closeness Centrality 166
4.6.1 Markov Closeness Centrality 166
4.6.2 Current-Flow Closeness Centrality 167
4.7* Bag-of-Paths Betweenness Centrality 168
4.7.1 Node Betweenness Centrality 169
4.7.2 Group Betweenness Centrality 172

4.8* Randomized Shortest-Path Node and Net Flow Betweennesses 174
4.8.1 Randomized Shortest-Path Node Betweenness Centrality 174
4.8.2 An Alternative RSP Betweenness Based on Net Flow 181
4.9 Some Node, Edge, and Network Criticality Measures 182
4.9.1 Some Standard Network Criticality Measures 183
4.9.2 Generic Node Criticality Measures Based on Node Removal 185
4.9.3 A Node Criticality Measure Based on Communicability 186
4.9.4 A Node and an Edge Criticality Measure Based on Sensitivity 187
4.9.5 An Edge Criticality Measure Based on Spanning Trees 190
4.9.6* A Node Criticality Measure Based on the Bag-of-Paths Framework 195
4.9.7* An Edge Criticality Measure Based on Simple Free Energy Distance 197

5 Identifying Prestigious Nodes 201
5.1 Introduction 201
5.2 Some Classic Node Prestige Measures 202
5.2.1 Node Indegree 202
5.2.2 Prestige by Proximity 202
5.2.3 A Spectral Measure of Prestige 203
5.2.4 Prestige Based on Indirect Links: Katz's Index, Hubbell's Index, and Total Communicability 204
5.3 Citation Influence 207
5.4 Some Rating Methods Based on Least Squares 208
5.4.1 Type of Graph on Which the Model Applies 209
5.4.2 A Standard Linear Least Squares Model 211
5.4.3 Maximum Likelihood Estimation 211
5.4.4 Interpreting the Measure 213
5.4.5 Probability of Winning against a Team 214
5.4.6 Generalized Row Sum Method 214
5.5 PageRank Algorithm 215
5.5.1 Basic PageRank 215
5.5.2 PageRank and the Random Walk on a Graph 217
5.5.3 Improvement to the Basic PageRank Model 218
5.5.4 Calculating Score Vectors 220
5.5.5 Personalized PageRank: Placing Weights on Nodes 221
5.5.6 A Consensus-Reaching Interpretation of PageRank 221
5.6 HITS: Hubs and Authorities 224
5.6.1 HITS Algorithm 224
5.6.2 HITS and Bibliometrics 227
5.6.3* HITS and Principal Components Analysis 227
5.7 Probabilistic HITS 229
5.7.1 Dealing with Multiple Clusters/Topics in a Graph 229
5.7.2 Probabilistic Latent Semantic Analysis 230
5.7.3 Probabilistic HITS 232

5.8* A Simple Bag-of-Paths Prestige Measure 232
5.9 Further Reading 233

6 Labeling Nodes: Within-Network Classification 235
6.1 Introduction 235
6.2 Least Squares with Laplacian Regularization 237
6.2.1 Standard Case 237
6.2.2 Direct Extensions 241
6.3 Classification through Harmonic Functions 243
6.3.1 Basic Elementwise Solution 243
6.3.2 A Matrix Closed-Form Solution 245
6.3.3 Solution in Terms of the Laplacian Matrix 245
6.3.4 Direct Extensions 247
6.4 Two Simple Random Walk–Based Approaches 248
6.4.1 Random Walk with Restart Approach 248
6.4.2 Discriminative Random Walks Approach 250
6.5* Classification through the Bag-of-Paths Group Betweenness 252
6.6 Considering Node Features: Regression Models with Laplacian Regularization 254
6.6.1 Simple Ridge Regression with Laplacian Regularization 255
6.6.2 Kernel Ridge Regression with Laplacian Regularization 257
6.6.3 Ridge Logistic Regression with Laplacian Regularization 259
6.7 Considering Node Features: Adding Graph Principal Scores as Structural Features 266
6.7.1 Maximizing Moran's I 266
6.7.2 Minimizing Geary's c 268
6.7.3 Local Principal Components Analysis 269
6.8 Considering Node Features: Autologit Model 271
6.9 Considering Node Features: A Kernel Ridge Logistic Regression 272
6.10 Further Reading 275

7 Clustering Nodes 276
7.1 Introduction 276
7.2 An Illustrative Example 278
7.3 A Simple, Generic, Distance-Based k-Means 278
7.4 Clustering with a Kernel k-Means 280
7.4.1 Main Idea 281
7.4.2 Kernel k-Means Algorithm 283
7.4.3 Kernel Iterative k-Means 285
7.4.4 Choice of Kernel Matrix 290
7.4.5 Application to the Illustrative Example 290
7.5 A Simple Label Propagation Algorithm 292
7.5.1 Basic Label Propagation Algorithm 292
7.5.2 An Improved Label Propagation Algorithm 294
7.6 Markov Cluster Process 295
7.6.1 Main Idea 295
7.6.2 Markov Cluster Algorithm 297

