Mining of Massive Datasets

Jure Leskovec Stanford Univ.

Anand Rajaraman Milliway Labs

Jeffrey D. Ullman Stanford Univ.

Copyright © 2010, 2011, 2012, 2013, 2014 An
and Rajaraman, Jure Leskovec, and Jeffrey D. Ullman

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.

iv PREFACE

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.
- Techniques for obtaining the important properties of a large dataset by dimensionality reduction, including singular-value decomposition and latent semantic indexing.
- 10. Machine-learning algorithms that can be applied to very large data, such as perceptrons, support-vector machines, and gradient descent.

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

Go to http://www.mmds.org for slides, homework assignments, project requirements, and exams from courses related to this book.

Gradiance Automated Homework

There are automated exercises based on this book, using the Gradiance root-question technology, available at www.gradiance.com/services. Students may enter a public class by creating an account at that site and entering the class with code 1EDD8A1D. Instructors may use the site by making an account there

PREFACE v

and then emailing support at gradiance dot com with their login name, the name of their school, and a request to use the MMDS materials.

Acknowledgements

Cover art is by Scott Ullman.

We would like to thank Foto Afrati, Arun Marathe, and Rok Sosic for critical readings of a draft of this manuscript.

Errors were also reported by Rajiv Abraham, Apoorv Agarwal, Aris Anagnostopoulos, Atilla Soner Balkir, Arnaud Belletoile, Robin Bennett, Susan Biancani, Amitabh Chaudhary, Leland Chen, Hua Feng, Marcus Gemeinder, Anastasios Gounaris, Clark Grubb, Shrey Gupta, Waleed Hameid, Saman Haratizadeh, Przemyslaw Horban, Jeff Hwang, Rafi Kamal, Lachlan Kang, Ed Knorr, Haewoon Kwak, Ellis Lau, Greg Lee, David Z. Liu, Ethan Lozano, Yunan Luo, Michael Mahoney, Justin Meyer, Bryant Moscon, Brad Penoff, John Phillips, Philips Kokoh Prasetyo, Qi Ge, Harizo Rajaona, Rich Seiter, Hitesh Shetty, Angad Singh, Sandeep Sripada, Dennis Sidharta, Krzysztof Stencel, Mark Storus, Roshan Sumbaly, Zack Taylor, Tim Triche Jr., Wang Bin, Weng Zhen-Bin, Robert West, Oscar Wu, Xie Ke, Nicolas Zhao, and Zhou Jingbo, The remaining errors are ours, of course.

J. L.A. R.J. D. U.Palo Alto, CAMarch, 2014

vi PREFACE

Contents

1	Data Mining				
	1.1	1.1 What is Data Mining?			
		1.1.1	Statistical Modeling		
		1.1.2	Machine Learning		
		1.1.3	Computational Approaches to Modeling 2		
		1.1.4	Summarization		
		1.1.5	Feature Extraction		
	1.2	Statis	tical Limits on Data Mining		
		1.2.1	Total Information Awareness		
		1.2.2	Bonferroni's Principle		
		1.2.3	An Example of Bonferroni's Principle 6		
		1.2.4	Exercises for Section 1.2		
	1.3	Thing	s Useful to Know		
		1.3.1	Importance of Words in Documents		
		1.3.2	Hash Functions		
		1.3.3	Indexes		
		1.3.4	Secondary Storage		
		1.3.5	The Base of Natural Logarithms		
		1.3.6	Power Laws		
		1.3.7	Exercises for Section 1.3		
	1.4	Outlin	ne of the Book		
	1.5	Summ	nary of Chapter 1		
	1.6		ences for Chapter 1		
2	Mai	nRedu	ce and the New Software Stack 21		
_	2.1	_	buted File Systems		
	2.1	2.1.1	Physical Organization of Compute Nodes		
		2.1.2	Large-Scale File-System Organization		
	2.2		deduce		
	2.2	2.2.1	The Map Tasks		
		2.2.1 $2.2.2$	Grouping by Key		
		2.2.3	The Reduce Tasks		
		2.2.3 $2.2.4$	Combiners		
		4.4.4	Communers		

