

Mining of Massive Datasets

Anand Rajaraman

Jure Leskovec
Stanford Univ.

Jeffrey D. Ullman
Stanford Univ.

Preface

This book evolved from material developed over several years by Anand Rajaraman and Jeff Ullman for a one-quarter course at Stanford. The course CS345A, titled “Web Mining,” was designed as an advanced graduate course, although it has become accessible and interesting to advanced undergraduates. When Jure Leskovec joined the Stanford faculty, we reorganized the material considerably. He introduced a new course CS224W on network analysis and added material to CS345A, which was renumbered CS246. The three authors also introduced a large-scale data-mining project course, CS341. The book now contains material taught in all three courses.

What the Book Is About

At the highest level of description, this book is about data mining. However, it focuses on data mining of very large amounts of data, that is, data so large it does not fit in main memory. Because of the emphasis on size, many of our examples are about the Web or data derived from the Web. Further, the book takes an algorithmic point of view: data mining is about applying algorithms to data, rather than using data to “train” a machine-learning engine of some sort. The principal topics covered are:

1. Distributed file systems and map-reduce as a tool for creating parallel algorithms that succeed on very large amounts of data.
2. Similarity search, including the key techniques of minhashing and locality-sensitive hashing.
3. Data-stream processing and specialized algorithms for dealing with data that arrives so fast it must be processed immediately or lost.
4. The technology of search engines, including Google’s PageRank, link-spam detection, and the hubs-and-authorities approach.
5. Frequent-itemset mining, including association rules, market-baskets, the A-Priori Algorithm and its improvements.
6. Algorithms for clustering very large, high-dimensional datasets.

7. Two key problems for Web applications: managing advertising and recommendation systems.
8. Algorithms for analyzing and mining the structure of very large graphs, especially social-network graphs.

Prerequisites

To appreciate fully the material in this book, we recommend the following prerequisites:

1. An introduction to database systems, covering SQL and related programming systems.
2. A sophomore-level course in data structures, algorithms, and discrete math.
3. A sophomore-level course in software systems, software engineering, and programming languages.

Exercises

The book contains extensive exercises, with some for almost every section. We indicate harder exercises or parts of exercises with an exclamation point. The hardest exercises have a double exclamation point.

Support on the Web

You can find materials from past offerings of CS345A at:

<http://infolab.stanford.edu/~ullman/mining/mining.html>

There, you will find slides, homework assignments, project requirements, and in some cases, exams.

Acknowledgements

Cover art is by Scott Ullman.

We would like to thank Foto Afrati and Arun Marathe for critical readings of the draft of this manuscript.

Errors were also reported by Apoorv Agarwal, Susan Biancani, Leland Chen, Shrey Gupta, Xie Ke, Haewoon Kwak, Ellis Lau, Ethan Lozano, Justin Meyer, Brad Penoff, Philips Kokoh Prasetyo, Angad Singh, Sandeep Sripada, Dennis

Sidharta, Mark Storus, Roshan Sumbaly, Tim Triche Jr., and Robert West.
The remaining errors are ours, of course.

A. R.
J. L.
J. D. U.
Palo Alto, CA
July, 2012

Contents

1	Data Mining	1
1.1	What is Data Mining?	1
1.1.1	Statistical Modeling	1
1.1.2	Machine Learning	2
1.1.3	Computational Approaches to Modeling	2
1.1.4	Summarization	3
1.1.5	Feature Extraction	4
1.2	Statistical Limits on Data Mining	4
1.2.1	Total Information Awareness	5
1.2.2	Bonferroni's Principle	5
1.2.3	An Example of Bonferroni's Principle	6
1.2.4	Exercises for Section 1.2	7
1.3	Things Useful to Know	7
1.3.1	Importance of Words in Documents	7
1.3.2	Hash Functions	9
1.3.3	Indexes	10
1.3.4	Secondary Storage	11
1.3.5	The Base of Natural Logarithms	12
1.3.6	Power Laws	13
1.3.7	Exercises for Section 1.3	15
1.4	Outline of the Book	15
1.5	Summary of Chapter 1	17
1.6	References for Chapter 1	17
2	Large-Scale File Systems and Map-Reduce	19
2.1	Distributed File Systems	20
2.1.1	Physical Organization of Compute Nodes	20
2.1.2	Large-Scale File-System Organization	21
2.2	Map-Reduce	22
2.2.1	The Map Tasks	23
2.2.2	Grouping and Aggregation	24
2.2.3	The Reduce Tasks	24
2.2.4	Combiners	25

