

Charu C. Aggarwal

Linear Algebra and Optimization for Machine Learning

A Textbook

A frequent challenge faced by beginners in machine learning is the extensive background requirement in linear algebra and optimization. This makes the learning curve very steep. This book, therefore, reverses the focus by teaching linear algebra and optimization as the primary topics of interest, and solutions to machine learning problems as applications of these methods. Therefore, the book also provides significant exposure to machine learning. The chapters of this book belong to two categories:

1. Linear algebra and its applications: These chapters focus on the basics of linear algebra together with their common applications to singular value decomposition, similarity matrices (kernel methods), and graph analysis. Numerous machine learning applications have been used as examples, such as spectral clustering, kernel-based classification, and outlier detection.

2. Optimization and its applications: Basic methods in optimization such as gradient descent, Newton's method, and coordinate descent are discussed. Constrained optimization methods are introduced as well. Machine learning applications such as linear regression, SVMs, logistic regression, matrix factorization, recommender systems, and K-means clustering are discussed in detail. A general view of optimization in computational graphs is discussed together with its applications to backpropagation in neural networks.

Exercises are included both within the text of the chapters and at the end of the chapters. The book is written for a diverse audience, including graduate students, researchers, and practitioners.

About the Author

Charu C. Aggarwal is a Distinguished Research Staff Member (DRSM) at the IBM T. J. Watson Research Center in Yorktown Heights, New York. He completed his undergraduate degree in Computer Science from the Indian Institute of Technology at Kanpur in 1993 and his Ph.D. in Operations Research from the Massachusetts Institute of Technology in 1996. He has published more than 400 papers in refereed conferences and journals, and has applied for or been granted more than 80 patents.

He is author or editor of 19 books, including textbooks on data mining, neural networks, machine learning (for text), recommender systems, and outlier analysis. Because of the commercial value of his patents, he has thrice been designated a Master Inventor at IBM. He has received several internal and external awards, including the EDBT Test-of-Time Award (2014), the IEEE ICDM Research Contributions Award (2015), and the ACM SIGKDD Innovation Award (2019). He has served as editor-in-chief of the ACM SIGKDD Explorations, and is currently serving as an editor-in-chief of the ACM Transactions on Knowledge Discovery from Data. He is also an editor-in-chief of ACM Books. He is a fellow of the SIAM, ACM, and the IEEE, for "contributions to knowledge discovery and data mining algorithms."

ISBN 978-3-030-40343-0



9 783030 403430

► springer.com

Aggarwal



Linear Algebra and Optimization for Machine Learning

Charu C. Aggarwal

Linear Algebra and Optimization for Machine Learning

A Textbook

 Springer

Linear Algebra and Optimization for Machine Learning

Charu C. Aggarwal

Linear Algebra and Optimization for Machine Learning

A Textbook

 Springer

Charu C. Aggarwal
Distinguished Research Staff Member
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA

ISBN 978-3-030-40343-0 ISBN 978-3-030-40344-7 (eBook)
<https://doi.org/10.1007/978-3-030-40344-7>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To my wife Lata, my daughter Sayani,
and all my mathematics teachers*

Contents

- 1 Linear Algebra and Optimization: An Introduction** **1**
- 1.1 Introduction 1
- 1.2 Scalars, Vectors, and Matrices 2
 - 1.2.1 Basic Operations with Scalars and Vectors 3
 - 1.2.2 Basic Operations with Vectors and Matrices 8
 - 1.2.3 Special Classes of Matrices 12
 - 1.2.4 Matrix Powers, Polynomials, and the Inverse 14
 - 1.2.5 The Matrix Inversion Lemma: Inverting the Sum of Matrices 17
 - 1.2.6 Frobenius Norm, Trace, and Energy 19
- 1.3 Matrix Multiplication as a Decomposable Operator 21
 - 1.3.1 Matrix Multiplication as Decomposable Row and Column Operators 21
 - 1.3.2 Matrix Multiplication as Decomposable Geometric Operators 25
- 1.4 Basic Problems in Machine Learning 27
 - 1.4.1 Matrix Factorization 27
 - 1.4.2 Clustering 28
 - 1.4.3 Classification and Regression Modeling 29
 - 1.4.4 Outlier Detection 30
- 1.5 Optimization for Machine Learning 31
 - 1.5.1 The Taylor Expansion for Function Simplification 31
 - 1.5.2 Example of Optimization in Machine Learning 33
 - 1.5.3 Optimization in Computational Graphs 34
- 1.6 Summary 35
- 1.7 Further Reading 35
- 1.8 Exercises 36