viii CONTENTS

		2.2.5	Details of MapReduce Execution	28
		2.2.6	Coping With Node Failures	29
		2.2.7	Exercises for Section 2.2	30
	2.3	Algoria	thms Using MapReduce	30
		2.3.1	Matrix-Vector Multiplication by MapReduce	31
		2.3.2	If the Vector v Cannot Fit in Main Memory	31
		2.3.3	Relational-Algebra Operations	32
		2.3.4	Computing Selections by MapReduce	35
		2.3.5	Computing Projections by MapReduce	36
		2.3.6	Union, Intersection, and Difference by MapReduce	36
		2.3.7	Computing Natural Join by MapReduce	37
		2.3.8	Grouping and Aggregation by MapReduce	37
		2.3.9	Matrix Multiplication	38
		2.3.10	Matrix Multiplication with One MapReduce Step	39
		2.3.11	Exercises for Section 2.3	40
	2.4	Extens	sions to MapReduce	41
		2.4.1	Workflow Systems	41
		2.4.2	Recursive Extensions to MapReduce	42
		2.4.3	Pregel	45
		2.4.4	Exercises for Section 2.4	46
	2.5	The C	ommunication Cost Model	46
		2.5.1	Communication-Cost for Task Networks	47
		2.5.2	Wall-Clock Time	49
		2.5.3	Multiway Joins	49
		2.5.4	Exercises for Section 2.5	52
	2.6	Compl	lexity Theory for MapReduce	54
		2.6.1	Reducer Size and Replication Rate	54
		2.6.2	An Example: Similarity Joins	55
		2.6.3	A Graph Model for MapReduce Problems	57
		2.6.4	Mapping Schemas	58
		2.6.5	When Not All Inputs Are Present	60
		2.6.6	Lower Bounds on Replication Rate	61
		2.6.7	Case Study: Matrix Multiplication	62
		2.6.8	Exercises for Section 2.6	66
	2.7	Summ	ary of Chapter 2	67
	2.8		nces for Chapter 2	69
3	Fine	ding Si	milar Items	73
	3.1	Applic	eations of Near-Neighbor Search	73
		3.1.1	Jaccard Similarity of Sets	74
		3.1.2	Similarity of Documents	74
		3.1.3	Collaborative Filtering as a Similar-Sets Problem	75
		3.1.4	Exercises for Section 3.1	77
	3.2	Shingli	ing of Documents	77
		3.2.1	k-Shingles	77

CONTENTS ix

	3.2.2	Choosing the Shingle Size
	3.2.3	Hashing Shingles
	3.2.4	Shingles Built from Words
	3.2.5	Exercises for Section 3.2
3.3	Simila	rity-Preserving Summaries of Sets 80
	3.3.1	Matrix Representation of Sets 81
	3.3.2	Minhashing
	3.3.3	Minhashing and Jaccard Similarity 82
	3.3.4	Minhash Signatures
	3.3.5	Computing Minhash Signatures 83
	3.3.6	Exercises for Section 3.3
3.4	Locali	ty-Sensitive Hashing for Documents 87
	3.4.1	LSH for Minhash Signatures
	3.4.2	Analysis of the Banding Technique 89
	3.4.3	Combining the Techniques
	3.4.4	Exercises for Section 3.4
3.5	Distan	ce Measures
	3.5.1	Definition of a Distance Measure 92
	3.5.2	Euclidean Distances
	3.5.3	Jaccard Distance
	3.5.4	Cosine Distance
	3.5.5	Edit Distance
	3.5.6	Hamming Distance
	3.5.7	Exercises for Section 3.5
3.6	The T	heory of Locality-Sensitive Functions
	3.6.1	Locality-Sensitive Functions
	3.6.2	Locality-Sensitive Families for Jaccard Distance 100
	3.6.3	Amplifying a Locality-Sensitive Family 101
	3.6.4	Exercises for Section 3.6
3.7	LSH F	'amilies for Other Distance Measures
	3.7.1	LSH Families for Hamming Distance 104
	3.7.2	Random Hyperplanes and the Cosine Distance 105
	3.7.3	Sketches
	3.7.4	LSH Families for Euclidean Distance 107
	3.7.5	More LSH Families for Euclidean Spaces 108
	3.7.6	Exercises for Section 3.7
3.8		eations of Locality-Sensitive Hashing
	3.8.1	Entity Resolution
	3.8.2	An Entity-Resolution Example
	3.8.3	Validating Record Matches
	3.8.4	Matching Fingerprints
	3.8.5	A LSH Family for Fingerprint Matching
	3.8.6	Similar News Articles
	3.8.7	Exercises for Section 3.8
3.9	Metho	ds for High Degrees of Similarity