2.2.5	Details of Map-Reduce Execution	25
2.2.6	Coping With Node Failures	26
2.3	Algorithms Using Map-Reduce	27
2.3.1	Matrix-Vector Multiplication by Map-Reduce	27
2.3.2	If the Vector v Cannot Fit in Main Memory	28
2.3.3	Relational-Algebra Operations	29
2.3.4	Computing Selections by Map-Reduce	32
2.3.5	Computing Projections by Map-Reduce	32
2.3.6	Union, Intersection, and Difference by Map-Reduce	33
2.3.7	Computing Natural Join by Map-Reduce	34
2.3.8	Generalizing the Join Algorithm	34
2.3.9	Grouping and Aggregation by Map-Reduce	35
2.3.10	Matrix Multiplication	35
2.3.11	Matrix Multiplication with One Map-Reduce Step	36
2.3.12	Exercises for Section 2.3	37
2.4	Extensions to Map-Reduce	38
2.4.1	Workflow Systems	38
2.4.2	Recursive Extensions to Map-Reduce	40
2.4.3	Pregel	42
2.4.4	Exercises for Section 2.4	43
2.5	Efficiency of Cluster-Computing Algorithms	43
2.5.1	The Communication-Cost Model for Cluster Computing	44
2.5.2	Elapsed Communication Cost	46
2.5.3	Multiway Joins	46
2.5.4	Exercises for Section 2.5	49
2.6	Summary of Chapter 2	51
2.7	References for Chapter 2	52
3	Finding Similar Items	55
3.1	Applications of Near-Neighbor Search	55
3.1.1	Jaccard Similarity of Sets	56
3.1.2	Similarity of Documents	56
3.1.3	Collaborative Filtering as a Similar-Sets Problem	57
3.1.4	Exercises for Section 3.1	59
3.2	Shingling of Documents	59
3.2.1	k -Shingles	59
3.2.2	Choosing the Shingle Size	60
3.2.3	Hashing Shingles	60
3.2.4	Shingles Built from Words	61
3.2.5	Exercises for Section 3.2	62
3.3	Similarity-Preserving Summaries of Sets	62
3.3.1	Matrix Representation of Sets	62
3.3.2	Minhashing	63
3.3.3	Minhashing and Jaccard Similarity	64

3.3.4	Minhash Signatures	65
3.3.5	Computing Minhash Signatures	65
3.3.6	Exercises for Section 3.3	67
3.4	Locality-Sensitive Hashing for Documents	69
3.4.1	LSH for Minhash Signatures	69
3.4.2	Analysis of the Banding Technique	71
3.4.3	Combining the Techniques	72
3.4.4	Exercises for Section 3.4	73
3.5	Distance Measures	74
3.5.1	Definition of a Distance Measure	74
3.5.2	Euclidean Distances	74
3.5.3	Jaccard Distance	75
3.5.4	Cosine Distance	76
3.5.5	Edit Distance	77
3.5.6	Hamming Distance	78
3.5.7	Exercises for Section 3.5	79
3.6	The Theory of Locality-Sensitive Functions	80
3.6.1	Locality-Sensitive Functions	81
3.6.2	Locality-Sensitive Families for Jaccard Distance	82
3.6.3	Amplifying a Locality-Sensitive Family	83
3.6.4	Exercises for Section 3.6	85
3.7	LSH Families for Other Distance Measures	86
3.7.1	LSH Families for Hamming Distance	86
3.7.2	Random Hyperplanes and the Cosine Distance	86
3.7.3	Sketches	88
3.7.4	LSH Families for Euclidean Distance	89
3.7.5	More LSH Families for Euclidean Spaces	90
3.7.6	Exercises for Section 3.7	90
3.8	Applications of Locality-Sensitive Hashing	91
3.8.1	Entity Resolution	92
3.8.2	An Entity-Resolution Example	92
3.8.3	Validating Record Matches	93
3.8.4	Matching Fingerprints	94
3.8.5	A LSH Family for Fingerprint Matching	95
3.8.6	Similar News Articles	97
3.8.7	Exercises for Section 3.8	98
3.9	Methods for High Degrees of Similarity	99
3.9.1	Finding Identical Items	99
3.9.2	Representing Sets as Strings	100
3.9.3	Length-Based Filtering	100
3.9.4	Prefix Indexing	101
3.9.5	Using Position Information	102
3.9.6	Using Position and Length in Indexes	104
3.9.7	Exercises for Section 3.9	106
3.10	Summary of Chapter 3	107