- 2 Linear Transformations and Linear Systems** **41**
- 2.1 Introduction 41
 - 2.1.1 What Is a Linear Transform? 42
- 2.2 The Geometry of Matrix Multiplication 43

2.3	Vector Spaces and Their Geometry	51
2.3.1	Coordinates in a Basis System	55
2.3.2	Coordinate Transformations Between Basis Sets	57
2.3.3	Span of a Set of Vectors	59
2.3.4	Machine Learning Example: Discrete Wavelet Transform	60
2.3.5	Relationships Among Subspaces of a Vector Space	61
2.4	The Linear Algebra of Matrix Rows and Columns	63
2.5	The Row Echelon Form of a Matrix	64
2.5.1	LU Decomposition	66
2.5.2	Application: Finding a Basis Set	67
2.5.3	Application: Matrix Inversion	67
2.5.4	Application: Solving a System of Linear Equations	68
2.6	The Notion of Matrix Rank	70
2.6.1	Effect of Matrix Operations on Rank	71
2.7	Generating Orthogonal Basis Sets	73
2.7.1	Gram-Schmidt Orthogonalization and QR Decomposition	73
2.7.2	QR Decomposition	74
2.7.3	The Discrete Cosine Transform	77
2.8	An Optimization-Centric View of Linear Systems	79
2.8.1	Moore-Penrose Pseudoinverse	81
2.8.2	The Projection Matrix	82
2.9	Ill-Conditioned Matrices and Systems	85
2.10	Inner Products: A Geometric View	86
2.11	Complex Vector Spaces	87
2.11.1	The Discrete Fourier Transform	89
2.12	Summary	90
2.13	Further Reading	91
2.14	Exercises	91
3	Eigenvectors and Diagonalizable Matrices	97
3.1	Introduction	97
3.2	Determinants	98
3.3	Diagonalizable Transformations and Eigenvectors	103
3.3.1	Complex Eigenvalues	107
3.3.2	Left Eigenvectors and Right Eigenvectors	108
3.3.3	Existence and Uniqueness of Diagonalization	109
3.3.4	Existence and Uniqueness of Triangulization	111
3.3.5	Similar Matrix Families Sharing Eigenvalues	113
3.3.6	Diagonalizable Matrix Families Sharing Eigenvectors	115
3.3.7	Symmetric Matrices	115
3.3.8	Positive Semidefinite Matrices	117
3.3.9	Cholesky Factorization: Symmetric LU Decomposition	119
3.4	Machine Learning and Optimization Applications	120
3.4.1	Fast Matrix Operations in Machine Learning	121
3.4.2	Examples of Diagonalizable Matrices in Machine Learning	121
3.4.3	Symmetric Matrices in Quadratic Optimization	124
3.4.4	Diagonalization Application: Variable Separation for Optimization	128
3.4.5	Eigenvectors in Norm-Constrained Quadratic Programming	130