x CONTENTS

		3.9.1	Finding Identical Items	118
		3.9.2	Representing Sets as Strings	
		3.9.3	Length-Based Filtering	
		3.9.4	Prefix Indexing	
		3.9.5	Using Position Information	121
		3.9.6	Using Position and Length in Indexes	122
		3.9.7	Exercises for Section 3.9	125
	3.10	Summ	nary of Chapter 3	126
	3.11	Refere	ences for Chapter 3	128
4	Min	ing D	ata Streams	131
	4.1	The S	tream Data Model	131
		4.1.1	A Data-Stream-Management System	132
		4.1.2	Examples of Stream Sources	133
		4.1.3	Stream Queries	134
		4.1.4	Issues in Stream Processing	135
	4.2	Sampl	ling Data in a Stream	136
		4.2.1	A Motivating Example	136
		4.2.2	Obtaining a Representative Sample	
		4.2.3	The General Sampling Problem	137
		4.2.4	Varying the Sample Size	
		4.2.5	Exercises for Section 4.2	138
	4.3	Filteri	ing Streams	139
		4.3.1	A Motivating Example	139
		4.3.2	The Bloom Filter	
		4.3.3	Analysis of Bloom Filtering	
		4.3.4	Exercises for Section 4.3	
	4.4	Count	ing Distinct Elements in a Stream	
		4.4.1	The Count-Distinct Problem	
		4.4.2	The Flajolet-Martin Algorithm	
		4.4.3	Combining Estimates	
		4.4.4	Space Requirements	
		4.4.5	Exercises for Section 4.4	
	4.5		ating Moments	
		4.5.1	Definition of Moments	145
		4.5.2	The Alon-Matias-Szegedy Algorithm for Second	
			Moments	
		4.5.3	Why the Alon-Matias-Szegedy Algorithm Works	
		4.5.4	Higher-Order Moments	
		4.5.5	Dealing With Infinite Streams	
		4.5.6	Exercises for Section 4.5	
	4.6		sing Ones in a Window	
		4.6.1	The Cost of Exact Counts	
		4.6.2	The Datar-Gionis-Indyk-Motwani Algorithm	
		4.6.3	Storage Requirements for the DGIM Algorithm	153

CONTENTS xi

		4.6.4	Query Answering in the DGIM Algorithm	. 153
		4.6.5	Maintaining the DGIM Conditions	. 154
		4.6.6	Reducing the Error	. 155
		4.6.7	Extensions to the Counting of Ones	
		4.6.8	Exercises for Section 4.6	. 157
	4.7	Decayi	ing Windows	. 157
		4.7.1	The Problem of Most-Common Elements	. 157
		4.7.2	Definition of the Decaying Window	. 158
		4.7.3	Finding the Most Popular Elements	. 159
	4.8	Summa	ary of Chapter 4	. 160
	4.9	Refere	nces for Chapter 4	. 161
5	Lin	k Analy	ysis	163
	5.1	PageR	ank	. 163
		5.1.1	Early Search Engines and Term Spam	. 164
		5.1.2	Definition of PageRank	. 165
		5.1.3	Structure of the Web	. 169
		5.1.4	Avoiding Dead Ends	. 170
		5.1.5	Spider Traps and Taxation	. 173
		5.1.6	Using PageRank in a Search Engine	. 175
		5.1.7	Exercises for Section 5.1	
	5.2	Efficier	nt Computation of PageRank	. 177
		5.2.1	Representing Transition Matrices	
		5.2.2	PageRank Iteration Using MapReduce	
		5.2.3	Use of Combiners to Consolidate the Result Vector	
		5.2.4	Representing Blocks of the Transition Matrix	
		5.2.5	Other Efficient Approaches to PageRank Iteration	
		5.2.6	Exercises for Section 5.2	
	5.3	Topic-S	Sensitive PageRank	
		5.3.1	Motivation for Topic-Sensitive Page Rank	
		5.3.2	Biased Random Walks	
		5.3.3	Using Topic-Sensitive PageRank	
		5.3.4	Inferring Topics from Words	
		5.3.5	Exercises for Section $5.3 \ldots \ldots \ldots \ldots$	
	5.4	Link S	•	
		5.4.1	Architecture of a Spam Farm	
		5.4.2	Analysis of a Spam Farm	
		5.4.3	Combating Link Spam	
		5.4.4	TrustRank	
		5.4.5	Spam Mass	
		5.4.6	Exercises for Section 5.4	
	5.5		and Authorities	_
		5.5.1	The Intuition Behind HITS	
		5.5.2	Formalizing Hubbiness and Authority	
		5.5.3	Exercises for Section 5.5	. 196