7.6.3 Regularized Markov Cluster Algorithm 298
7.6.4 Application to the Illustrative Example 299
7.7 Simple Top-Down, Divisive, Greedy Clustering: Kernighan-Lin Algorithm 301
7.7.1 A Heuristic Procedure for Minimizing Graph Cut 301
7.7.2 Difference in Graph Cut When Swapping Two Nodes 302
7.7.3 A Heuristic Algorithm Greedily Improving Graph Cut 303
7.8 Spectral Clustering 304
7.8.1 Graph Cut 305
7.8.2 Ratio Cut 307
7.8.3 Normalized Cut 311
7.8.4 Partitioning Nodes into Three or More Clusters 315
7.8.5 Some Links between Ratio Cut and the k-Means Algorithm 319
7.8.6 Variations on Spectral Clustering 322
7.9 Modularity Criterion and Its Spectral Optimization 323
7.9.1 Modularity Criterion 324
7.9.2 Maximization of Modularity 332
7.9.3 Two-Way Partitioning Based on Modularity 333
7.9.4 Splitting into More Than Two Clusters: Recursive Partitioning Based on Modularity 334
7.9.5 Application to the Illustrative Example 336
7.10 A Latent Class Model Based on a Bag of Links 336
7.10.1 Latent Class Model 336
7.10.2 Application of the Expectation-Maximization Algorithm 339
7.10.3 Estimating the Number of Natural Latent Classes 344
7.10.4* Expectation-Maximization Revisited 345
7.10.5* A Few Words about the Basic Stochastic Block Model 346

8 Finding Dense Regions 349
8.1 Introduction 349
8.2 Basic Local Density Measures 349
8.2.1 Local Density Measure 350
8.2.2 Clustering Coefficient 351
8.3 Smoothing the Local Measures 354
8.3.1 PageRank-Like Smoothing 355
8.3.2 Smoothing through Laplacian Regularization 356
8.4* Bag-of-Forests Density Index 357
8.4.1 A Boltzmann Distribution on the Set of Forests 358
8.4.2 Bag-of-Forests Density Index 359
8.4.3 Computation of the Partition Function $\mathcal{Z}$ 360
8.4.4 Computation of the Bag-of-Forests Density Index 361
8.4.5 A Link with the Spanning Tree Criticality Measure 363
8.5 Identifying Network k-Cores 364
8.5.1 Basic Properties of k-Cores 366
8.5.2 Computing k-Cores 366
8.5.3 Computing the Core Number 367
8.5.4 Generalized Cores 369
8.5.5 Links with a Greedy Algorithm for Finding Dense Subgraphs 371

8.6 Kernel Bottom-Up Hierarchical Clustering 372
8.6.1 A Kernel Version of Ward's Hierarchical Clustering 373
8.6.2 Some Links with Spectral Clustering 376
8.7 Bottom-up, Agglomerative Clustering Based on Modularity: Louvain Method 377
8.7.1 Description of the Algorithm 377
8.7.2 Gain in Modularity When Moving One Node 380
8.8 Bottom-up, Agglomerative Clustering Based on a Spin-Glass Process 381
8.8.1 A Generic Cost Function for Community Detection 381
8.8.2 A Simple Particular Cost Function 382
8.8.3 Optimizing the Cost Function 384
8.8.4 Difference in Cost Function When Moving One Node 385
8.9 A Heuristic Procedure for Maximum Clique Detection 386
8.9.1 A Quadratic Problem Formulation of the Maximum Clique Problem 386
8.9.2 An Extension to L_p-Norm Constraint 387
8.9.3 A Fixed-Point Procedure 387
8.10 Further Reading 389

9 Bipartite Graph Analysis 390
9.1 Introduction 390
9.2 An Illustrative Example and Definition of the Biadjacency Matrix 391
9.3 Simple Correspondence Analysis 393
9.3.1 Introduction and Notation 394
9.3.2 A First Procedure: Maximizing Correlation 395
9.3.3 Dealing with More Than One Dimension 401
9.3.4 Other Derivations of Correspondence Analysis 402
9.3.5 Application to the Illustrative Example 404
9.4 A Probabilistic Reputation Model 405
9.4.1 Description of the Model 406
9.4.2 Likelihood Function 407
9.4.3 Estimating the Reputation Scores 407
9.4.4 A Simple Bayesian Extension 409
9.5 Bi-Clustering Bipartite Graphs 411
9.5.1 Chi Square Statistic 412
9.5.2 Profile Vectors 413
9.5.3 Chi Square Distance 414
9.5.4 Total Inertia of the Cloud of Profile Vectors 415
9.5.5 Decomposition of Inertia 416
9.5.6 Bi-Clustering Procedure 418
9.6 Nonnegative Matrix Factorization 422
9.6.1 Introduction 422
9.6.2 Multiplicative Update Procedure 426
9.6.3 Alternating Least Squares Procedure 431
9.6.4 Extensions of the Basic Models 432
9.6.5 Problem of Link Prediction 433