3.5	Numerical Algorithms for Finding Eigenvectors	131
3.5.1	The QR Method via Schur Decomposition	132
3.5.2	The Power Method for Finding Dominant Eigenvectors	133
3.6	Summary	135
3.7	Further Reading	135
3.8	Exercises	135
4	Optimization Basics: A Machine Learning View	141
4.1	Introduction	141
4.2	The Basics of Optimization	142
4.2.1	Univariate Optimization	142
4.2.1.1	Why We Need Gradient Descent	146
4.2.1.2	Convergence of Gradient Descent	147
4.2.1.3	The Divergence Problem	148
4.2.2	Bivariate Optimization	149
4.2.3	Multivariate Optimization	151
4.3	Convex Objective Functions	154
4.4	The Minutiae of Gradient Descent	159
4.4.1	Checking Gradient Correctness with Finite Differences	159
4.4.2	Learning Rate Decay and Bold Driver	159
4.4.3	Line Search	160
4.4.3.1	Binary Search	161
4.4.3.2	Golden-Section Search	161
4.4.3.3	Armijo Rule	162
4.4.4	Initialization	163
4.5	Properties of Optimization in Machine Learning	163
4.5.1	Typical Objective Functions and Additive Separability	163
4.5.2	Stochastic Gradient Descent	164
4.5.3	How Optimization in Machine Learning Is Different	165
4.5.4	Tuning Hyperparameters	168
4.5.5	The Importance of Feature Preprocessing	168
4.6	Computing Derivatives with Respect to Vectors	169
4.6.1	Matrix Calculus Notation	170
4.6.2	Useful Matrix Calculus Identities	171
4.6.2.1	Application: Unconstrained Quadratic Programming	173
4.6.2.2	Application: Derivative of Squared Norm	174
4.6.3	The Chain Rule of Calculus for Vecteded Derivatives	174
4.6.3.1	Useful Examples of Vecteded Derivatives	175
4.7	Linear Regression: Optimization with Numerical Targets	176
4.7.1	Tikhonov Regularization	178
4.7.1.1	Pseudoinverse and Connections to Regularization	179
4.7.2	Stochastic Gradient Descent	179
4.7.3	The Use of Bias	179
4.7.3.1	Heuristic Initialization	180
4.8	Optimization Models for Binary Targets	180
4.8.1	Least-Squares Classification: Regression on Binary Targets	181
4.8.1.1	Why Least-Squares Classification Loss Needs Repair	183

4.8.2	The Support Vector Machine	184
4.8.2.1	Computing Gradients	185
4.8.2.2	Stochastic Gradient Descent	186
4.8.3	Logistic Regression	186
4.8.3.1	Computing Gradients	188
4.8.3.2	Stochastic Gradient Descent	188
4.8.4	How Linear Regression Is a Parent Problem in Machine Learning	189
4.9	Optimization Models for the MultiClass Setting	190
4.9.1	Weston-Watkins Support Vector Machine	190
4.9.1.1	Computing Gradients	191
4.9.2	Multinomial Logistic Regression	192
4.9.2.1	Computing Gradients	193
4.9.2.2	Stochastic Gradient Descent	194
4.10	Coordinate Descent	194
4.10.1	Linear Regression with Coordinate Descent	196
4.10.2	Block Coordinate Descent	197
4.10.3	K-Means as Block Coordinate Descent	197
4.11	Summary	198
4.12	Further Reading	199
4.13	Exercises	199
5	Advanced Optimization Solutions	205
5.1	Introduction	205
5.2	Challenges in Gradient-Based Optimization	206
5.2.1	Local Optima and Flat Regions	207
5.2.2	Differential Curvature	208
5.2.2.1	Revisiting Feature Normalization	209
5.2.3	Examples of Difficult Topologies: Cliffs and Valleys	210
5.3	Adjusting First-Order Derivatives for Descent	212
5.3.1	Momentum-Based Learning	212
5.3.2	AdaGrad	214
5.3.3	RMSProp	215
5.3.4	Adam	215
5.4	The Newton Method	216
5.4.1	The Basic Form of the Newton Method	217
5.4.2	Importance of Line Search for Non-quadratic Functions	219
5.4.3	Example: Newton Method in the Quadratic Bowl	220
5.4.4	Example: Newton Method in a Non-quadratic Function	220
5.5	Newton Methods in Machine Learning	221
5.5.1	Newton Method for Linear Regression	221
5.5.2	Newton Method for Support-Vector Machines	223
5.5.3	Newton Method for Logistic Regression	225
5.5.4	Connections Among Different Models and Unified Framework	228
5.6	Newton Method: Challenges and Solutions	229
5.6.1	Singular and Indefinite Hessian	229
5.6.2	The Saddle-Point Problem	229