xii CONTENTS

	5.6	Summ	nary of Chapter 5
	5.7	Refere	ences for Chapter 5
6	Frequent Itemsets		Itemsets 201
U	6.1		Aarket-Basket Model
	0.1	6.1.1	Definition of Frequent Itemsets
		6.1.2	Applications of Frequent Itemsets
		6.1.3	Association Rules
		6.1.4	Finding Association Rules with High Confidence 207
		6.1.5	Exercises for Section 6.1
	6.2	-	et Baskets and the A-Priori Algorithm 209
	0.2	6.2.1	Representation of Market-Basket Data 209
		6.2.2	Use of Main Memory for Itemset Counting
		6.2.3	Monotonicity of Itemsets
		6.2.4	Tyranny of Counting Pairs
		6.2.5	The A-Priori Algorithm
		6.2.6	A-Priori for All Frequent Itemsets
		6.2.7	Exercises for Section 6.2
	6.3	Handl	ing Larger Datasets in Main Memory
		6.3.1	The Algorithm of Park, Chen, and Yu
		6.3.2	The Multistage Algorithm
		6.3.3	The Multihash Algorithm
		6.3.4	Exercises for Section 6.3
	6.4	Limite	ed-Pass Algorithms
		6.4.1	The Simple, Randomized Algorithm
		6.4.2	Avoiding Errors in Sampling Algorithms
		6.4.3	The Algorithm of Savasere, Omiecinski, and
			Navathe
		6.4.4	The SON Algorithm and MapReduce 229
		6.4.5	Toivonen's Algorithm
		6.4.6	Why Toivonen's Algorithm Works
		6.4.7	Exercises for Section 6.4
	6.5	Count	ing Frequent Items in a Stream
		6.5.1	Sampling Methods for Streams
		6.5.2	Frequent Itemsets in Decaying Windows
		6.5.3	Hybrid Methods
		6.5.4	Exercises for Section 6.5
	6.6	Summ	nary of Chapter 6
	6.7	Refere	ences for Chapter 6
7	Clu	stering	241
	7.1	-	luction to Clustering Techniques
		7.1.1	Points, Spaces, and Distances
		7.1.2	Clustering Strategies
		7.1.3	The Curse of Dimensionality
			V

CONTENTS xiii

		7.1.4	Exercises for Section 7.1	245
	7.2	Hierai	rchical Clustering	
		7.2.1	Hierarchical Clustering in a Euclidean Space	
		7.2.2	Efficiency of Hierarchical Clustering	248
		7.2.3	Alternative Rules for Controlling Hierarchical	
			Clustering	249
		7.2.4	Hierarchical Clustering in Non-Euclidean Spaces	252
		7.2.5	Exercises for Section 7.2	253
	7.3	K-mea	ans Algorithms	254
		7.3.1	K-Means Basics	255
		7.3.2	Initializing Clusters for K-Means	255
		7.3.3	Picking the Right Value of k	256
		7.3.4	The Algorithm of Bradley, Fayyad, and Reina	257
		7.3.5	Processing Data in the BFR Algorithm	259
		7.3.6	Exercises for Section 7.3	262
	7.4	The C	CURE Algorithm	262
		7.4.1	Initialization in CURE	263
		7.4.2	Completion of the CURE Algorithm	264
		7.4.3	Exercises for Section 7.4	265
	7.5	Cluste	ering in Non-Euclidean Spaces	266
		7.5.1	Representing Clusters in the GRGPF Algorithm	266
		7.5.2	Initializing the Cluster Tree	
		7.5.3	Adding Points in the GRGPF Algorithm	268
		7.5.4	Splitting and Merging Clusters	269
		7.5.5	Exercises for Section 7.5	270
	7.6	Cluste	ering for Streams and Parallelism	270
		7.6.1	The Stream-Computing Model	271
		7.6.2	A Stream-Clustering Algorithm	271
		7.6.3	Initializing Buckets	
		7.6.4	Merging Buckets	272
		7.6.5	Answering Queries	275
		7.6.6	Clustering in a Parallel Environment	
		7.6.7	Exercises for Section 7.6	
	7.7		nary of Chapter 7	
	7.8	Refere	ences for Chapter 7	280
8	۸ds	zortici	ng on the Web	281
G	8.1		in On-Line Advertising	
	0.1	8.1.1	Advertising Opportunities	
		8.1.2	Direct Placement of Ads	
		8.1.3	Issues for Display Ads	
	8.2		ne Algorithms	
	0.2	8.2.1	On-Line and Off-Line Algorithms	
		8.2.2	Greedy Algorithms	
		8.2.3	The Competitive Ratio	
		0.4.0	The compositive rand	200