9.7 A Latent Class Model 434
9.7.1 Description of the Model 434
9.7.2 Application to the Illustrative Example 435

10 Graph Embedding **437**
10.1 Introduction 437
10.2 Kernel Principal Components Analysis 438
10.2.1 Defining the Embedding 440
10.2.2 Finding the Axes in the Direction of Maximum Variance 441
10.2.3 Computing the Coordinates, or Scores 443
10.2.4* Dealing with Indefinite Similarity Matrices 445
10.3 Classical Multidimensional Scaling: Basic Notions 447
10.3.1 Inner Products from Euclidean Distances 447
10.3.2 Node Vectors from the Spectral Decomposition of the Inner Products Matrix 448
10.3.3 Case of a Non-Euclidean Distance 449
10.4 Markov Diffusion Map 450
10.4.1 Diffusion Distance and Diffusion Map 450
10.4.2 Links with Spectral Clustering 455
10.4.3* Another Interpretation of the Diffusion Map 456
10.4.4 A Kernel View of the Diffusion Map Embedding 457
10.4.5 Working with a Subgraph of G: Computing a Reduced Markov Chain by Stochastic Complementation 459
10.5 Laplacian Eigenmap 460
10.5.1 A First View on the Laplacian Eigenmap 460
10.5.2 An Intuitive Interpretation of the Laplacian Eigenmap 464
10.5.3 A Second View on the Laplacian Eigenmap Based on Graph Cut 466
10.6 A Latent Space Approach to Graph Embedding 466
10.6.1 Definition of the Model 466
10.6.2 Estimation of the Parameters 467
10.7 Basics of Force-Directed Graph Drawing 469
10.7.1 A Spring Network Model 472
10.7.2 An Energy Model Based on Repulsive and Attractive Forces 473
10.7.3 Algorithmic Details 477

Bibliography 479
Index 515

Chapter or section that contains more advanced material that can be skipped.

List of Algorithms

1.1	Expectation-maximization algorithm	*page* 52
1.2	Directed shortest-path distance matrix: Elementwise form	56
1.3	Directed shortest-path distance matrix: Matrix form	57
2.1	Local similarity measures between two nodes	67
2.2	Katz similarity matrix and Leicht's extension	68
2.3	Commute-time and Euclidean commute-time distances	75
2.4	SimRank similarity matrix	85
2.5	Blondel et al. similarity matrix between nodes of two graphs	86
2.6	Exponential diffusion kernel matrix and Laplacian exponential diffusion kernel matrix	88
2.7	Modified regularized Laplacian kernel matrix	91
2.8	Commute-time kernel matrix	93
2.9	Regularized commute-time kernel and random walk with restart similarity	95
2.10	Markov diffusion square distance and kernel matrix	100
3.1	Logarithmic forest distance matrix	104
3.2	Regular bag-of-paths probability matrix	114
3.3	Bag-of-hitting-paths probability matrix	119
3.4	Bag-of-hitting-paths surprisal distance matrix	123
3.5	Bag-of-hitting-paths potential, or free energy, distance matrix	124
3.6	Randomized shortest-path dissimilarity matrix for hitting paths	129
3.7	Bag-of-paths absorption probabilities to a set of absorbing nodes	132
3.8	Bag-of-paths covariance matrix	141
4.1	A naive algorithm for computing Freeman's shortest-path betweenness	148
4.2	Brandes's algorithm for computing Freeman's shortest-path betweenness	152
4.3	Various closeness centrality measures based on node similarity	156
4.4	Random eccentricity of nodes	160
4.5	Current-flow betweenness of nodes	163
4.6	Markov closeness centrality of nodes	167
4.7	Current-flow closeness centrality of nodes	168
4.8	Bag-of-paths betweenness vector	171
4.9	Bag-of-paths group betweenness vector	173
4.10	Randomized shortest-path betweenness vector	179
4.11	Approximating Freeman's shortest-path betweenness vector	180
4.12	Randomized shortest-path net flow betweenness vector	183