5.6.3	Convergence Problems and Solutions with Non-quadratic Functions	231
5.6.3.1	Trust Region Method	232
5.7	Computationally Efficient Variations of Newton Method	233
5.7.1	Conjugate Gradient Method	233
5.7.2	Quasi-Newton Methods and BFGS	237
5.8	Non-differentiable Optimization Functions	239
5.8.1	The Subgradient Method	240
5.8.1.1	Application: L_1 -Regularization	242
5.8.1.2	Combining Subgradients with Coordinate Descent	243
5.8.2	Proximal Gradient Method	244
5.8.2.1	Application: Alternative for L_1 -Regularized Regression	245
5.8.3	Designing Surrogate Loss Functions for Combinatorial Optimization	246
5.8.3.1	Application: Ranking Support Vector Machine	247
5.8.4	Dynamic Programming for Optimizing Sequential Decisions	248
5.8.4.1	Application: Fast Matrix Multiplication	249
5.9	Summary	250
5.10	Further Reading	250
5.11	Exercises	251
6	Constrained Optimization and Duality	255
6.1	Introduction	255
6.2	Primal Gradient Descent Methods	256
6.2.1	Linear Equality Constraints	257
6.2.1.1	Convex Quadratic Program with Equality Constraints	259
6.2.1.2	Application: Linear Regression with Equality Constraints	261
6.2.1.3	Application: Newton Method with Equality Constraints	262
6.2.2	Linear Inequality Constraints	262
6.2.2.1	The Special Case of Box Constraints	263
6.2.2.2	General Conditions for Projected Gradient Descent to Work	264
6.2.2.3	Sequential Linear Programming	266
6.2.3	Sequential Quadratic Programming	267
6.3	Primal Coordinate Descent	267
6.3.1	Coordinate Descent for Convex Optimization Over Convex Set	268
6.3.2	Machine Learning Application: Box Regression	269
6.4	Lagrangian Relaxation and Duality	270
6.4.1	Kuhn-Tucker Optimality Conditions	274
6.4.2	General Procedure for Using Duality	276
6.4.2.1	Inferring the Optimal Primal Solution from Optimal Dual Solution	276
6.4.3	Application: Formulating the SVM Dual	276
6.4.3.1	Inferring the Optimal Primal Solution from Optimal Dual Solution	278

6.4.4	Optimization Algorithms for the SVM Dual	279
6.4.4.1	Gradient Descent	279
6.4.4.2	Coordinate Descent	280
6.4.5	Getting the Lagrangian Relaxation of Unconstrained Problems	281
6.4.5.1	Machine Learning Application: Dual of Linear Regression	283
6.5	Penalty-Based and Primal-Dual Methods	286
6.5.1	Penalty Method with Single Constraint	286
6.5.2	Penalty Method: General Formulation	287
6.5.3	Barrier and Interior Point Methods	288
6.6	Norm-Constrained Optimization	290
6.7	Primal Versus Dual Methods	292
6.8	Summary	293
6.9	Further Reading	294
6.10	Exercises	294
7	Singular Value Decomposition	299
7.1	Introduction	299
7.2	SVD: A Linear Algebra Perspective	300
7.2.1	Singular Value Decomposition of a Square Matrix	300
7.2.2	Square SVD to Rectangular SVD via Padding	304
7.2.3	Several Definitions of Rectangular Singular Value Decomposition	305
7.2.4	Truncated Singular Value Decomposition	307
7.2.4.1	Relating Truncation Loss to Singular Values	309
7.2.4.2	Geometry of Rank- k Truncation	311
7.2.4.3	Example of Truncated SVD	311
7.2.5	Two Interpretations of SVD	313
7.2.6	Is Singular Value Decomposition Unique?	315
7.2.7	Two-Way Versus Three-Way Decompositions	316
7.3	SVD: An Optimization Perspective	317
7.3.1	A Maximization Formulation with Basis Orthogonality	318
7.3.2	A Minimization Formulation with Residuals	319
7.3.3	Generalization to Matrix Factorization Methods	320
7.3.4	Principal Component Analysis	320
7.4	Applications of Singular Value Decomposition	323
7.4.1	Dimensionality Reduction	323
7.4.2	Noise Removal	324
7.4.3	Finding the Four Fundamental Subspaces in Linear Algebra	325
7.4.4	Moore-Penrose Pseudoinverse	325
7.4.4.1	Ill-Conditioned Square Matrices	326
7.4.5	Solving Linear Equations and Linear Regression	327
7.4.6	Feature Preprocessing and Whitening in Machine Learning	327
7.4.7	Outlier Detection	328
7.4.8	Feature Engineering	329
7.5	Numerical Algorithms for SVD	330
7.6	Summary	332
7.7	Further Reading	332
7.8	Exercises	333

