

A Matrix Algebra Approach to Artificial Intelligence

Xian-Da Zhang

A Matrix Algebra Approach to Artificial Intelligence



Springer

Xian-Da Zhang (Deceased)
Department of Automation
Tsinghua University
Beijing, Beijing, China

ISBN 978-981-15-2769-2 ISBN 978-981-15-2770-8 (eBook)
<https://doi.org/10.1007/978-981-15-2770-8>

© Springer Nature Singapore Pte Ltd. 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 Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

Human intelligence is the intellectual prowess of humans, which is marked by four basic and important abilities: learning ability, cognition (acquiring and storing knowledge) ability, generalization ability, and computation ability. Correspondingly, artificial intelligence (AI) also consists of four basic and important methods: machine learning (learning intelligence), neural network (cognitive intelligence), support vector machines (generalization intelligence), and evolutionary computation (computational intelligence).

The development of AI is built on mathematics. For example, multivariate calculus deals with the aspect of numerical optimization, which is the driving force behind most machine learning algorithms. The main math applications in AI are matrix algebra, optimization, and mathematical statistics, but the latter two are usually described and applied in the form of matrix. Therefore, matrix algebra is a vast mathematical tool of fundamental importance in most AI subjects.

The aim of this book is to provide the solid matrix algebra theory and methods for four basic and important AI fields, including machine learning, neural networks, support vector machines, and evolutionary computation.

Structure and Contents

The book consists of two parts.

Part I (Introduction to Matrix Algebra) provides fundamentals of matrix algebra and contains Chaps. 1 through 5. Chapter 1 presents the basic operations and performances of matrices, followed by a description of vectorization of matrix and matricization of vector. Chapter 2 is devoted to matrix differential as an important and effective tool in gradient computation and optimization. Chapter 3 is concerned with convex optimization theory and methods by focusing on gradient/subgradient methods in smooth and nonsmooth convex optimizations, and constrained convex optimization. Chapter 4 describes singular value decomposition (SVD) together with Tikhonov regularization and total least squares for solving over-determined

matrix equations, followed by the Lasso and LARS methods for solving under-determined matrix equations. Chapter 5 is devoted to the eigenvalue decomposition (EVD), the generalized eigenvalue decomposition, the Rayleigh quotient, and the generalized Rayleigh quotient.

Part II (Artificial Intelligence) focuses on machine learning, neural networks, support vector machines (SVMs), and evolutionary computation from the perspective of matrix algebra. This part is the main body of the book and consists of the following four chapters.

Chapter 6 (Machine Learning) presents first the basic theory and methods in machine learning including single-objective optimization, feature selection, principal component analysis and canonical correlation analysis together with supervised, unsupervised, and semi-supervised learning and active learning. Then, this chapter highlights topics and advances in machine learning: graph machine learning, reinforcement learning, Q-learning, and transfer learning.

Chapter 7 (Neural Networks) describes optimization problem, activation functions, and basic neural networks. The core part of this chapter are topics and advances in neural networks: convolutional neural networks (CNNs), dropout learning, autoencoders, extreme learning machine (ELM), graph embedding, network embedding, graph neural networks (GNNs), batch normalization networks, and generative adversarial networks (GANs).

Chapter 8 (Support Vector Machines) discusses the support vector machine regression and classification, and the relevance vector machine.

Chapter 9 (Evolutionary Computation) is concerned primarily with multiobjective optimization, multiobjective simulated annealing, multiobjective genetic algorithms, multiobjective evolutionary algorithms, evolutionary programming, differential evolution together with ant colony optimization, artificial bee colony algorithms, and particle swarm optimization. In particular, this chapter highlights also topics and advances in evolutionary computation: Pareto optimization theory, noisy multiobjective optimization, and opposition-based evolutionary computation.

Part I uses some revised content and materials from my book *Matrix Analysis and Applications* (Cambridge University Press, 2017), but there are considerable differences in content and book objective. This book is concentrated on the applications of the matrix algebra approaches in AI in Part II (561 pages of text), compared to Part I which is only 205 pages of text. This book is also related to my previous book *Linear Algebra in Signal Processing* (in Chinese, Science Press, Beijing, 1997; Japanese translation, Morikita Press, Tokyo, 2008) in some ideas.

Features and Contributions

- The first book on matrix algebra methods and applications in artificial intelligence.
- Introduces the machine learning tree, the neural network tree, and the evolutionary tree.

- Presents the solid matrix algebra theory and methods for four core AI areas: Machine Learning, Neural Networks, Support Vector Machines, and Evolutionary Computation.
- Highlights selected topics and advances of machine learning, neural networks, and evolutionary computation.
- Summarizes about 80 AI algorithms so that readers can further understand and implement AI methods.

Audience

This book is widely suitable for scientists, engineers, and graduate students in many disciplines, including but not limited to artificial intelligence, computer science, mathematics, engineering, etc.

Acknowledgments

I want to thank more than 40 of my former doctoral students and more than 20 graduate students for their cooperative research in intelligent signal and information processing and pattern recognition.

I am grateful to the several anonymous internal and external experts invited by Springer for their comments and suggestions. I am very grateful to my editor, Dr. Celine Lanlan Chang, Executive Editor, Computer Science, for her patience, understanding, and help in the course of my writing this book.

Finally, I would also like to thank my wife Xiao-Ying Tang for her consistent understanding and support for my work, teaching, and research in the past near 50 years.

Beijing, China
November, 2019

Xian-Da Zhang

A Note from the Family of Dr. Zhang

It is with a heavy heart we share with you that the author of this book, our beloved father, passed away before this book was published. We want to share with you some inspirational thoughts about his life's journey and how passionate he was about learning and teaching.

Our father endured many challenges in his life. During his high school years, the Great Chinese Famine occurred, but hunger could not deter him from studying. During his college years, our father's family was facing poverty, so he sought to help alleviate this by undergoing his studies at a military institution, where free meals were provided. Not long after, his tenacity to learn would be tested again, as the Cultural Revolution started, closing all the universities in China. He was assigned to work in a remote factory, but his perseverance to learn endured as he continued his education by studying hard secretly during off-hours.

After all the universities were reopened, our father left us to continue his study in a different city. He obtained his PhD degree at the age of 41 and became a professor at Tsinghua University in 1992.

Our father has taught and mentored many students both professionally and personally throughout his life. He is even more passionate in sharing his ideas and knowledge through writing as he believes that books, with its greater reach, will benefit many more people. He had been working for more than 12 h a day before he was admitted into the hospital. He planned to take a break after he finishes three books this year. Unfortunately, he could not handle such a heavy workload that he asked our help in editing this book in our last conversation.

Our father has lived a life with purpose. He discovered his great passion when he was young: learning and teaching. For him, it was even something to die for. He told our mom once that he would rather live fewer years to produce a high-quality book. Despite the numerous challenges and hardships he faced throughout his life, he never stopped learning and teaching. He self-studied Artificial Intelligence in the past few years and completed this book before his final days.

We sincerely hope you will enjoy reading his final work, as we believe that he would undoubtedly have been very happy to inform and teach and share with you his latest learning.