4.13	Communicability criticality of nodes	187
4.14	Node criticality with respect to the Kirchhoff index	189
4.15	Spanning tree edge criticality	194
4.16	Bag-of-paths node criticality	198
4.17	Simple potential directed distance edge criticality	199
5.1	Proximity prestige score of nodes	203
5.2	Bonacich's spectral measure of node prestige (eigenvector prestige)	204
5.3	Katz and Hubbell importance scores for nodes	206
5.4	Citation influence score for nodes	208
5.5	Least squares rating or prestige score for nodes	215
5.6	Power method calculation of the PageRank with personalization score	222
5.7	HITS hub and authority scores	226
6.1	A simple regularization framework for labeling nodes	240
6.2	Harmonic function approach for labeling nodes	246
6.3	Random walk with restart approach for labeling nodes	249
6.4	$\mathcal{D}$-walk approach for labeling nodes	252
6.5	Bag-of-paths group betweenness approach for labeling nodes	254
6.6	Laplacian-regularized ridge regression for labeling nodes	257
6.7	Laplacian-regularized kernel ridge regression for labeling nodes	259
6.8	Laplacian-regularized logistic regression for labeling nodes	264
6.9	Structural features associated with Moran's I index	269
6.10	Structural features associated with Geary's c and the contiguity ratio index	270
6.11	Fitting an autologistic model for labeling nodes	273
6.12	Fitting a regularized kernel logistic regression based on two kernel matrices computed from the structure of a graph and the features defined on nodes	274
7.1	Standard distance–based k-means clustering	281
7.2	Simple kernel k-means clustering of nodes	286
7.3	Simple iterative kernel k-means clustering of nodes	289
7.4	Simple label propagation clustering of nodes	296
7.5	Markov cluster algorithm for clustering nodes	298
7.6	Regularized Markov cluster algorithm for clustering nodes	299
7.7	A heuristic procedure greedily improving graph cut: Kernighan-Lin algorithm	305
7.8	m-way ratio cut spectral clustering	320
7.9	m-way normalized cut spectral clustering and Ng et al. spectral clustering	321
7.10	Modularity-based two-way partitioning of a graph	334
7.11	Latent class model for clustering nodes	343
8.1	Smoothed value of a measure defined on nodes through a PageRank-like and a Laplacian regularization algorithm	356
8.2	Bag-of-forests density index of nodes	363
8.3	k-core of a graph	367
8.4	Core decomposition: Computing the core number	369
8.5	Generalized k-core of a graph	371

8.6	A kernel-based Ward hierarchical clustering of nodes	375
8.7	A local optimization procedure for clustering nodes based on modularity: Louvain method	379
8.8	Coarsening step of the Louvain method	380
9.1	Computing coordinates in a common space for nodes of a bipartite graph by using simple correspondence analysis	400
9.2	Reputation scores for a weighted directed bipartite multigraph	410
9.3	Bi-clustering of a bipartite graph by maximizing chi square	421
9.4	Standard weighted k-means clustering	422
9.5	Nonnegative matrix factorization of a bipartite graph, with a multiplicative update procedure	429
9.6	Nonnegative matrix factorization of a bipartite graph, with two alternating least squares procedures	433
10.1	Kernel principal components analysis of a graph	444
10.2	Classical multidimensional scaling	450
10.3	Diffusion map embedding of a graph	455
10.4	Laplacian eigenmap embedding of a graph	464
10.5	Latent social space embedding of a graph	470
10.6	Spring network layout of a graph	474
10.7	(a, r) force-directed layout for a graph	476

List of Symbols and Notation

General

$a, b, c, \ldots, x, y, z$	scalar variables or random variables, depending on the context
$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \ldots, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$	random vectors (bold italic)
α, θ, T, etc.	scalar quantity, parameter, or constant
$\mathcal{S}, \mathcal{T}$, etc.	a set, in calligraphic letters

Special Symbols

$\mathscr{L}$	Lagrange function
$\mathcal{L}$	set of different class labels in clustering, supervised or semi-supervised classification problems
t	time (either discrete – time step – or continuous)
T	temperature of the system
θ	inverse temperature. $\theta = 1/T$
$\mathcal{Z}$	partition function
ϕ	free energy
$\varnothing$	empty set
$\bullet$	a missing value
$\triangleq$	equal to and defined as

Functions and Probability

$\lvert\mathcal{S}\rvert = \#\mathcal{S}$	number of elements, or cardinality, of a set $\mathcal{S}$
δ_{kl}	Kronecker delta whose value is 1 if $k = l$, and 0 otherwise
δ(some predicate)	equals 1 if the predicate is true, and 0 otherwise
$\mathbb{E}_x[f(x)] = \langle f(x) \rangle$	expectation with respect to the random variable x
J(P|Q)	relative entropy or Kullback-Leibler directed divergence between probability distributions P and Q
$\mathcal{L}(\boldsymbol{\theta})$	likelihood function
$l(\boldsymbol{\theta}) = \log \mathcal{L}(\boldsymbol{\theta})$	log-likelihood function
P(some predicate)	probability that the predicate containing random events is true
$\hat{\text{P}}$(some predicate)	estimate of the probability based on empirical data. More generally, any estimate is denoted by a hat

$\mathrm{P}(s = i)$	probability that the discrete random variable s takes value i
$p_{xy}(k, l)$	probability mass function of $\mathrm{P}(x = k, y = l)$, providing the probability that discrete random variables x, y take values k, l
$\mathcal{X} = \mathcal{R}(x), \mathcal{Y} = \mathcal{R}(y)$	set of values, or range, taken by (random) variables x, y