8 Matrix Factorization 339

- 8.1 Introduction 339
- 8.2 Optimization-Based Matrix Factorization 341
 - 8.2.1 Example: K-Means as Constrained Matrix Factorization 342
- 8.3 Unconstrained Matrix Factorization 342
 - 8.3.1 Gradient Descent with Fully Specified Matrices 343
 - 8.3.2 Application to Recommender Systems 346
 - 8.3.2.1 Stochastic Gradient Descent 348
 - 8.3.2.2 Coordinate Descent 348
 - 8.3.2.3 Block Coordinate Descent: Alternating Least Squares 349
- 8.4 Nonnegative Matrix Factorization 350
 - 8.4.1 Optimization Problem with Frobenius Norm 350
 - 8.4.1.1 Projected Gradient Descent with Box Constraints 351
 - 8.4.2 Solution Using Duality 351
 - 8.4.3 Interpretability of Nonnegative Matrix Factorization 353
 - 8.4.4 Example of Nonnegative Matrix Factorization 353
 - 8.4.5 The I-Divergence Objective Function 356
- 8.5 Weighted Matrix Factorization 356
 - 8.5.1 Practical Use Cases of Nonnegative and Sparse Matrices 357
 - 8.5.2 Stochastic Gradient Descent 359
 - 8.5.2.1 Why Negative Sampling Is Important 360
 - 8.5.3 Application: Recommendations with Implicit Feedback Data 360
 - 8.5.4 Application: Link Prediction in Adjacency Matrices 360
 - 8.5.5 Application: Word-Word Context Embedding with GloVe 361
- 8.6 Nonlinear Matrix Factorizations 362
 - 8.6.1 Logistic Matrix Factorization 362
 - 8.6.1.1 Gradient Descent Steps for Logistic Matrix Factorization 363
 - 8.6.2 Maximum Margin Matrix Factorization 364
- 8.7 Generalized Low-Rank Models 365
 - 8.7.1 Handling Categorical Entries 367
 - 8.7.2 Handling Ordinal Entries 367
- 8.8 Shared Matrix Factorization 369
 - 8.8.1 Gradient Descent Steps for Shared Factorization 370
 - 8.8.2 How to Set Up Shared Models in Arbitrary Scenarios 370
- 8.9 Factorization Machines 371
- 8.10 Summary 375
- 8.11 Further Reading 375
- 8.12 Exercises 375

9 The Linear Algebra of Similarity 379

- 9.1 Introduction 379
- 9.2 Equivalence of Data and Similarity Matrices 379
 - 9.2.1 From Data Matrix to Similarity Matrix and Back 380
 - 9.2.2 When Is Data Recovery from a Similarity Matrix Useful? 381
 - 9.2.3 What Types of Similarity Matrices Are “Valid”? 382
 - 9.2.4 Symmetric Matrix Factorization as an Optimization Model 383
 - 9.2.5 Kernel Methods: The Machine Learning Terminology 383

9.3	Efficient Data Recovery from Similarity Matrices	385
9.3.1	Nyström Sampling	385
9.3.2	Matrix Factorization with Stochastic Gradient Descent	386
9.3.3	Asymmetric Similarity Decompositions	388
9.4	Linear Algebra Operations on Similarity Matrices	389
9.4.1	Energy of Similarity Matrix and Unit Ball Normalization	390
9.4.2	Norm of the Mean and Variance	390
9.4.3	Centering a Similarity Matrix	391
9.4.3.1	Application: Kernel PCA	391
9.4.4	From Similarity Matrix to Distance Matrix and Back	392
9.4.4.1	Application: ISOMAP	393
9.5	Machine Learning with Similarity Matrices	394
9.5.1	Feature Engineering from Similarity Matrix	395
9.5.1.1	Kernel Clustering	395
9.5.1.2	Kernel Outlier Detection	396
9.5.1.3	Kernel Classification	396
9.5.2	Direct Use of Similarity Matrix	397
9.5.2.1	Kernel K-Means	397
9.5.2.2	Kernel SVM	398
9.6	The Linear Algebra of the Representer Theorem	399
9.7	Similarity Matrices and Linear Separability	403
9.7.1	Transformations That Preserve Positive Semi-definiteness	405
9.8	Summary	407
9.9	Further Reading	407
9.10	Exercises	407
10	The Linear Algebra of Graphs	411
10.1	Introduction	411
10.2	Graph Basics and Adjacency Matrices	411
10.3	Powers of Adjacency Matrices	416
10.4	The Perron-Frobenius Theorem	419
10.5	The Right Eigenvectors of Graph Matrices	423
10.5.1	The Kernel View of Spectral Clustering	423
10.5.1.1	Relating Shi-Malik and Ng-Jordan-Weiss Embeddings	425
10.5.2	The Laplacian View of Spectral Clustering	426
10.5.2.1	Graph Laplacian	426
10.5.2.2	Optimization Model with Laplacian	428
10.5.3	The Matrix Factorization View of Spectral Clustering	430
10.5.3.1	Machine Learning Application: Directed Link Prediction	430
10.5.4	Which View of Spectral Clustering Is Most Informative?	431
10.6	The Left Eigenvectors of Graph Matrices	431
10.6.1	PageRank as Left Eigenvector of Transition Matrix	433
10.6.2	Related Measures of Prestige and Centrality	434
10.6.3	Application of Left Eigenvectors to Link Prediction	435
10.7	Eigenvectors of Reducible Matrices	436
10.7.1	Undirected Graphs	436
10.7.2	Directed Graphs	436