Fremont, CA, USA
Philadelphia, PA, USA
April, 2020

Yewei Zhang and Wei Wei (Daughter and Son In Law)
Zhang and John Zhang (Son and Grand Son)

Contents

Part I Introduction to Matrix Algebra

1	Basic Matrix Computation	3
1.1	Basic Concepts of Vectors and Matrices	3
1.1.1	Vectors and Matrices	3
1.1.2	Basic Vector Calculus	6
1.1.3	Basic Matrix Calculus	7
1.2	Sets and Linear Mapping	10
1.2.1	Sets	10
1.2.2	Linear Mapping	12
1.3	Norms	14
1.3.1	Vector Norms	14
1.3.2	Matrix Norms	18
1.4	Random Vectors	20
1.4.1	Statistical Interpretation of Random Vectors	20
1.4.2	Gaussian Random Vectors	24
1.5	Basic Performance of Matrices	27
1.5.1	Quadratic Forms	27
1.5.2	Determinants	27
1.5.3	Matrix Eigenvalues	30
1.5.4	Matrix Trace	32
1.5.5	Matrix Rank	33
1.6	Inverse Matrices and Moore–Penrose Inverse Matrices	35
1.6.1	Inverse Matrices	35
1.6.2	Left and Right Pseudo-Inverse Matrices	39
1.6.3	Moore–Penrose Inverse Matrices	40
1.7	Direct Sum and Hadamard Product	42
1.7.1	Direct Sum of Matrices	43
1.7.2	Hadamard Product	43

1.8	Kronecker Products	46
1.8.1	Definitions of Kronecker Products	46
1.8.2	Performance of Kronecker Products	47
1.9	Vectorization and Matricization	48
1.9.1	Vectorization and Commutation Matrix	48
1.9.2	Matricization of Vectors	50
	References	53
2	Matrix Differential	55
2.1	Jacobian Matrix and Gradient Matrix	55
2.1.1	Jacobian Matrix	55
2.1.2	Gradient Matrix	57
2.1.3	Calculation of Partial Derivative and Gradient	59
2.2	Real Matrix Differential	61
2.2.1	Calculation of Real Matrix Differential	62
2.2.2	Jacobian Matrix Identification	64
2.2.3	Jacobian Matrix of Real Matrix Functions	68
2.3	Complex Gradient Matrices	71
2.3.1	Holomorphic Function and Complex Partial Derivative ...	72
2.3.2	Complex Matrix Differential	75
2.3.3	Complex Gradient Matrix Identification	82
	References	87
3	Gradient and Optimization	89
3.1	Real Gradient	89
3.1.1	Stationary Points and Extreme Points	90
3.1.2	Real Gradient of $f(\mathbf{X})$	93
3.2	Gradient of Complex Variable Function	95
3.2.1	Extreme Point of Complex Variable Function	95
3.2.2	Complex Gradient	98
3.3	Convex Sets and Convex Function Identification	100
3.3.1	Standard Constrained Optimization Problems	100
3.3.2	Convex Sets and Convex Functions	102
3.3.3	Convex Function Identification	104
3.4	Gradient Methods for Smooth Convex Optimization	106
3.4.1	Gradient Method	106
3.4.2	Projected Gradient Method	108
3.4.3	Convergence Rates	110
3.5	Nesterov Optimal Gradient Method	112
3.5.1	Lipschitz Continuous Function	113
3.5.2	Nesterov Optimal Gradient Algorithms	115
3.6	Nonsmooth Convex Optimization	118
3.6.1	Subgradient and Subdifferential	119
3.6.2	Proximal Operator	123
3.6.3	Proximal Gradient Method	128

3.7	Constrained Convex Optimization	132
3.7.1	Penalty Function Method	133
3.7.2	Augmented Lagrange Multiplier Method	135
3.7.3	Lagrange Dual Method	137
3.7.4	Karush–Kuhn–Tucker Conditions	139
3.7.5	Alternating Direction Method of Multipliers	144
3.8	Newton Methods	147
3.8.1	Newton Method for Unconstrained Optimization	147
3.8.2	Newton Method for Constrained Optimization	150
	References	153
4	Solution of Linear Systems	157
4.1	Gauss Elimination	157
4.1.1	Elementary Row Operations	158
4.1.2	Gauss Elimination for Solving Matrix Equations	159
4.1.3	Gauss Elimination for Matrix Inversion	160
4.2	Conjugate Gradient Methods	162
4.2.1	Conjugate Gradient Algorithm	163
4.2.2	Biconjugate Gradient Algorithm	164
4.2.3	Preconditioned Conjugate Gradient Algorithm	165
4.3	Condition Number of Matrices	167
4.4	Singular Value Decomposition (SVD)	171
4.4.1	Singular Value Decomposition	171
4.4.2	Properties of Singular Values	173
4.4.3	Singular Value Thresholding	174
4.5	Least Squares Method	175
4.5.1	Least Squares Solution	175
4.5.2	Rank-Deficient Least Squares Solutions	178
4.6	Tikhonov Regularization and Gauss–Seidel Method	179
4.6.1	Tikhonov Regularization	179
4.6.2	Gauss–Seidel Method	181
4.7	Total Least Squares Method	184
4.7.1	Total Least Squares Solution	184
4.7.2	Performances of TLS Solution	189
4.7.3	Generalized Total Least Square	191
4.8	Solution of Under-Determined Systems	193
4.8.1	ℓ_1 -Norm Minimization	193
4.8.2	Lasso	196
4.8.3	LARS	197
	References	199
5	Eigenvalue Decomposition	203
5.1	Eigenvalue Problem and Characteristic Equation	203
5.1.1	Eigenvalue Problem	203
5.1.2	Characteristic Polynomial	205

5.2	Eigenvalues and Eigenvectors	206
5.2.1	Eigenvalues	206
5.2.2	Eigenvectors	208
5.3	Generalized Eigenvalue Decomposition (GEVD).....	210
5.3.1	Generalized Eigenvalue Decomposition	210
5.3.2	Total Least Squares Method for GEVD.....	214
5.4	Rayleigh Quotient and Generalized Rayleigh Quotient.....	214
5.4.1	Rayleigh Quotient.....	215
5.4.2	Generalized Rayleigh Quotient.....	216
5.4.3	Effectiveness of Class Discrimination	217
	References	220

Part II Artificial Intelligence

6	Machine Learning	223
6.1	Machine Learning Tree	223
6.2	Optimization in Machine Learning	226
6.2.1	Single-Objective Composite Optimization	226
6.2.2	Gradient Aggregation Methods	230
6.2.3	Coordinate Descent Methods	232
6.2.4	Benchmark Functions for Single-Objective Optimization	236
6.3	Majorization-Minimization Algorithms	241
6.3.1	MM Algorithm Framework.....	242
6.3.2	Examples of Majorization-Minimization Algorithms	244
6.4	Boosting and Probably Approximately Correct Learning	247
6.4.1	Boosting for Weak Learners	248
6.4.2	Probably Approximately Correct Learning	250
6.5	Basic Theory of Machine Learning	253
6.5.1	Learning Machine.....	253
6.5.2	Machine Learning Methods	254
6.5.3	Expected Performance of Machine Learning Algorithms	256
6.6	Classification and Regression	256
6.6.1	Pattern Recognition and Classification.....	257
6.6.2	Regression	259
6.7	Feature Selection	260
6.7.1	Supervised Feature Selection	261
6.7.2	Unsupervised Feature Selection	264
6.7.3	Nonlinear Joint Unsupervised Feature Selection	266
6.8	Principal Component Analysis	269
6.8.1	Principal Component Analysis Basis	269
6.8.2	Minor Component Analysis	270
6.8.3	Principal Subspace Analysis.....	271