Matrices and Vectors

$\mathbf{M}$	a matrix (uppercase bold)
$\mathbf{M}^{\mathrm{T}}$	transpose of matrix $\mathbf{M}$ containing elements $[\mathbf{M}^{\mathrm{T}}]_{ij} = [\mathbf{M}]_{ji}$
$\mathbf{M}^q$	matrix q-power of $\mathbf{M}$
$\mathbf{M}^{(q)}$	Hadamard (elementwise) q-power of $\mathbf{M}$ containing elements m_{ij}^q
$\mathbf{M} \circ \mathbf{N}$	Hadamard (elementwise) matrix product providing elements $m_{ij} n_{ij}$
$\mathbf{M} \div \mathbf{N}$	Hadamard (elementwise) matrix division providing elements m_{ij}/n_{ij}
$\mathbf{M}^{\div} = \mathbf{M}^{(-1)}$	elementwise reciprocal of $\mathbf{M}$ containing elements $m_{ij}^{\div} = 1/m_{ij}$
$\mathbf{M}^{+}$	Moore-Penrose pseudoinverse of $\mathbf{M}$
$m_{ij} = [\mathbf{M}]_{ij}$	element i, j (in the ith row and the jth column) of matrix $\mathbf{M}$
$m_{ij}^{(q)} = [\mathbf{M}^q]_{ij}$	element i, j of $\mathbf{M}^q$, the matrix q-power of $\mathbf{M}$
$m_{i\bullet} = \sum_j m_{ij}$	sum of the elements of the ith row of $\mathbf{M}$ (row sum)
$m_{\bullet j} = \sum_i m_{ij}$	sum of the elements of the jth column of $\mathbf{M}$ (column sum)
$m_{\bullet\bullet} = \sum_{ij} m_{ij}$	sum over all the elements of the matrix $\mathbf{M}$
$\mathbf{m}_j = \mathbf{m}_j^{\mathrm{c}} = \mathbf{col}_j(\mathbf{M})$	column j of matrix $\mathbf{M}$
$\mathbf{m}_i^{\mathrm{r}} = \mathbf{row}_i(\mathbf{M})$	row i of matrix $\mathbf{M}$ viewed as a column vector
$\mathbf{v}$	a column vector (lowercase upright bold)
$\mathbf{v}^{\mathrm{T}}$	a row vector; the transpose of column vector $\mathbf{v}$
$v_i^c = [\mathbf{v}_c]_i$	ith element of column vector $\mathbf{v}_c$

Special Matrices and Vectors

$\mathbf{0}$	null column vector full of 0s of the appropriate size
$\mathbf{e}$	unit column vector full of 1s of the appropriate size
$\mathbf{e}_i$	ith column of $\mathbf{I}$, containing zero everywhere, except on row i, containing a 1
$\mathbf{E} = \mathbf{e}\mathbf{e}^{\mathrm{T}}$	matrix full of 1s of the appropriate size
$\mathbf{I}$	identity matrix
$\mathbf{H} = \mathbf{I} - \frac{\mathbf{E}}{n}$	$n \times n$ centering matrix; $\mathbf{Hx}$ provides a centered vector whose entries sum to 0
$\mathbf{O}$	matrix full of 0s of the appropriate size

Matrix and Vector Functions

$\|\mathbf{v}\| = \|\mathbf{v}\|_2$	L_2-norm of vector $\mathbf{v}$
$\|\mathbf{v}\|_1$	L_1-norm of vector $\mathbf{v}$
Block($\mathbf{M}$, $\mathcal{A}$, $\mathcal{B}$)	submatrix of $\mathbf{M}$ containing the elements of $\mathbf{M}$ in the intersection between rows in set $i \in \mathcal{A}$ and columns in set $j \in \mathcal{B}$
Block($\mathbf{M}$, $i_1 : i_2$, $j_1 : j_2$)	submatrix of $\mathbf{M}$ containing the elements of $\mathbf{M}$ in the intersection between rows i_1 to i_2 and columns j_1 to j_2
Diag($\mathbf{M}$)	diagonal matrix containing the diagonal of the square matrix $\mathbf{M}$
Diag($\mathbf{v}$) or **Diag**(v_i)	diagonal matrix containing the vector $\mathbf{v}$ on its diagonal
diag($\mathbf{M}$)	column vector containing the diagonal of the square matrix $\mathbf{M}$
exp[$\mathbf{M}$]	elementwise exponential of matrix $\mathbf{M}$
expm[$\mathbf{M}$] $= e^{\mathbf{M}}$	matrix exponential of matrix $\mathbf{M}$
log[$\mathbf{M}$]	elementwise natural logarithm of matrix $\mathbf{M}$
size($\mathbf{v}$)	number of elements of vector $\mathbf{v}$
trace($\mathbf{M}$)	trace of the square matrix $\mathbf{M}$
vec($\mathbf{M}$)	vec operator stacking the columns of $\mathbf{M}$ in a column vector
$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{\mathrm{T}}\mathbf{y}$	inner product between vector $\mathbf{x}$ and vector $\mathbf{y}$
$\mathbf{x} \propto \mathbf{y}$	$\mathbf{x}$ is proportional to $\mathbf{y}$, that is, $\mathbf{x} = \alpha\mathbf{y}$ for some scalar α