xiv CONTENTS

		8.2.4	Exercises for Section 8.2	28	36
	8.3	The N	Matching Problem	28	37
		8.3.1	Matches and Perfect Matches	28	37
		8.3.2	The Greedy Algorithm for Maximal Matching	28	38
		8.3.3	Competitive Ratio for Greedy Matching	28	39
		8.3.4	Exercises for Section 8.3	29	0
	8.4	The A	Adwords Problem	29	0
		8.4.1	History of Search Advertising	29	1
		8.4.2	Definition of the Adwords Problem	29	1
		8.4.3	The Greedy Approach to the Adwords Problem		
		8.4.4	The Balance Algorithm	29	13
		8.4.5	A Lower Bound on Competitive Ratio for Balance .		
		8.4.6	The Balance Algorithm with Many Bidders	29	16
		8.4.7	The Generalized Balance Algorithm	29	17
		8.4.8	Final Observations About the Adwords Problem	29	18
		8.4.9	Exercises for Section 8.4	29	9
	8.5	Adwo	rds Implementation	29	19
		8.5.1	Matching Bids and Search Queries	30	0
		8.5.2	More Complex Matching Problems	30	0
		8.5.3	A Matching Algorithm for Documents and Bids		
	8.6		nary of Chapter 8		
	8.7	Refere	ences for Chapter 8	30)5
9	Rec	omme	endation Systems	30	7
	9.1	A Mo	del for Recommendation Systems	30)7
		9.1.1	The Utility Matrix	30	8(
		9.1.2	The Long Tail	30	9
		9.1.3	Applications of Recommendation Systems	30	9
		9.1.4	Populating the Utility Matrix	31	.1
	9.2	Conte	ent-Based Recommendations	31	.2
		9.2.1	Item Profiles		
		9.2.2	Discovering Features of Documents		
		9.2.3	Obtaining Item Features From Tags		
		9.2.4	Representing Item Profiles		
		9.2.5	User Profiles		
		9.2.6	Recommending Items to Users Based on Content		
		9.2.7	Classification Algorithms		
		9.2.8	Exercises for Section 9.2	32	20
	9.3		porative Filtering		
		9.3.1	Measuring Similarity		
		9.3.2	The Duality of Similarity		
		9.3.3	Clustering Users and Items		
		9.3.4	Exercises for Section 9.3		
	9.4	Dimer	nsionality Reduction	32	2
	0.1	Dillici			
	0.1	9.4.1	UV-Decomposition		

CONTENTS xv

		9.4.2	Root-Mean-Square Error	. 329
		9.4.3	Incremental Computation of a UV-Decomposition	. 330
		9.4.4	Optimizing an Arbitrary Element	. 332
		9.4.5	Building a Complete UV-Decomposition Algorithm	. 334
		9.4.6	Exercises for Section 9.4	. 336
	9.5	The N	etFlix Challenge	. 337
	9.6	Summ	ary of Chapter 9	. 338
	9.7	Refere	nces for Chapter 9	. 340
10	Min	ing So	cial-Network Graphs	343
			Networks as Graphs	. 343
	-		What is a Social Network?	
			Social Networks as Graphs	
			Varieties of Social Networks	
			Graphs With Several Node Types	
			Exercises for Section 10.1	
	10.2		ring of Social-Network Graphs	
			Distance Measures for Social-Network Graphs	
			Applying Standard Clustering Methods	
			Betweenness	
			The Girvan-Newman Algorithm	
			Using Betweenness to Find Communities	
			Exercises for Section 10.2	
	10.3	Direct	Discovery of Communities	. 357
		10.3.1	Finding Cliques	. 357
			Complete Bipartite Graphs	
			Finding Complete Bipartite Subgraphs	
		10.3.4	Why Complete Bipartite Graphs Must Exist	. 359
		10.3.5	Exercises for Section 10.3	. 361
	10.4	Partiti	oning of Graphs	. 361
			What Makes a Good Partition?	
		10.4.2	Normalized Cuts	. 362
		10.4.3	Some Matrices That Describe Graphs	. 363
			Eigenvalues of the Laplacian Matrix	
			Alternative Partitioning Methods	
		10.4.6	Exercises for Section 10.4	. 368
	10.5	Findin	g Overlapping Communities	. 369
		10.5.1	The Nature of Communities	. 369
			Maximum-Likelihood Estimation	
			The Affiliation-Graph Model	
			Avoiding the Use of Discrete Membership Changes	
			Exercises for Section 10.5	
	10.6		nk	
			Random Walkers on a Social Graph	
		10.6.2	Random Walks with Restart	. 377