6.8.4	Robust Principal Component Analysis	276
6.8.5	Sparse Principal Component Analysis	278
6.9	Supervised Learning Regression	281
6.9.1	Principle Component Regression	282
6.9.2	Partial Least Squares Regression	285
6.9.3	Penalized Regression	289
6.9.4	Gradient Projection for Sparse Reconstruction	293
6.10	Supervised Learning Classification	296
6.10.1	Binary Linear Classifiers	296
6.10.2	Multiclass Linear Classifiers	298
6.11	Supervised Tensor Learning (STL)	301
6.11.1	Tensor Algebra Basics	301
6.11.2	Supervised Tensor Learning Problems	307
6.11.3	Tensor Fisher Discriminant analysis	309
6.11.4	Tensor Learning for Regression	311
6.11.5	Tensor K-Means Clustering	314
6.12	Unsupervised Clustering	314
6.12.1	Similarity Measures	316
6.12.2	Hierarchical Clustering	320
6.12.3	Fisher Discriminant Analysis (FDA)	324
6.12.4	K-Means Clustering	327
6.13	Spectral Clustering	330
6.13.1	Spectral Clustering Algorithms	330
6.13.2	Constrained Spectral Clustering	333
6.13.3	Fast Spectral Clustering	335
6.14	Semi-Supervised Learning Algorithms	337
6.14.1	Semi-Supervised Inductive/Transductive Learning	338
6.14.2	Self-Training	340
6.14.3	Co-training	341
6.15	Canonical Correlation Analysis	343
6.15.1	Canonical Correlation Analysis Algorithm	344
6.15.2	Kernel Canonical Correlation Analysis	347
6.15.3	Penalized Canonical Correlation Analysis	352
6.16	Graph Machine Learning	354
6.16.1	Graphs	354
6.16.2	Graph Laplacian Matrices	358
6.16.3	Graph Spectrum	360
6.16.4	Graph Signal Processing	363
6.16.5	Semi-Supervised Graph Learning: Harmonic Function Method	368
6.16.6	Semi-Supervised Graph Learning: Min-Cut Method	371
6.16.7	Unsupervised Graph Learning: Sparse Coding Method	377
6.17	Active Learning	378
6.17.1	Active Learning: Background	379
6.17.2	Statistical Active Learning	380

6.17.3	Active Learning Algorithms	382
6.17.4	Active Learning Based Binary Linear Classifiers	383
6.17.5	Active Learning Using Extreme Learning Machine.....	384
6.18	Reinforcement Learning	387
6.18.1	Basic Concepts and Theory.....	387
6.18.2	Markov Decision Process (MDP)	391
6.19	Q-Learning	394
6.19.1	Basic Q-Learning	394
6.19.2	Double Q-Learning and Weighted Double Q-Learning	396
6.19.3	Online Connectionist Q-Learning Algorithm.....	399
6.19.4	Q-Learning with Experience Replay	400
6.20	Transfer Learning	402
6.20.1	Notations and Definitions.....	403
6.20.2	Categorization of Transfer Learning	406
6.20.3	Boosting for Transfer Learning	410
6.20.4	Multitask Learning	411
6.20.5	EigenTransfer	414
6.21	Domain Adaptation	417
6.21.1	Feature Augmentation Method	418
6.21.2	Cross-Domain Transform Method	421
6.21.3	Transfer Component Analysis Method	423
	References	427
7	Neural Networks	441
7.1	Neural Network Tree	441
7.2	From Modern Neural Networks to Deep Learning	444
7.3	Optimization of Neural Networks	445
7.3.1	Online Optimization Problems	446
7.3.2	Adaptive Gradient Algorithm	447
7.3.3	Adaptive Moment Estimation	449
7.4	Activation Functions.....	452
7.4.1	Logistic Regression and Sigmoid Function.....	452
7.4.2	Softmax Regression and Softmax Function	454
7.4.3	Other Activation Functions	456
7.5	Recurrent Neural Networks	460
7.5.1	Conventional Recurrent Neural Networks	460
7.5.2	Backpropagation Through Time (BPTT)	463
7.5.3	Jordan Network and Elman Network	467
7.5.4	Bidirectional Recurrent Neural Networks	468
7.5.5	Long Short-Term Memory (LSTM).....	471
7.5.6	Improvement of Long Short-Term Memory	474
7.6	Boltzmann Machines	477
7.6.1	Hopfield Network and Boltzmann Machines	477
7.6.2	Restricted Boltzmann Machine.....	481

7.6.3	Contrastive Divergence Learning	484
7.6.4	Multiple Restricted Boltzmann Machines	487
7.7	Bayesian Neural Networks	490
7.7.1	Naive Bayesian Classification	490
7.7.2	Bayesian Classification Theory	491
7.7.3	Sparse Bayesian Learning	493
7.8	Convolutional Neural Networks	496
7.8.1	Hankel Matrix and Convolution	498
7.8.2	Pooling Layer	504
7.8.3	Activation Functions in CNNs	507
7.8.4	Loss Function	510
7.9	Dropout Learning	514
7.9.1	Dropout for Shallow and Deep Learning	515
7.9.2	Dropout Spherical K-Means	518
7.9.3	DropConnect	520
7.10	Autoencoders	524
7.10.1	Basic Autoencoder	525
7.10.2	Stacked Sparse Autoencoder	532
7.10.3	Stacked Denoising Autoencoders	535
7.10.4	Convolutional Autoencoders (CAE)	538
7.10.5	Stacked Convolutional Denoising Autoencoder	539
7.10.6	Nonnegative Sparse Autoencoder	540
7.11	Extreme Learning Machine	542
7.11.1	Single-Hidden Layer Feedforward Networks with Random Hidden Nodes	543
7.11.2	Extreme Learning Machine Algorithm for Regression and Binary Classification	545
7.11.3	Extreme Learning Machine Algorithm for Multiclass Classification	549
7.12	Graph Embedding	551
7.12.1	Proximity Measures and Graph Embedding	552
7.12.2	Multidimensional Scaling	557
7.12.3	Manifold Learning: Isometric Map	559
7.12.4	Manifold Learning: Locally Linear Embedding	560
7.12.5	Manifold Learning: Laplacian Eigenmap	563
7.13	Network Embedding	567
7.13.1	Structure and Property Preserving Network Embedding	568
7.13.2	Community Preserving Network Embedding	569
7.13.3	Higher-Order Proximity Preserved Network Embedding	573
7.14	Neural Networks on Graphs	576
7.14.1	Graph Neural Networks (GNNs)	577
7.14.2	DeepWalk and GraphSAGE	579
7.14.3	Graph Convolutional Networks (GCNs)	583