3.11	References for Chapter 3	110
4	Mining Data Streams	113
4.1	The Stream Data Model	113
4.1.1	A Data-Stream-Management System	114
4.1.2	Examples of Stream Sources	115
4.1.3	Stream Queries	116
4.1.4	Issues in Stream Processing	117
4.2	Sampling Data in a Stream	118
4.2.1	A Motivating Example	118
4.2.2	Obtaining a Representative Sample	119
4.2.3	The General Sampling Problem	119
4.2.4	Varying the Sample Size	120
4.2.5	Exercises for Section 4.2	120
4.3	Filtering Streams	121
4.3.1	A Motivating Example	121
4.3.2	The Bloom Filter	122
4.3.3	Analysis of Bloom Filtering	122
4.3.4	Exercises for Section 4.3	123
4.4	Counting Distinct Elements in a Stream	124
4.4.1	The Count-Distinct Problem	124
4.4.2	The Flajolet-Martin Algorithm	125
4.4.3	Combining Estimates	126
4.4.4	Space Requirements	126
4.4.5	Exercises for Section 4.4	127
4.5	Estimating Moments	127
4.5.1	Definition of Moments	127
4.5.2	The Alon-Matias-Szegedy Algorithm for Second Moments	128
4.5.3	Why the Alon-Matias-Szegedy Algorithm Works	129
4.5.4	Higher-Order Moments	130
4.5.5	Dealing With Infinite Streams	130
4.5.6	Exercises for Section 4.5	131
4.6	Counting Ones in a Window	132
4.6.1	The Cost of Exact Counts	133
4.6.2	The Datar-Gionis-Indyk-Motwani Algorithm	133
4.6.3	Storage Requirements for the DGIM Algorithm	135
4.6.4	Query Answering in the DGIM Algorithm	135
4.6.5	Maintaining the DGIM Conditions	136
4.6.6	Reducing the Error	137
4.6.7	Extensions to the Counting of Ones	138
4.6.8	Exercises for Section 4.6	139
4.7	Decaying Windows	139
4.7.1	The Problem of Most-Common Elements	139
4.7.2	Definition of the Decaying Window	140

4.7.3	Finding the Most Popular Elements	141
4.8	Summary of Chapter 4	142
4.9	References for Chapter 4	143
5	Link Analysis	145
5.1	PageRank	145
5.1.1	Early Search Engines and Term Spam	146
5.1.2	Definition of PageRank	147
5.1.3	Structure of the Web	151
5.1.4	Avoiding Dead Ends	152
5.1.5	Spider Traps and Taxation	155
5.1.6	Using PageRank in a Search Engine	157
5.1.7	Exercises for Section 5.1	157
5.2	Efficient Computation of PageRank	159
5.2.1	Representing Transition Matrices	160
5.2.2	PageRank Iteration Using Map-Reduce	161
5.2.3	Use of Combiners to Consolidate the Result Vector	161
5.2.4	Representing Blocks of the Transition Matrix	162
5.2.5	Other Efficient Approaches to PageRank Iteration	163
5.2.6	Exercises for Section 5.2	165
5.3	Topic-Sensitive PageRank	165
5.3.1	Motivation for Topic-Sensitive Page Rank	165
5.3.2	Biased Random Walks	166
5.3.3	Using Topic-Sensitive PageRank	167
5.3.4	Inferring Topics from Words	168
5.3.5	Exercises for Section 5.3	169
5.4	Link Spam	169
5.4.1	Architecture of a Spam Farm	169
5.4.2	Analysis of a Spam Farm	171
5.4.3	Combating Link Spam	172
5.4.4	TrustRank	172
5.4.5	Spam Mass	173
5.4.6	Exercises for Section 5.4	173
5.5	Hubs and Authorities	174
5.5.1	The Intuition Behind HITS	174
5.5.2	Formalizing Hubbiness and Authority	175
5.5.3	Exercises for Section 5.5	178
5.6	Summary of Chapter 5	178
5.7	References for Chapter 5	182
6	Frequent Itemsets	183
6.1	The Market-Basket Model	184
6.1.1	Definition of Frequent Itemsets	184
6.1.2	Applications of Frequent Itemsets	185
6.1.3	Association Rules	187