- 10.8 Machine Learning Applications 439
 - 10.8.1 Application to Vertex Classification 440
 - 10.8.2 Applications to Multidimensional Data 442
- 10.9 Summary 443
- 10.10 Further Reading 443
- 10.11 Exercises 444
- 11 Optimization in Computational Graphs 447**
 - 11.1 Introduction 447
 - 11.2 The Basics of Computational Graphs 448
 - 11.2.1 Neural Networks as Directed Computational Graphs 451
 - 11.3 Optimization in Directed Acyclic Graphs 453
 - 11.3.1 The Challenge of Computational Graphs 453
 - 11.3.2 The Broad Framework for Gradient Computation 455
 - 11.3.3 Computing Node-to-Node Derivatives Using Brute Force 456
 - 11.3.4 Dynamic Programming for Computing Node-to-Node Derivatives 459
 - 11.3.4.1 Example of Computing Node-to-Node Derivatives 461
 - 11.3.5 Converting Node-to-Node Derivatives into Loss-to-Weight Derivatives 464
 - 11.3.5.1 Example of Computing Loss-to-Weight Derivatives 465
 - 11.3.6 Computational Graphs with Vector Variables 466
 - 11.4 Application: Backpropagation in Neural Networks 468
 - 11.4.1 Derivatives of Common Activation Functions 470
 - 11.4.2 Vector-Centric Backpropagation 471
 - 11.4.3 Example of Vector-Centric Backpropagation 473
 - 11.5 A General View of Computational Graphs 475
 - 11.6 Summary 478
 - 11.7 Further Reading 478
 - 11.8 Exercises 478
- Bibliography 483**
- Index 491**

Preface

“Mathematics is the language with which God wrote the universe.”– Galileo

A frequent challenge faced by beginners in machine learning is the extensive background required in linear algebra and optimization. One problem is that the existing linear algebra and optimization courses are not specific to machine learning; therefore, one would typically have to complete more course material than is necessary to pick up machine learning. Furthermore, certain types of ideas and tricks from optimization and linear algebra recur more frequently in machine learning than other application-centric settings. Therefore, there is significant value in developing a view of linear algebra and optimization that is better suited to the specific perspective of machine learning.

It is common for machine learning practitioners to pick up missing bits and pieces of linear algebra and optimization via “osmosis” while studying the solutions to machine learning applications. However, this type of unsystematic approach is unsatisfying, because the primary focus on machine learning gets in the way of learning linear algebra and optimization in a generalizable way across new situations and applications. Therefore, we have inverted the focus in this book, with linear algebra and optimization as the primary topics of interest and solutions to machine learning problems as the *applications* of this machinery. *In other words, the book goes out of its way to teach linear algebra and optimization with machine learning examples.* By using this approach, the book focuses on those aspects of linear algebra and optimization that are more relevant to machine learning and also teaches the reader how to apply them in the machine learning context. As a side benefit, the reader will pick up knowledge of several fundamental problems in machine learning. At the end of the process, the reader will become familiar with many of the basic linear-algebra- and optimization-centric algorithms in machine learning. Although the book is not intended to provide exhaustive coverage of machine learning, it serves as a “technical starter” for the key models and optimization methods in machine learning. Even for seasoned practitioners of machine learning, a systematic introduction to fundamental linear algebra and optimization methodologies can be useful in terms of providing a fresh perspective.