7.15	Batch Normalization Networks	588
7.15.1	Batch Normalization	588
7.15.2	Variants and Extensions of Batch Normalization.....	593
7.16	Generative Adversarial Networks (GANs)	598
7.16.1	Generative Adversarial Network Framework.....	598
7.16.2	Bidirectional Generative Adversarial Networks	602
7.16.3	Variational Autoencoders	604
	References	607
8	Support Vector Machines	617
8.1	Support Vector Machines: Basic Theory	617
8.1.1	Statistical Learning Theory	618
8.1.2	Linear Support Vector Machines	621
8.2	Kernel Regression Methods	624
8.2.1	Reproducing Kernel and Mercer Kernel	624
8.2.2	Representer Theorem and Kernel Regression	628
8.2.3	Semi-Supervised and Graph Regression	630
8.2.4	Kernel Partial Least Squares Regression.....	632
8.2.5	Laplacian Support Vector Machines	633
8.3	Support Vector Machine Regression.....	635
8.3.1	Support Vector Machine Regressor	635
8.3.2	ϵ -Support Vector Regression	636
8.3.3	ν -Support Vector Machine Regression.....	639
8.4	Support Vector Machine Binary Classification.....	641
8.4.1	Support Vector Machine Binary Classifier.....	642
8.4.2	ν -Support Vector Machine Binary Classifier	645
8.4.3	Least Squares SVM Binary Classifier.....	647
8.4.4	Proximal Support Vector Machine Binary Classifier	649
8.4.5	SVM-Recursive Feature Elimination	651
8.5	Support Vector Machine Multiclass Classification.....	653
8.5.1	Decomposition Methods for Multiclass Classification.....	653
8.5.2	Least Squares SVM Multiclass Classifier.....	656
8.5.3	Proximal Support Vector Machine Multiclass Classifier...	659
8.6	Gaussian Process for Regression and Classification	660
8.6.1	Joint, Marginal, and Conditional Probabilities	661
8.6.2	Gaussian Process	661
8.6.3	Gaussian Process Regression	663
8.6.4	Gaussian Process Classification	666
8.7	Relevance Vector Machine	667
8.7.1	Sparse Bayesian Regression	668
8.7.2	Sparse Bayesian Classification	672
8.7.3	Fast Marginal Likelihood Maximization.....	673
	References	677

9	Evolutionary Computation	681
9.1	Evolutionary Computation Tree	681
9.2	Multiobjective Optimization	683
9.2.1	Multiobjective Combinatorial Optimization	684
9.2.2	Multiobjective Optimization Problems	686
9.3	Pareto Optimization Theory	691
9.3.1	Pareto Concepts	691
9.3.2	Fitness Selection Approach	698
9.3.3	Nondominated Sorting Approach	700
9.3.4	Crowding Distance Assignment Approach	702
9.3.5	Hierarchical Clustering Approach	703
9.3.6	Benchmark Functions for Multiobjective Optimization	704
9.4	Noisy Multiobjective Optimization	707
9.4.1	Pareto Concepts for Noisy Multiobjective Optimization	708
9.4.2	Performance Metrics for Approximation Sets	712
9.5	Multiobjective Simulated Annealing	714
9.5.1	Principle of Simulated Annealing	714
9.5.2	Multiobjective Simulated Annealing Algorithm	718
9.5.3	Archived Multiobjective Simulated Annealing	721
9.6	Genetic Algorithm	723
9.6.1	Basic Genetic Algorithm Operations	724
9.6.2	Genetic Algorithm with Gene Rearrangement Clustering	729
9.7	Nondominated Multiobjective Genetic Algorithms	733
9.7.1	Fitness Functions	733
9.7.2	Fitness Selection	734
9.7.3	Nondominated Sorting Genetic Algorithms	736
9.7.4	Elitist Nondominated Sorting Genetic Algorithm	739
9.8	Evolutionary Algorithms (EAs)	740
9.8.1	(1 + 1) Evolutionary Algorithm	741
9.8.2	Theoretical Analysis on Evolutionary Algorithms	742
9.9	Multiobjective Evolutionary Algorithms	743
9.9.1	Classical Methods for Solving Multiobjective Optimization Problems	743
9.9.2	MOEA Based on Decomposition (MOEA/D)	745
9.9.3	Strength Pareto Evolutionary Algorithm	749
9.9.4	Achievement Scalarizing Functions	754
9.10	Evolutionary Programming	758
9.10.1	Classical Evolutionary Programming	758
9.10.2	Fast Evolutionary Programming	759
9.10.3	Hybrid Evolutionary Programming	761
9.11	Differential Evolution	763
9.11.1	Classical Differential Evolution	763
9.11.2	Differential Evolution Variants	765

9.12	Ant Colony Optimization	767
9.12.1	Real Ants and Artificial Ants	768
9.12.2	Typical Ant Colony Optimization Problems.....	771
9.12.3	Ant System and Ant Colony System	773
9.13	Multiobjective Artificial Bee Colony Algorithms.....	776
9.13.1	Artificial Bee Colony Algorithms	776
9.13.2	Variants of ABC Algorithms	778
9.14	Particle Swarm Optimization	780
9.14.1	Basic Concepts	780
9.14.2	The Canonical Particle Swarm	781
9.14.3	Genetic Learning Particle Swarm Optimization	783
9.14.4	Particle Swarm Optimization for Feature Selection	785
9.15	Opposition-Based Evolutionary Computation	788
9.15.1	Opposition-Based Learning	788
9.15.2	Opposition-Based Differential Evolution	790
9.15.3	Two Variants of Opposition-Based Learning	791
	References	795
	Index	805

List of Notations

\forall	For all
$ $	Such that
\ni	Such that
\exists	There exists
\nexists	There does not exist
\wedge	Logical AND
\vee	Logical OR
$ A $	Cardinality of a set A
$A \Rightarrow B$	“condition A results in B ” or “ A implies B ”
$A \subseteq B$	A is a subset of B
$A \subset B$	A is a proper subset of B
$A = B$	Sets $A = B$
$A \cup B$	Union of sets A and B
$A \cap B$	Intersection of sets A and B
$A \cap B = \emptyset$	Sets A and B are disjoint
$A + B$	Sum set of sets A and B
$A - B$	The set of elements of A that are not in B
$X \setminus A$	Complement of the set A in the set X
$A > B$	Set A dominates set B : every $\mathbf{f}(\mathbf{x}_2) \in B$ is dominated by at least one $\mathbf{f}(\mathbf{x}_1) \in A$ such that $\mathbf{f}(\mathbf{x}_1) <_{IN} \mathbf{f}(\mathbf{x}_2)$ (for minimization) or $\mathbf{f}(\mathbf{x}_1) >_{IN} \mathbf{f}(\mathbf{x}_2)$ (for maximization)
$A \geq B$	A weakly dominates B : for every $\mathbf{f}(\mathbf{x}_2) \in B$ and at least one $\mathbf{f}(\mathbf{x}_1) \in A$ $\mathbf{f}(\mathbf{x}_1) \leq_{IN} \mathbf{f}(\mathbf{x}_2)$ (for minimization) or $\mathbf{f}(\mathbf{x}_1) \geq_{IN} \mathbf{f}(\mathbf{x}_2)$ (for maximization)
$A >> B$	A strictly dominates B : every $\mathbf{f}(\mathbf{x}_2) \in B$ is strictly dominated by at least one $\mathbf{f}(\mathbf{x}_1) \in A$ such that $f_i(\mathbf{x}_1) <_{IN} f_i(\mathbf{x}_2), \forall i = \{1, \dots, m\}$ (for minimization) or $f_i(\mathbf{x}_1) >_{IN} f_i(\mathbf{x}_2), \forall i = \{1, \dots, m\}$ (for maximization)
$A \parallel B$	Sets A and B are incomparable: neither $A \geq B$ nor $B \geq A$
$A \triangleright B$	A is better than B : every $\mathbf{f}(\mathbf{x}_2) \in B$ is weakly dominated by at least one $\mathbf{f}(\mathbf{x}_1) \in A$ and $A \neq B$