6.1.4	Finding Association Rules with High Confidence	189
6.1.5	Exercises for Section 6.1	189
6.2	Market Baskets and the A-Priori Algorithm	190
6.2.1	Representation of Market-Basket Data	191
6.2.2	Use of Main Memory for Itemset Counting	192
6.2.3	Monotonicity of Itemsets	194
6.2.4	Tyranny of Counting Pairs	194
6.2.5	The A-Priori Algorithm	195
6.2.6	A-Priori for All Frequent Itemsets	197
6.2.7	Exercises for Section 6.2	198
6.3	Handling Larger Datasets in Main Memory	200
6.3.1	The Algorithm of Park, Chen, and Yu	200
6.3.2	The Multistage Algorithm	202
6.3.3	The Multihash Algorithm	204
6.3.4	Exercises for Section 6.3	206
6.4	Limited-Pass Algorithms	208
6.4.1	The Simple, Randomized Algorithm	208
6.4.2	Avoiding Errors in Sampling Algorithms	209
6.4.3	The Algorithm of Savasere, Omiecinski, and Navathe	210
6.4.4	The SON Algorithm and Map-Reduce	210
6.4.5	Toivonen's Algorithm	211
6.4.6	Why Toivonen's Algorithm Works	213
6.4.7	Exercises for Section 6.4	213
6.5	Counting Frequent Items in a Stream	214
6.5.1	Sampling Methods for Streams	214
6.5.2	Frequent Itemsets in Decaying Windows	215
6.5.3	Hybrid Methods	216
6.5.4	Exercises for Section 6.5	217
6.6	Summary of Chapter 6	217
6.7	References for Chapter 6	220
7	Clustering	221
7.1	Introduction to Clustering Techniques	221
7.1.1	Points, Spaces, and Distances	221
7.1.2	Clustering Strategies	223
7.1.3	The Curse of Dimensionality	224
7.1.4	Exercises for Section 7.1	225
7.2	Hierarchical Clustering	225
7.2.1	Hierarchical Clustering in a Euclidean Space	226
7.2.2	Efficiency of Hierarchical Clustering	228
7.2.3	Alternative Rules for Controlling Hierarchical Clustering	229
7.2.4	Hierarchical Clustering in Non-Euclidean Spaces	232
7.2.5	Exercises for Section 7.2	233

7.3	K-means Algorithms	234
7.3.1	K-Means Basics	235
7.3.2	Initializing Clusters for K-Means	235
7.3.3	Picking the Right Value of k	236
7.3.4	The Algorithm of Bradley, Fayyad, and Reina	237
7.3.5	Processing Data in the BFR Algorithm	239
7.3.6	Exercises for Section 7.3	242
7.4	The CURE Algorithm	242
7.4.1	Initialization in CURE	243
7.4.2	Completion of the CURE Algorithm	244
7.4.3	Exercises for Section 7.4	245
7.5	Clustering in Non-Euclidean Spaces	246
7.5.1	Representing Clusters in the GRGPF Algorithm	246
7.5.2	Initializing the Cluster Tree	247
7.5.3	Adding Points in the GRGPF Algorithm	248
7.5.4	Splitting and Merging Clusters	249
7.5.5	Exercises for Section 7.5	250
7.6	Clustering for Streams and Parallelism	250
7.6.1	The Stream-Computing Model	251
7.6.2	A Stream-Clustering Algorithm	251
7.6.3	Initializing Buckets	252
7.6.4	Merging Buckets	252
7.6.5	Answering Queries	255
7.6.6	Clustering in a Parallel Environment	255
7.6.7	Exercises for Section 7.6	256
7.7	Summary of Chapter 7	256
7.8	References for Chapter 7	260
8	Advertising on the Web	261
8.1	Issues in On-Line Advertising	261
8.1.1	Advertising Opportunities	261
8.1.2	Direct Placement of Ads	262
8.1.3	Issues for Display Ads	263
8.2	On-Line Algorithms	264
8.2.1	On-Line and Off-Line Algorithms	264
8.2.2	Greedy Algorithms	265
8.2.3	The Competitive Ratio	266
8.2.4	Exercises for Section 8.2	266
8.3	The Matching Problem	267
8.3.1	Matches and Perfect Matches	267
8.3.2	The Greedy Algorithm for Maximal Matching	268
8.3.3	Competitive Ratio for Greedy Matching	269
8.3.4	Exercises for Section 8.3	270
8.4	The Adwords Problem	270
8.4.1	History of Search Advertising	271