xvi CONTENTS

		10.6.3	Exercises for Section 10.6	 380
	10.7		ing Triangles	
			Why Count Triangles?	
			An Algorithm for Finding Triangles	
			Optimality of the Triangle-Finding Algorithm	
			Finding Triangles Using MapReduce	
			Using Fewer Reduce Tasks	
			Exercises for Section 10.7	
	10.8	Neighb	oorhood Properties of Graphs	 386
		10.8.1	Directed Graphs and Neighborhoods	 386
		10.8.2	The Diameter of a Graph	 388
		10.8.3	Transitive Closure and Reachability	 389
		10.8.4	Transitive Closure Via MapReduce	 390
			Smart Transitive Closure	
		10.8.6	Transitive Closure by Graph Reduction	 393
		10.8.7	Approximating the Sizes of Neighborhoods	 395
		10.8.8	Exercises for Section 10.8	 397
	10.9	Summa	ary of Chapter 10	 398
	10.10	Refere	nces for Chapter 10	 402
11	Dim	ension	nality Reduction	405
			values and Eigenvectors of Symmetric Matrices	
			Definitions	
			Computing Eigenvalues and Eigenvectors	
		11.1.3	Finding Eigenpairs by Power Iteration	 408
			The Matrix of Eigenvectors	
		11.1.5	Exercises for Section 11.1	 411
	11.2	Princip	pal-Component Analysis	 412
		11.2.1	An Illustrative Example	 413
		11.2.2	Using Eigenvectors for Dimensionality Reduction .	 416
		11.2.3	The Matrix of Distances	 417
			Exercises for Section 11.2	
	11.3		ar-Value Decomposition	
			Definition of SVD	
			Interpretation of SVD	
			Dimensionality Reduction Using SVD	
			Why Zeroing Low Singular Values Works	
			Querying Using Concepts	
			Computing the SVD of a Matrix	
			Exercises for Section 11.3	
	11.4		Decomposition	
			Definition of CUR	
			Choosing Rows and Columns Properly	
			Constructing the Middle Matrix	
		11.4.4	The Complete CUR Decomposition	 432

CONTENTS		xvii

			Eliminating Duplicate Rows and Columns 433
			Exercises for Section 11.4
			ary of Chapter 11
	11.6	Referei	aces for Chapter 11
12	Larg	ge-Scal	e Machine Learning 439
	12.1		achine-Learning Model
		12.1.1	Training Sets
			Some Illustrative Examples
		12.1.3	Approaches to Machine Learning
			Machine-Learning Architecture
		12.1.5	Exercises for Section 12.1 $\dots \dots \dots$
	12.2	Percep	trons $\dots \dots \dots$
		12.2.1	Training a Perceptron with Zero Threshold 447
			Convergence of Perceptrons
			The Winnow Algorithm
		12.2.4	Allowing the Threshold to Vary
		12.2.5	Multiclass Perceptrons
		12.2.6	Transforming the Training Set
		12.2.7	Problems With Perceptrons
		12.2.8	Parallel Implementation of Perceptrons 458
		12.2.9	Exercises for Section 12.2
	12.3		t-Vector Machines
		12.3.1	The Mechanics of an SVM
		12.3.2	Normalizing the Hyperplane
			Finding Optimal Approximate Separators 464
			SVM Solutions by Gradient Descent
		12.3.5	Stochastic Gradient Descent
		12.3.6	Parallel Implementation of SVM 471
		12.3.7	Exercises for Section 12.3 $\dots \dots \dots$
	12.4	Learnin	ng from Nearest Neighbors
		12.4.1	The Framework for Nearest-Neighbor Calculations \dots 473
		12.4.2	Learning with One Nearest Neighbor 473
		12.4.3	Learning One-Dimensional Functions 474
		12.4.4	Kernel Regression
		12.4.5	Dealing with High-Dimensional Euclidean Data 477
		12.4.6	Dealing with Non-Euclidean Distances 479
		12.4.7	Exercises for Section 12.4
	12.5	Compa	arison of Learning Methods
			ary of Chapter $\overset{\circ}{12}$
			nces for Chapter 12