$\text{AGG}_{\text{mean}}(z)$	Mean aggregate function of z
$\text{AGG}_{\text{LSTM}}(z)$	LSTM aggregate function of z
$\text{AGG}_{\text{pool}}(z)$	Pooling aggregate function of z
\mathbb{C}	Complex numbers
\mathbb{C}^n	Complex n -vector
$\mathbb{C}^{m \times n}$	Complex $m \times n$ matrix
$\mathbb{C}[x]$	Complex polynomial
$\mathbb{C}[x]^{m \times n}$	Complex $m \times n$ polynomial matrix
$\mathbb{C}^{I \times J \times K}$	Complex third-order tensors
$\mathbb{C}^{I_1 \times \dots \times I_N}$	Complex N -order tensor
\mathbb{K}	Real or complex number
\mathbb{K}^n	Real or complex n -vector
$\mathbb{K}^{m \times n}$	Real or complex $m \times n$ matrix
$\mathbb{K}^{I \times J \times K}$	Real or complex third-order tensor
$\mathbb{K}^{I_1 \times \dots \times I_N}$	Real or complex N -order tensor
$G(V, E, \mathbf{W})$	Graph with vertex set V , edge set E and adjacency matrix \mathbf{W}
$\mathcal{N}(v)$	Neighbors of a vertex (node) v
$\text{PReLU}(z)$	Parametric rectified linear unit activation function of z
$\text{ReLU}(z)$	Rectified linear unit activation function of z
\mathbb{R}	Real number
\mathbb{R}^n	Real n -vectors ($n \times 1$ real matrix)
$\mathbb{R}^{m \times n}$	Real $m \times n$ matrix
$\mathbb{R}[x]$	Real polynomial
$\mathbb{R}[x]^{m \times n}$	Real $m \times n$ polynomial matrix
$\mathbb{R}^{I \times J \times K}$	Real third-order tensors
$\mathbb{R}^{I_1 \times \dots \times I_N}$	Real N -order tensor
\mathbb{R}_+	Nonnegative real numbers, nonnegative orthant
\mathbb{R}_{++}	Positive real number
$\sigma(z)$	Sigmoid activation function of z
$\text{softmax}(z)$	Softmax activation function of z
$\text{softplus}(z)$	Softplus activation function of z
$\text{softsign}(z)$	Softsign activation function of z
$\tanh(z)$	Tangent (tanh) hyperbolic activation function of z
$T : V \rightarrow W$	Mapping the vectors in V to corresponding vectors in W
$T^{-1} : W \rightarrow V$	Inverse mapping of the one-to-one mapping $T : V \rightarrow W$
$X_1 \times \dots \times X_n$	Cartesian product of n sets X_1, \dots, X_n
$\{(\mathbf{x}_i, y_i = +1)\}$	Set of training data vectors \mathbf{x}_i belonging to the classes (+)
$\{(\mathbf{x}_i, y_i = -1)\}$	Set of training data vectors \mathbf{x}_i belonging to the classes (-)
$\mathbf{1}_n$	n -dimensional summing vector with all entries 1
$\mathbf{0}_n$	n -dimensional zero vector with all zero entries
\mathbf{e}_i	Base vector whose i th entry equal to 1 and others being zero
$\mathbf{x} \sim N(\bar{\mathbf{x}}, \mathbf{\Gamma}_x)$	Gaussian vector with mean vector $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{\Gamma}_x$
$\ \mathbf{x}\ _0$	ℓ_0 -norm: number of nonzero entries of vector \mathbf{x}
$\ \mathbf{x}\ _1$	ℓ_1 -norm of vector \mathbf{x}
$\ \mathbf{x}\ _2$	Euclidean form of vector \mathbf{x}

$\ \mathbf{x}\ _p$	ℓ_p -norm or Hölder norm of vector \mathbf{x}
$\ \mathbf{x}\ _*$	Nuclear norm of vector \mathbf{x}
$\ \mathbf{x}\ _\infty$	ℓ_∞ -norm of vector \mathbf{x}
$\langle \mathbf{x}, \mathbf{y} \rangle$	Inner product of vectors \mathbf{x} and \mathbf{y}
$d(\mathbf{x}, \mathbf{y})$	Distance or dissimilarity between vectors \mathbf{x} and \mathbf{y}
$N_\epsilon(\mathbf{x})$	ϵ -neighborhood of vector \mathbf{x}
$\rho(\mathbf{x}, \mathbf{y})$	Correlation coefficient between two random vectors \mathbf{x} and \mathbf{y}
$\mathbf{x} \in A$	\mathbf{x} belongs to the set A , i.e., \mathbf{x} is an element or member of A
$\mathbf{x} \notin A$	\mathbf{x} is not an element of the set A
$\mathbf{x} \circ \mathbf{y} = \mathbf{xy}^H$	Outer product of vectors \mathbf{x} and \mathbf{y}
$\mathbf{x} \perp \mathbf{y}$	Vector orthogonal
$\mathbf{x} > 0$	Positive vector with components $x_i > 0, \forall i$
$\mathbf{x} \geq 0$	Nonnegative vector with components $x_i \geq 0, \forall i$
$\mathbf{x} \geq \mathbf{y}$	Vector elementwise inequality $x_i \geq y_i, \forall i$
$\mathbf{x} \succ \mathbf{x}'$	\mathbf{x} domains (or outperforms) $\mathbf{x}' : \mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}')$ for minimization
$\mathbf{x} \succ' \mathbf{x}'$	\mathbf{x} domains (or outperforms) $\mathbf{x}' : \mathbf{f}(\mathbf{x}) > \mathbf{f}(\mathbf{x}')$ for maximization
$\mathbf{x} \succeq \mathbf{x}'$	\mathbf{x} weakly dominates $\mathbf{x}' : \mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}')$ for minimization
$\mathbf{x} \succeq' \mathbf{x}'$	\mathbf{x} weakly dominates $\mathbf{x}' : \mathbf{f}(\mathbf{x}) \geq \mathbf{f}(\mathbf{x}')$ for maximization
$\mathbf{x} \succ \succ \mathbf{x}'$	\mathbf{x} strictly dominates $\mathbf{x}' : f_i(\mathbf{x}) < f_i(\mathbf{x}'), \forall i$ for minimization
$\mathbf{x} \succ \succ' \mathbf{x}'$	\mathbf{x} strictly dominates $\mathbf{x}' : f_i(\mathbf{x}) > f_i(\mathbf{x}'), \forall i$ for maximization
$\mathbf{x} \parallel \mathbf{x}'$	\mathbf{x} and \mathbf{x}' are incomparable, i.e., $\mathbf{x} \not\preceq \mathbf{x}' \wedge \mathbf{x}' \not\preceq \mathbf{x}$
$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}')$	$f_i(\mathbf{x}) = f_i(\mathbf{x}'), \forall i = 1, \dots, m$
$\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}')$	$f_i(\mathbf{x}) \neq f_i(\mathbf{x}')$, for at least one $i \in \{1, \dots, m\}$
$\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}')$	$f_i(\mathbf{x}) \leq f_i(\mathbf{x}'), \forall i = 1, \dots, m$
$\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}')$	$\forall i = 1, \dots, m : f_i(\mathbf{x}) \leq f_i(\mathbf{x}') \wedge \exists j \in \{1, \dots, m\} : f_j(\mathbf{x}) < f_j(\mathbf{x}')$
$\mathbf{f}(\mathbf{x}) \geq \mathbf{f}(\mathbf{x}')$	$f_i(\mathbf{x}) \geq f_i(\mathbf{x}'), \forall i = 1, \dots, m$
$\mathbf{f}(\mathbf{x}) > \mathbf{f}(\mathbf{x}')$	$\forall i = 1, \dots, m : f_i(\mathbf{x}) \geq f_i(\mathbf{x}') \wedge \exists j \in \{1, \dots, m\} : f_j(\mathbf{x}) > f_j(\mathbf{x}')$
$\mathbf{f}(\mathbf{x}_1) <_{IN} \mathbf{f}(\mathbf{x}_2)$	Interval order relation: $\forall i = 1, \dots, m : \underline{f}_i(\mathbf{x}_1) \leq \underline{f}_i(\mathbf{x}_2) \wedge \overline{f}_i(\mathbf{x}_1) \leq \overline{f}_i(\mathbf{x}_2) \wedge \exists j \in \{1, \dots, m\} : \underline{f}_j(\mathbf{x}_1) \neq \underline{f}_j(\mathbf{x}_2) \vee \overline{f}_j(\mathbf{x}_1) \neq \overline{f}_j(\mathbf{x}_2)$
$\mathbf{f}(\mathbf{x}_1) >_{IN} \mathbf{f}(\mathbf{x}_2)$	Interval order relation: $\forall i = 1, \dots, m : \underline{f}_i(\mathbf{x}_1) \geq \underline{f}_i(\mathbf{x}_2) \wedge \overline{f}_i(\mathbf{x}_1) \geq \overline{f}_i(\mathbf{x}_2) \wedge \exists j \in \{1, \dots, m\} : \underline{f}_j(\mathbf{x}_1) \neq \underline{f}_j(\mathbf{x}_2) \vee \overline{f}_j(\mathbf{x}_1) \neq \overline{f}_j(\mathbf{x}_2)$
$\mathbf{f}(\mathbf{x}_1) \leq_{IN} \mathbf{f}(\mathbf{x}_2)$	Weak interval order relation: $\forall i \in \{1, \dots, m\} : \underline{f}_i(\mathbf{x}_1) \leq \underline{f}_i(\mathbf{x}_2) \wedge \overline{f}_i(\mathbf{x}_1) \leq \overline{f}_i(\mathbf{x}_2)$
$\mathbf{f}(\mathbf{x}_1) \geq_{IN} \mathbf{f}(\mathbf{x}_2)$	Weak interval order relation: $\forall i \in \{1, \dots, m\} : \underline{f}_i(\mathbf{x}_1) \geq \underline{f}_i(\mathbf{x}_2) \wedge \overline{f}_i(\mathbf{x}_1) \geq \overline{f}_i(\mathbf{x}_2)$
\mathbf{A}^T	Transpose of matrix \mathbf{A}
\mathbf{A}^H	Complex conjugate transpose of matrix \mathbf{A}
\mathbf{A}^{-1}	Inverse of nonsingular matrix \mathbf{A}
\mathbf{A}^\dagger	Moore–Penrose inverse of matrix \mathbf{A}