8.4.2	Definition of the Adwords Problem	271
8.4.3	The Greedy Approach to the Adwords Problem	272
8.4.4	The Balance Algorithm	273
8.4.5	A Lower Bound on Competitive Ratio for Balance	274
8.4.6	The Balance Algorithm with Many Bidders	276
8.4.7	The Generalized Balance Algorithm	277
8.4.8	Final Observations About the Adwords Problem	278
8.4.9	Exercises for Section 8.4	279
8.5	Adwords Implementation	279
8.5.1	Matching Bids and Search Queries	280
8.5.2	More Complex Matching Problems	280
8.5.3	A Matching Algorithm for Documents and Bids	281
8.6	Summary of Chapter 8	283
8.7	References for Chapter 8	285
9	Recommendation Systems	287
9.1	A Model for Recommendation Systems	287
9.1.1	The Utility Matrix	288
9.1.2	The Long Tail	289
9.1.3	Applications of Recommendation Systems	289
9.1.4	Populating the Utility Matrix	291
9.2	Content-Based Recommendations	292
9.2.1	Item Profiles	292
9.2.2	Discovering Features of Documents	293
9.2.3	Obtaining Item Features From Tags	294
9.2.4	Representing Item Profiles	295
9.2.5	User Profiles	296
9.2.6	Recommending Items to Users Based on Content	297
9.2.7	Classification Algorithms	298
9.2.8	Exercises for Section 9.2	300
9.3	Collaborative Filtering	301
9.3.1	Measuring Similarity	301
9.3.2	The Duality of Similarity	304
9.3.3	Clustering Users and Items	305
9.3.4	Exercises for Section 9.3	307
9.4	Dimensionality Reduction	308
9.4.1	UV-Decomposition	308
9.4.2	Root-Mean-Square Error	309
9.4.3	Incremental Computation of a UV-Decomposition	309
9.4.4	Optimizing an Arbitrary Element	312
9.4.5	Building a Complete UV-Decomposition Algorithm	314
9.4.6	Exercises for Section 9.4	316
9.5	The NetFlix Challenge	316
9.6	Summary of Chapter 9	318
9.7	References for Chapter 9	320

10 Mining Social-Network Graphs	321
10.1 Social Networks as Graphs	321
10.1.1 What is a Social Network?	322
10.1.2 Social Networks as Graphs	322
10.1.3 Varieties of Social Networks	324
10.1.4 Graphs With Several Node Types	325
10.1.5 Exercises for Section 10.1	326
10.2 Clustering of Social-Network Graphs	326
10.2.1 Distance Measures for Social-Network Graphs	327
10.2.2 Applying Standard Clustering Methods	327
10.2.3 Betweenness	329
10.2.4 The Girvan-Newman Algorithm	329
10.2.5 Using Betweenness to Find Communities	332
10.2.6 Exercises for Section 10.2	333
10.3 Direct Discovery of Communities	334
10.3.1 Finding Cliques	335
10.3.2 Complete Bipartite Graphs	335
10.3.3 Finding Complete Bipartite Subgraphs	336
10.3.4 Why Complete Bipartite Graphs Must Exist	337
10.3.5 Exercises for Section 10.3	339
10.4 Partitioning of Graphs	339
10.4.1 What Makes a Good Partition?	340
10.4.2 Normalized Cuts	340
10.4.3 Some Matrices That Describe Graphs	341
10.4.4 Eigenvalues of the Laplacian Matrix	342
10.4.5 Alternative Partitioning Methods	345
10.4.6 Exercises for Section 10.4	346
10.5 Simrank	346
10.5.1 Random Walkers on a Social Graph	347
10.5.2 Random Walks with Restart	348
10.5.3 Exercises for Section 10.5	350
10.6 Counting Triangles	351
10.6.1 Why Count Triangles?	351
10.6.2 An Algorithm for Finding Triangles	351
10.6.3 Optimality of the Triangle-Finding Algorithm	353
10.6.4 Finding Triangles Using Map-Reduce	353
10.6.5 Using Fewer Reduce Tasks	355
10.6.6 Exercises for Section 10.6	355
10.7 Neighborhood Properties of Graphs	357
10.7.1 Directed Graphs and Neighborhoods	357
10.7.2 The Diameter of a Graph	358
10.7.3 Transitive Closure and Reachability	359
10.7.4 Transitive Closure Via Map-Reduce	360
10.7.5 Smart Transitive Closure	362
10.7.6 Transitive Closure by Graph Reduction	364

10.7.7 Approximating the Sizes of Neighborhoods	366
10.7.8 Exercises for Section 10.7	367
10.8 Summary of Chapter 10	368
10.9 References for Chapter 10	371