The chapters of the book are organized as follows:

1. *Linear algebra and its applications:* The chapters focus on the basics of linear algebra together with their common applications to singular value decomposition, matrix factorization, similarity matrices (kernel methods), and graph analysis. Numerous machine learning applications have been used as examples, such as spectral clustering,

kernel-based classification, and outlier detection. The tight integration of linear algebra methods with examples from machine learning differentiates this book from generic volumes on linear algebra. The focus is clearly on the most relevant aspects of linear algebra for machine learning and to teach readers how to apply these concepts.

2. *Optimization and its applications:* Much of machine learning is posed as an optimization problem in which we try to maximize the accuracy of regression and classification models. The “parent problem” of optimization-centric machine learning is least-squares regression. Interestingly, this problem arises in both linear algebra and optimization and is one of the key connecting problems of the two fields. Least-squares regression is also the starting point for support vector machines, logistic regression, and recommender systems. Furthermore, the methods for dimensionality reduction and matrix factorization also require the development of optimization methods. A general view of optimization in computational graphs is discussed together with its applications to backpropagation in neural networks.

This book contains exercises both within the text of the chapter and at the end of the chapter. The exercises within the text of the chapter should be solved as one reads the chapter in order to solidify the concepts. This will lead to slower progress, but a better understanding. For in-chapter exercises, hints for the solution are given in order to help the reader along. The exercises at the end of the chapter are intended to be solved as refreshers after completing the chapter.

Throughout this book, a vector or a multidimensional data point is annotated with a bar, such as \bar{X} or \bar{y} . A vector or multidimensional point may be denoted by either small letters or capital letters, as long as it has a bar. Vector dot products are denoted by centered dots, such as $\bar{X} \cdot \bar{Y}$. A matrix is denoted in capital letters without a bar, such as R . Throughout the book, the $n \times d$ matrix corresponding to the entire training data set is denoted by D , with n data points and d dimensions. The individual data points in D are therefore d -dimensional row vectors and are often denoted by $\bar{X}_1 \dots \bar{X}_n$. Conversely, vectors with one component for each data point are usually n -dimensional column vectors. An example is the n -dimensional column vector \bar{y} of class variables of n data points. An observed value y_i is distinguished from a predicted value \hat{y}_i by a circumflex at the top of the variable.

Yorktown Heights, NY, USA

Charu C. Aggarwal

Acknowledgments

I would like to thank my family for their love and support during the busy time spent in writing this book. Knowledge of the very basics of optimization (e.g., calculus) and linear algebra (e.g., vectors and matrices) starts in high school and increases over the course of many years of undergraduate/graduate education as well as during the postgraduate years of research. As such, I feel indebted to a large number of teachers and collaborators over the years. This section is, therefore, a rather incomplete attempt to express my gratitude.

My initial exposure to vectors, matrices, and optimization (calculus) occurred during my high school years, where I was ably taught these subjects by S. Adhikari and P. C. Pathrose. Indeed, my love of mathematics started during those years, and I feel indebted to both these individuals for instilling the love of these subjects in me. During my undergraduate study in computer science at IIT Kanpur, I was taught several aspects of linear algebra and optimization by Dr. R. Ahuja, Dr. B. Bhatia, and Dr. S. Gupta. Even though linear algebra and mathematical optimization are distinct (but interrelated) subjects, Dr. Gupta's teaching style often provided an integrated view of these topics. I was able to fully appreciate the value of such an integrated view when working in machine learning. For example, one can approach many problems such as solving systems of equations or singular value decomposition either from a linear algebra viewpoint or from an optimization viewpoint, and both perspectives provide complementary views in different machine learning applications. Dr. Gupta's courses on linear algebra and mathematical optimization had a profound influence on me in choosing mathematical optimization as my field of study during my PhD years; this choice was relatively unusual for undergraduate computer science majors at that time. Finally, I had the good fortune to learn about linear and nonlinear optimization methods from several luminaries on these subjects during my graduate years at MIT. In particular, I feel indebted to my PhD thesis advisor James B. Orlin for his guidance during my early years. In addition, Nagui Halim has provided a lot of support for all my book-writing projects over the course of a decade and deserves a lot of credit for my work in this respect. My manager, Horst Samulowitz, has supported my work over the past year, and I would like to thank him for his help.