\mathbf{A}^*	Conjugate of matrix \mathbf{A}
$\mathbf{A} \succ 0$	Positive definite matrix
$\mathbf{A} \succeq 0$	Positive semi-definite matrix
$\mathbf{A} \prec 0$	Negative definite matrix
$\mathbf{A} \preceq 0$	Negative semi-definite matrix
$\mathbf{A} \succ 0$	Positive (or elementwise positive) matrix
$\mathbf{A} \succeq 0$	Nonnegative (or elementwise nonnegative) matrix
$\mathbf{A} \succeq \mathbf{B}$	Matrix elementwise inequality $a_{ij} \geq b_{ij}, \forall i, j$
\mathbf{I}_n	$n \times n$ Identity matrix
\mathbf{O}_n	$n \times n$ Null matrix
$ \mathbf{A} $	Determinant of matrix \mathbf{A}
$\ \mathbf{A}\ _1$	Maximum absolute column-sum norm of matrix \mathbf{A}
$\ \mathbf{A}\ _2 = \ \mathbf{A}\ _{\text{spec}}$	Spectrum norm of matrix \mathbf{A}
$\ \mathbf{A}\ _F$	Frobenius norm of matrix \mathbf{A}
$\ \mathbf{A}\ _\infty$	Max norm of \mathbf{A} : absolute maximum of all entries of \mathbf{A}
$\ \mathbf{A}\ _{\mathbf{G}}$	Mahalanobis norm of matrix \mathbf{A}
$\ \mathbf{A}\ _*$	Nuclear norm, called also the trace norm, of matrix \mathbf{A}
$\mathbf{A} \oplus \mathbf{B}$	Direct sum of an $m \times m$ matrix \mathbf{A} and an $n \times n$ matrix \mathbf{B}
$\mathbf{A} \odot \mathbf{B}$	Hadamard product (or elementwise product) of \mathbf{A} and \mathbf{B}
$\mathbf{A} \oslash \mathbf{B}$	Elementwise division of matrices \mathbf{A} and \mathbf{B}
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of matrices \mathbf{A} and \mathbf{B}
$\langle \mathbf{A}, \mathbf{B} \rangle$	Inner (or dot) product of \mathbf{A} and \mathbf{B} : $\langle \mathbf{A}, \mathbf{B} \rangle = \langle \text{vec}(\mathbf{A}), \text{vec}(\mathbf{B}) \rangle$
$\rho(\mathbf{A})$	Spectral radius of matrix \mathbf{A}
$\text{cond}(\mathbf{A})$	Condition number of matrix \mathbf{A}
$\text{diag}(\mathbf{A})$	Diagonal function of $\mathbf{A} = [a_{ij}]$: $\sum_{i=1}^n a_{ii} ^2$
$\mathbf{Diag}(\mathbf{A})$	Diagonal matrix consisting of diagonal entries of \mathbf{A}
$\text{eig}(\mathbf{A})$	Eigenvalues of the Hermitian matrix \mathbf{A}
$\text{Gr}(n, r)$	Grassmann manifold
$\text{rvec}(\mathbf{A})$	Row vectorization of matrix \mathbf{A}
$\text{off}(\mathbf{A})$	Off function of $\mathbf{A} = [a_{ij}]$: $\sum_{i=1, i \neq j}^m \sum_{j=1}^n a_{ij} ^2$
$\text{tr}(\mathbf{A})$	Trace of matrix \mathbf{A}
$\text{vec}(\mathbf{A})$	Vectorization of matrix \mathbf{A}

List of Figures

Fig. 1.1	Classification of vectors	5
Fig. 6.1	Machine learning tree.....	225
Fig. 6.2	Three-way array (left) and a tensor modeling a face (right)	301
Fig. 6.3	An MDP models the synchronous interaction between agent and environment	391
Fig. 6.4	One neural network as a collection of interconnected units, where i , j , and k are neurons in output layer, hidden layer, and input layer, respectively	399
Fig. 6.5	Transfer learning setup: storing knowledge gained from solving one problem and applying it to a different but related problem	405
Fig. 6.6	The transformation mapping. (a) The symmetric transformation (T_S and T_T) of the source-domain feature set $X_S = \{\mathbf{x}_{S_i}\}$ and target-domain feature set $X_T = \{\mathbf{x}_{T_i}\}$ into a common latent feature set $X_C = \{\mathbf{x}_{C_i}\}$. (b) The asymmetric transformation T_T of the source-domain feature set X_S to the target-domain feature set X_T	407
Fig. 7.1	Neural network tree.....	444
Fig. 7.2	General structure of a regular unidirectional three-layer RNN with hidden state recurrence, where z^{-1} denotes a delay line. Left is RNN with a delay line, and right is RNN unfolded in time for two time steps	462
Fig. 7.3	General structure of a regular unidirectional three-layer RNN with output recurrence, where z^{-1} denotes a delay line. Left is RNN with a delay line, and right is RNN unfolded in time for two time steps	463