Graphs

G, H	a graph or a subgraph; by extension, the set of nodes and edges composing G, H
$G \setminus i$	subgraph of G obtained by deleting node i from G as well as its incident edges
$\mathcal{V}$, $\mathcal{V}(G)$, or simply G	set of nodes (or vertices) of a graph or subgraph G
$v \in \mathcal{V}$, $i \in \mathcal{V}$	a node or vertex belonging to the set $\mathcal{V}$
v_i, node i, or simply i	node whose index is i (ith node of G)
$\mathcal{E}$ or $\mathcal{E}(G)$	set of edges (or arcs, links, connections) of a graph or subgraph G
$\bar{\mathcal{E}}$ or $\bar{\mathcal{E}}(G)$	set of *missing* edges (for which $a_{ij} = 0$) of a graph or subgraph G.
$n = \|\mathcal{V}(G)\| = n(G)$	number of nodes of graph G
$e = \|\mathcal{E}(G)\|$	number of edges of graph G
$w_{ij} > 0$, or $w_{ij}(G)$	weights associated with the edges (i, j) of the graph G; they represent affinities between pairs of nodes
$\mathrm{vol}(G) = \sum_{i,j\in\mathcal{V}} w_{ij} = w_{\bullet\bullet} = d$	volume of the graph G
$i \to j$ or (i, j)	directed edge connecting nodes i and j in a directed graph
$i \nrightarrow j$	a missing directed edge between nodes i and j in a directed graph ($a_{ij} = 0$)
$i \leftrightarrow j$ or (i, j)	undirected edge connecting nodes i and j in an undirected graph

$i \nleftrightarrow j$	a missing undirected edge between nodes i and j in an undirected graph ($a_{ij} = a_{ji} = 0$)
$i \rightsquigarrow j$	a path or walk connecting node i and node j
$\{i \rightarrow j\}$	set of edges linking node i to node j in a multigraph
$\mathcal{N}(k)$	set of neighbors of node k (with k excluded), in an undirected graph; also called the adjacent nodes
$\mathcal{N}^{(t)}(k)$	set of t-steps neighbors of node k in an undirected graph
$\mathcal{R}(i) = \{i \mid i \rightsquigarrow j\}$	region of influence of a node i, that is, the set of nodes that can be reached when starting from i
$\mathcal{P}red(k)$	set of predecessor nodes of node k in a directed graph
$\mathcal{S}ucc(k)$	set of successor nodes of node k in a directed graph
$\mathcal{C}_k$	set of nodes belonging to class, cluster, or community, k
$\ell(i)$	class or cluster label of node i – usually an integer
$\sum_{i \in \mathcal{C}_k}$	sum over all nodes belonging to set $\mathcal{C}_k$
$n_k = \lvert\mathcal{C}_k\rvert$	number of nodes belonging to class, cluster, or community, k
$\text{cut}(\mathcal{C}_k, \mathcal{C}_l) = w(\mathcal{C}_k, \mathcal{C}_l)$	total weight of edges connecting cluster $\mathcal{C}_k$ to $\mathcal{C}_l$, $\sum_{i \in \mathcal{C}_k} \sum_{j \in \mathcal{C}_l} a_{ij}$
$\mathcal{P}_{ij}$	set of all paths starting from node i and ending in node j, including cycles
$\mathcal{P}_{ij}(t)$	set of all t-steps paths of G, starting from node i and ending in node j, including cycles
$\wp$	a particular path of G
$\tilde{c}(\wp)$	total cost along path $\wp$; that is, the sum of the individual costs c_{ij} along path $\wp$
$\tilde{\pi}^{\text{ref}}(\wp)$	likelihood of path $\wp$; that is, the product of the reference transition probabilities, p_{ij}^{ref}, of a natural random walk on G along path $\wp$
Δ_{ij}	dissimilarity (or distance) between node i and node j
$\mathcal{F} = \{\varphi_1, \varphi_2, \dots\}$	set of rooted forests φ_i that can be defined in the graph G
$\text{bet}(k)$; **bet**	betweenness of node k and the betweenness vector
$\text{den}(k)$; $\text{den}(H)$	density measure associated with node k or with subgraph H

Matrices and Vectors Associated with Graphs

$\mathbf{A}$	adjacency matrix of G: $a_{ij} = w_{ij}$ when there is an edge between nodes i and j; $a_{ij} = 0$ otherwise
$\mathbf{C}$	cost matrix associated with a graph G containing transition costs c_{ij}
$\mathbf{D} = \mathbf{Diag}(\mathbf{Ae})$	diagonal degree matrix of undirected graph G containing degrees $a_{i\bullet}$ on its diagonal
$\mathbf{d} = \mathbf{Ae}$	$n \times 1$ degree vector of the undirected graph G containing $a_{i\bullet} = a_{\bullet i}$