I also learned a lot from my collaborators in machine learning over the years. One often appreciates the true usefulness of linear algebra and optimization only in an applied setting, and I had the good fortune of working with many researchers from different areas on a wide range of machine learning problems. A lot of the emphasis in this book to specific aspects of linear algebra and optimization is derived from these invaluable experiences and

collaborations. In particular, I would like to thank Tarek F. Abdelzaher, Jinghui Chen, Jing Gao, Quanquan Gu, Manish Gupta, Jiawei Han, Alexander Hinneburg, Thomas Huang, Nan Li, Huan Liu, Ruoming Jin, Daniel Keim, Arijit Khan, Latifur Khan, Mohammad M. Masud, Jian Pei, Magda Procopiuc, Guojun Qi, Chandan Reddy, Saket Sathe, Jaideep Srivastava, Karthik Subbian, Yizhou Sun, Jiliang Tang, Min-Hsuan Tsai, Haixun Wang, Jianyong Wang, Min Wang, Suhang Wang, Wei Wang, Joel Wolf, Xifeng Yan, Wenchao Yu, Mohammed Zaki, ChengXiang Zhai, and Peixiang Zhao.

Several individuals have also reviewed the book. Quanquan Gu provided suggestions on Chapter 6. Jiliang Tang and Xiaorui Liu examined several portions of Chapter 6 and pointed out corrections and improvements. Shuiwang Ji contributed Problem 7.2.3. Jie Wang reviewed several chapters of the book and pointed out corrections. Hao Liu also provided several suggestions.

Last but not least, I would like to thank my daughter Sayani for encouraging me to write this book at a time when I had decided to hang up my boots on the issue of book writing. She encouraged me to write this one. I would also like to thank my wife for fixing some of the figures in this book.

Author Biography

Charu C. Aggarwal is a Distinguished Research Staff Member (DRSM) at the IBM T. J. Watson Research Center in Yorktown Heights, New York. He completed his undergraduate degree in Computer Science from the Indian Institute of Technology at Kanpur in 1993 and his Ph.D. from the Massachusetts Institute of Technology in 1996.



He has worked extensively in the field of data mining. He has published more than 400 papers in refereed conferences and journals and authored more than 80 patents. He is the author or editor of 19 books, including textbooks on data mining, recommender systems, and outlier analysis. Because of the commercial value of his patents, he has thrice been designated a Master Inventor at IBM. He is a recipient of an IBM Corporate Award (2003) for his work on bioterrorist threat detection in data streams, a recipient of the IBM Outstanding Innovation Award (2008) for his scientific contributions to privacy technology, and a recipient of two IBM Outstanding Technical Achievement Awards (2009, 2015) for his work on data streams/high-dimensional data. He received the EDBT 2014 Test of Time Award for his work on condensation-based privacy-preserving data mining. He is also a recipient of the IEEE ICDM Research Contributions Award (2015) and the ACM SIGKDD Innovation Award (2019), which are the two highest awards for influential research contributions in data mining.

He has served as the general cochair of the IEEE Big Data Conference (2014) and as the program cochair of the ACM CIKM Conference (2015), the IEEE ICDM Conference (2015), and the ACM KDD Conference (2016). He served as an associate editor of the IEEE Transactions on Knowledge and Data Engineering from 2004 to 2008. He is an associate editor of the IEEE Transactions on Big Data, an action editor of the Data Mining and Knowledge Discovery Journal, and an associate editor of the Knowledge and Information Systems Journal. He serves as the editor-in-chief of the ACM Transactions on Knowledge Discovery from Data as well as the ACM SIGKDD Explorations. He serves on the advisory board of the Lecture Notes on Social Networks, a publication by Springer. He has served as the vice president of the SIAM Activity Group on Data Mining and is a member of the SIAM Industry Committee. He is a fellow of the SIAM, ACM, and IEEE, for “contributions to knowledge discovery and data mining algorithms.”