Fig. 7.4	Illustration of Jordan network	469
Fig. 7.5	Illustration of Elman network	469
Fig. 7.6	Illustration of the Bidirectional Recurrent Neural Network (BRNN) unfolded in time for three time steps. Upper: output units, Middle: forward and backward hidden units, Lower: input units	470
Fig. 7.7	A comparison between Hopfield network and Boltzmann machine. Left: is Hopfield network in which a fully connected network of six binary thresholding neural units. Every unit is connected with data. Right: is Boltzmann machine whose model splits into two parts: visible units and hidden units (shaded nodes). The dashed line is used to highlight the model separation	478
Fig. 7.8	A comparison between Left Boltzmann machine and Right restricted Boltzmann machine. In the restricted Boltzmann machine there are no connections between hidden units (shaded nodes) and no connections between visible units (unshaded nodes)	481
Fig. 7.9	A toy example using four different pooling techniques. Left: resulting activations within a given pooling region. Right: pooling results given by four different pooling techniques. If $\lambda = 0.4$ is taken then the mixed pooling result is 1.46. The mixed pooling and the stochastic pooling can represent multi-modal distributions of activations within a region	508
Fig. 7.10	An example of a thinned network produced by applying dropout to a full-connection neural network. Crossed units have been dropped. The dropout probabilities of the first, second, and third layers are 0.4, 0.6, and 0.4, respectively,	516
Fig. 7.11	An example of the structure of DropConnect. The dotted lines show dropped connections	521
Fig. 7.12	MLP with one hidden layer for auto association. The shape of the whole network is similar to an hourglass	525
Fig. 7.13	Illustration of the architecture of the basic autoencoder with “encoder” and “decoder” networks for high-level feature learning. The leftmost layer of the whole network is called the input layer, and the rightmost layer the output layer. The middle layer of nodes is called the hidden layer, because its values are not observed in the training set. The circles labeled “+1” are called bias units, and correspond to the intercept term b . I_i denotes the index i , $i = 1, \dots, n$	526

Fig. 7.14	Autoencoder training flowchart. Encoder f encodes the input \mathbf{x} to $\mathbf{h} = f(\mathbf{x})$, and the decoder g decodes $\mathbf{h} = f(\mathbf{x})$ to the output $\mathbf{y} = \hat{\mathbf{x}} = g(f(\mathbf{x}))$ so that the reconstruction error $L(\mathbf{x}, \mathbf{y})$ is minimized	527
Fig. 7.15	Stacked autoencoder scheme with 2 hidden layers, where $\mathbf{h}^{(d)} = f(\mathbf{W}_1^{(d)}\mathbf{x}^{(d)} + \mathbf{b}_1^{(d)})$ and $\mathbf{y}^{(d)} = f(\mathbf{W}_2^{(d)}\mathbf{h}^{(d)} + \mathbf{b}_2^{(d)})$, $d = 1, 2$ with $\mathbf{x}^{(1)} = \mathbf{x}$, $\mathbf{x}^{(2)} = \mathbf{y}^{(1)}$ and $\hat{\mathbf{x}} = \mathbf{y}^{(2)}$	533
Fig. 7.16	Comparison between the two network architectures for a batch data: (a) the network with no BatchNorm layer; (b) the same network as in (a) with a BatchNorm layer inserted after the fully connected layer \mathbf{W} . All the layer parameters have exactly the same value in both networks, and the two networks have the same loss function \mathcal{L} , i.e., $\hat{\mathcal{L}} = \mathcal{L}$	590
Fig. 7.17	The basic structure of generative adversarial networks (GANs). The generator G uses a random noise \mathbf{z} to generate a synthetic data $\hat{\mathbf{x}} = G(\mathbf{z})$, and the discriminator D tries to identify whether the synthesized data $\hat{\mathbf{x}}$ is a real data \mathbf{x} , i.e., making a real or fake inference	599
Fig. 7.18	Bidirectional generative adversarial network (BiGAN) is a combination of a standard GAN and an Encoder. The generator G from the standard GAN framework maps latent samples \mathbf{z} to generated data $G(\mathbf{z})$. An encoder E maps data \mathbf{x} to the output $E(\mathbf{x})$. Both the generator tuple $(G(\mathbf{z}), \mathbf{z})$ and the encoder tuple $(\mathbf{x}, E(\mathbf{x}))$ act, in a bidirectional way, as inputs to the discriminator D	603
Fig. 7.19	A variational autoencoder (VAEs) connected with a standard generative adversarial network (GAN). Unlike the BiGAN in which the encoder's output $E(\mathbf{x})$ is one of two bidirectional inputs, the autoencoder E here executes variation $\mathbf{z} \sim E(\mathbf{x})$ that is used as the random noise in GAN	604
Fig. 8.1	Three points in \mathbb{R}^2 , shattered by oriented lines. The arrow points to the side with the points labeled black	620
Fig. 8.2	The decision directed acyclic graphs (DDAG) for finding the best class out of four classes	656
Fig. 9.1	Evolutionary computation tree	683

Fig. 9.2	Pareto efficiency and Pareto improvement. Point A is an inefficient allocation between preference criterions f_1 and f_2 because it does not satisfy the constraint curve of f_1 and f_2 . Two decisions to move from Point A to Points C and D would be a Pareto improvement, respectively. They improve both f_1 and f_2 , without making anyone else worse off. Hence, these two moves would be a Pareto improvement and be Pareto optimal, respectively. A move from Point A to Point B would not be a Pareto improvement because it decreases the cost f_1 by increasing another cost f_2 , thus making one side better off by making another worse off. The move from any point that lies under the curve to any point on the curve cannot be a Pareto improvement due to making one of two criterions f_1 and f_2 worse	695
Fig. 9.3	Illustration of degeneracy due to crossover (from [16]). Here, asterisk denotes three cluster centers (1.1, 1.0), (2.2, 2.0), (3.4, 1.2) for chromosome \mathbf{m}_1 , + denotes three cluster centers (3.2, 1.4), (1.8, 2.2), (0.5, 0.7) for the chromosome \mathbf{m}_2 , open triangle denotes the chromosome obtained by crossing \mathbf{m}_1 and \mathbf{m}_2 , and open circle denotes the chromosome obtained by crossing \mathbf{m}_1 and \mathbf{m}'_2 with three cluster centers (0.5, 0.7), (1.8, 2.2), (3.2, 1.4)	732
Fig. 9.4	Geometric representation of opposite number of a real number x	788
Fig. 9.5	The opposite point and the quasi-opposite point. Given a solution $\mathbf{x}_i = [x_{i1}, \dots, x_{iD}]^T$ with $x_{ij} \in [a_j, b_j]$ and $M_{ij} = (a_j + b_j)/2$, then x_{ij}^o and x_{ij}^q are the j th elements of the opposite point \mathbf{x}_i^o and the quasi-opposite point \mathbf{x}_i^q of \mathbf{x}_i , respectively. (a) When $x_{ij} > M_{ij}$. (b) When $x_{ij} < M_{ij}$	792
Fig. 9.6	The opposite, quasi-opposite, and quasi-reflected points of a solution (point) \mathbf{x}_i . Given a solution $\mathbf{x}_i = [x_{i1}, \dots, x_{iD}]^T$ with $x_{ij} \in [a_j, b_j]$ and $M_{ij} = (a_j + b_j)/2$, then x_{ij}^o , x_{ij}^q , and x_{ij}^{qr} are the j th elements of the opposite point \mathbf{x}_i^o , the quasi-opposite point \mathbf{x}_i^q , and the quasi-reflected opposite point \mathbf{x}_i^{qr} of \mathbf{x}_i , respectively. (a) When $x_{ij} > M_{ij}$. (b) When $x_{ij} < M_{ij}$	794