$\mathbf{d}_o = \mathbf{A}\mathbf{e}$; $\mathbf{D}_o = \mathbf{Diag}(\mathbf{d}_o)$	outdegree vector and diagonal matrix of the directed graph G; sometimes abbreviated as $\mathbf{d}$, $\mathbf{D}$, for simplicity, when there is no ambiguity
$\mathbf{d}_i = \mathbf{A}^T\mathbf{e}$; $\mathbf{D}_i = \mathbf{Diag}(\mathbf{d}_i)$	indegree vector and diagonal matrix of the directed graph G
$\mathbf{d}(H)$	degree of the nodes of subgraph H, with respect to H
$\mathbf{K}$	$n \times n$ kernel or similarity matrix on the graph G
$\mathbf{L} = \mathbf{D} - \mathbf{A}$	$n \times n$ Laplacian matrix of the undirected graph G
$\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$	$n \times n$ normalized Laplacian matrix of the undirected graph G
$\mathbf{Q}$	$n \times n$ modularity matrix of G
$\mathbf{\Delta}$	$n \times n$ dissimilarity (or distance) matrix on the graph G containing dissimilarities or distances $[\mathbf{\Delta}]_{ij} = \Delta_{ij}$
$\mathbf{\Delta}^{(2)}$	$n \times n$ matrix containing squared dissimilarities, $[\mathbf{\Delta}^{(2)}]_{ij} = \Delta_{ij}^2$.

Markov Chains

$p_{ij} = \mathrm{P}(s(t) = j \mid s(t-1) = i)$	elements of the transition probability matrix in a Markov chain
$\mathbf{P}$	$n \times n$ transition matrix of a time-independent Markov chain; contains the elements $p_{ij} = \mathrm{P}(s(t) = j \mid s(t-1) = i)$
$\mathbf{x}_i(t) = (\mathbf{P}^T)^t\mathbf{e}_i$	$n \times 1$ column vector containing the probability distribution of being in each state j at time step t while starting in state i at $t = 0$, $\mathrm{P}(s(t) = j \mid s(0) = i)$
$\boldsymbol{\pi}$	stationary distribution of a regular Markov chain
$\mathbf{D}_\pi = \mathbf{Diag}(\boldsymbol{\pi})$	diagonal matrix containing the stationary distribution on its diagonal
$\mathcal{A}, \mathcal{T}$	sets of absorbing and transient nodes in an absorbing Markov chain

Preface

The network science field. Since the start of the twenty-first century, *network science*, the field whose main goal is to analyze *network data*, has become more and more popular in various areas of science and technology [47]. This interest has grown in parallel with the popularity of large networks, especially online networks like the World Wide Web, where each node is a web page and hyperlinks can be viewed as edges linking the pages. Another obvious example is online social networks like Facebook, where nodes are persons and links are friendship relations.

Although networks have been studied for years in the fields of *social network analysis*,[1] *operations research*, *graph theory*, and *graph algorithmics*, the wide availability of such network structures on the Internet clearly boosted the field in the late 1990s. Computer scientists, physicists, chemists, economists, statisticians, and applied mathematicians all started to analyze network data.

In computer science, the field was called *link analysis*, while in physics, it was more often known as *network science*, a term that is now used across most disciplines. Roughly speaking, link analysis and network science aim at *analyzing* and *extracting information from complex relational data* (observed relations between entities like people, web pages, etc.) and is considered, in physics, to be a subfield of *complex systems*. The book is dedicated to this subject.

Intended audience. We have written this book for upper-level undergraduate or graduate students, researchers, and practitioners involved, or simply interested, in network data analysis. The book is not, however, intended as an introduction to network science. We assume that the reader has already followed an introductory course on graphs and networks (e.g., [47, 258, 468, 522, 608, 781] or the chapters dedicated to network data in [836]) as well as elementary courses in computer science, probability, statistics, and matrix theory. We nevertheless start with an introductory chapter, "Preliminaries and Notation," summarizing the necessary slightly more advanced material and the notation.

While the material of the book is oriented toward computer scientists and engineers, we think that it should also attract students, researchers, and practitioners in other fields having an interest in network science. The material can easily be followed by other scientists in many application areas, provided they have the basic background knowledge outlined previously.

[1] See, e.g., the scientific journal *Social Networks*, whose volume 1 appeared in 1979, or the book of Wasserman and Faust [804].

Content of the book. This book focuses on *static network data analysis* from different perspectives. Initially, our intention was to cover dynamic models as well (models of the evolution of networks and models of spread of information within a network), but we quickly found that this goal was too ambitious.[2] We therefore concentrate our effort on the extraction of useful information from static networks, adopting a *computer science oriented* and *engineering perspective* (pattern recognition, data mining, machine learning). But this focus is still very broad; we have omitted several interesting techniques, mainly because of constraints on time and space.

Each chapter covers models and algorithms used for tackling a family of *functional tasks*, such as "Identifying prestigious nodes," "Detecting the most central nodes," "Predicting information associated to the nodes," and "Finding dense communities." Each method is described in depth in a separate section that is – as far as possible – *self-contained*, so that each can be read independently. Some definitions and notation are therefore repeated, with the drawback that an assiduous reader will notice some redundancies. Moreover, important formulas – either for understanding the concept or for computing the quantity – are displayed in gray boxes. For each described method or model, we provide an algorithm in pseudocode clarifying the procedure to be followed for applying the method.

The algorithms described in the different chapters are carefully selected from different disciplines (computer science, physics, chemistry, social science, applied statistics, applied mathematics, etc.). The selection, of course, reflects our personal research preferences, but our choice is also clearly biased in favor of (enumerated in a random order)

- classical, *well-established algorithms* – not necessarily very popular in the field of computer science or physics – that can be applied to network data; examples are correspondence analysis, latent class models, and multidimensional scaling
- *principled methods* grounded on clear arguments, such as optimality principles
- *linear algebraic methods* as well as *linear models* relying on sound computational procedures like solving systems of linear equations, matrix inversion, and matrix factorization
- *random walk–based methods*, as well as their *current flow* interpretation, relying on clear, intuitive arguments and interpretations
- methods applying *mathematical* or *statistical techniques* that are sound and interesting by themselves (least squares, maximum likelihood, expectation-maximization, etc.)
- algorithms *scaling at least to medium-size graphs* (at least thousands of nodes)

However, concerning the last point, many of these techniques scale in $O(n^3)$, where n is the number of nodes, which prevents their straightforward application to large graphs. Fortunately, depending on the situation, some computational tricks can reduce this to O(number of edges), which allows us to process large sparse networks.

The methods described form a toolbox of existing techniques and models that can be tested when tackling a particular problem. Intriguingly, many of these methods use the Laplacian matrix of the graph, which plays a key role.

[2] For the interested reader, two chapters of Barabasi's book are dedicated to these subjects [47].

As a by-product, interesting results concerning the random walk on a graph and its relationship to electrical circuits are studied, such as the random walk interpretation of electrical current flow or the expression of the absorption probabilities, the expected number of visits, and the expected first passage time in function of the Laplacian matrix.

We also introduce *more advanced material*[3] that can be skipped during reading without any consequence. These chapters and sections are marked by an asterisk (∗) and describe extensions of the more fundamental methods, mainly developed by the authors, but not necessarily so.

Algorithms and code. For each method introduced, we describe an algorithm in pseudocode. Several of these algorithms use standard matrix operations. We therefore assume that a high-level language providing matrix computation facilities (e.g., Matlab, Octave, Python, Scilab, R, Stata, Maple, Mathematica) is used for implementing the algorithms. Note that algorithms are provided for educational purposes and are therefore not optimized.

The Matlab/Octave code of many of the algorithms will be made available gradually on the personal pages of the authors and made accessible from the Cambridge University Press web page for the book (http://www.cambridge.org/9781107125773). We chose Matlab because (i) there is an open source equivalent (Octave), (ii) it handles sparse matrices, and (iii) it is a high-level, compact, user-friendly language providing all the necessary matrix operations.

Acknowledgments. We express our gratitude to our families for their support during the long period of writing this book. We also thank our master's and PhD students, especially Kevin Françoisse, Silvia Garcia-Diez, Ilkka Kivimaki, Bertrand Lebichot, Sandrine Brognaux, Kristel Vignery, Robin Devooght, Amin Mantrach, Virginie Vandenbulcke, Felix Sommer, Mathieu Senelle, Luh Yen, Youssef Achbany, and Pascal Francq, for the interesting discussions, for implementing some of the algorithms described in the book, for agreeing to provide the code, and for their remarks.

Marco Saerens also thanks the Royal Library of Brussels (KBR) as well as the IRIDIA laboratory of the Université Libre de Bruxelles for hosting him during the writing periods.

Masashi Shimbo was partially supported by JSPS Kakenhi grant 24300057. Marco Saerens and his researchers were partially supported by WIST projects funded by the Walloon region and by InnovIris projects funded by the Brussels region. Some visiting travel and accommodation expenses were funded by the Louvain School of Management of the Université catholique de Louvain and by Nara Institute of Science and Technology. Some of our researchers were funded by the Belgian *Fonds de la Recherche Scientifique* (FNRS).

We also acknowledge Dr. Kivimaki, Dr. Traag, Prof. Chebotarev, Dr. Alamgir, Prof. Nadler, Prof. Von Luxburg, Prof. Guillaume, Prof. Delvenne, and Prof. Noack for their remarks.

[3] About 20 percent of content.