List of Tables

Table 1.1	Quadratic forms and positive definiteness of a Hermitian matrix \mathbf{A}	28
Table 2.1	Symbols of real functions	56
Table 2.2	Differential matrices and Jacobian matrices of trace functions	67
Table 2.3	Differentials and Jacobian matrices of determinant functions	69
Table 2.4	Matrix differentials and Jacobian matrices of real functions.....	70
Table 2.5	Differentials and Jacobian matrices of matrix functions	71
Table 2.6	Forms of complex-valued functions	73
Table 2.7	Nonholomorphic and holomorphic functions.....	74
Table 2.8	Complex gradient matrices of trace functions	82
Table 2.9	Complex gradient matrices of determinant functions	83
Table 2.10	Complex matrix differential and complex Jacobian matrix	86
Table 3.1	Extreme-point conditions for the complex variable functions	98
Table 3.2	Proximal operators of several typical functions	127
Table 6.1	Similarity values and clustering results	322
Table 6.2	Distance matrix in Example 6.4	323
Table 6.3	Distance matrix after first clustering in Example 6.4.....	323
Table 6.4	Distance matrix after second clustering in Example 6.4	323
Table 9.1	Relation comparison between objective vectors and approximation sets	711

List of Algorithms

Algorithm 3.1	Gradient descent algorithm and its variants	107
Algorithm 3.2	Nesterov (first) optimal gradient algorithm	116
Algorithm 3.3	Nesterov algorithm with adaptive convexity parameter	117
Algorithm 3.4	Nesterov (third) optimal gradient algorithm	118
Algorithm 3.5	FISTA algorithm with fixed step	132
Algorithm 3.6	Davidon–Fletcher–Powell (DFP) quasi-Newton method	149
Algorithm 3.7	Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton method	149
Algorithm 3.8	Newton algorithm via backtracking line search	150
Algorithm 3.9	Feasible-start Newton algorithm	151
Algorithm 3.10	Infeasible-start Newton algorithm	153
Algorithm 4.1	Conjugate gradient algorithm	164
Algorithm 4.2	Biconjugate gradient algorithm	165
Algorithm 4.3	PCG algorithm with preprocessor	166
Algorithm 4.4	PCG algorithm without preprocessor	167
Algorithm 4.5	TLS algorithm for minimum norm solution	187
Algorithm 4.6	Least angle regressions (LARS) algorithm with Lasso modification	199
Algorithm 5.1	Lanczos algorithm for GEVD	212
Algorithm 5.2	Tangent algorithm for computing the GEVD	213
Algorithm 5.3	GEVD algorithm for singular matrix \mathbf{B}	213
Algorithm 6.1	Stochastic gradient (SG) method	229
Algorithm 6.2	SVRG method for minimizing an empirical risk R_n	232
Algorithm 6.3	SAGA method for minimizing an empirical risk R_n	232
Algorithm 6.4	APPROX coordinate descent method	236
Algorithm 6.5	LogitBoost (two classes)	250
Algorithm 6.6	Adaptive boosting (AdaBoost) algorithm	252
Algorithm 6.7	Gentle AdaBoost	252

Algorithm 6.8	Feature clustering algorithm	266
Algorithm 6.9	Spectral projected gradient (SPG) algorithm for nonlinear joint unsupervised feature selection	268
Algorithm 6.10	Robust PCA via accelerated proximal gradient	278
Algorithm 6.11	Sparse principal component analysis (SPCA) algorithm	281
Algorithm 6.12	Principal component regression algorithm	284
Algorithm 6.13	Nonlinear iterative partial least squares (NIPALS) algorithm	286
Algorithm 6.14	Simple nonlinear iterative partial least squares regression algorithm	289
Algorithm 6.15	GPSR-BB algorithm.....	296
Algorithm 6.16	k -nearest neighbor (k NN)	299
Algorithm 6.17	Alternating projection algorithm for the supervised tensor learning	309
Algorithm 6.18	Tensor learning algorithm for regression	313
Algorithm 6.19	Random K-means algorithm	328
Algorithm 6.20	Normalized spectral clustering algorithm of Ng et al.	331
Algorithm 6.21	Normalized spectral clustering algorithm of Shi and Malik	332
Algorithm 6.22	Constrained spectral clustering for two-way partition	335
Algorithm 6.23	Constrained spectral clustering for K -way partition.....	336
Algorithm 6.24	Nyström method for matrix approximation	336
Algorithm 6.25	Fast spectral clustering algorithm.....	337
Algorithm 6.26	Self-training algorithm	340
Algorithm 6.27	Propagating 1-nearest neighbor clustering algorithm.....	341
Algorithm 6.28	Co-training algorithm	343
Algorithm 6.29	Canonical correlation analysis (CCA) algorithm	348
Algorithm 6.30	Penalized (sparse) canonical component analysis (CCA) algorithm.....	353
Algorithm 6.31	Manifold regularization algorithms.....	368
Algorithm 6.32	Harmonic function algorithm for semi-supervised graph learning	371
Algorithm 6.33	ℓ_1 directed graph construction algorithm.....	378
Algorithm 6.34	Active learning.....	383
Algorithm 6.35	Co-active learning	383
Algorithm 6.36	Active learning for finding opposite pair close to separating hyperplane	384
Algorithm 6.37	Active learning-extreme learning machine (AL-ELM) algorithm	387
Algorithm 6.38	Q-learning.....	396
Algorithm 6.39	Double Q-learning	397
Algorithm 6.40	Weighted double Q-learning	398
Algorithm 6.41	Modified connectionist Q-learning algorithm.....	400
Algorithm 6.42	Deep Q-learning with experience replay	401
Algorithm 6.43	TrAdaBoost algorithm.....	412

Algorithm 6.44	SVD-based alternating structure optimization algorithm	413
Algorithm 6.45	Structural correspondence learning algorithm	414
Algorithm 6.46	EigenCluster: a unified framework for transfer learning	417
Algorithm 6.47	ℓ_p -norm multiple kernel learning algorithm	422
Algorithm 6.48	Heterogeneous feature augmentation	422
Algorithm 6.49	Information-theoretic metric learning	424
Algorithm 7.1	ADAGRAD algorithm for minimizing $\{\langle \mathbf{u}, \mathbf{x} \rangle + \frac{1}{2} \langle \mathbf{x}, \mathbf{H}\mathbf{x} \rangle + \lambda \ \mathbf{x}\ _2\}$	450
Algorithm 7.2	ADAM algorithm for stochastic optimization.....	451
Algorithm 7.3	CD ₁ fast RBM learning algorithm.....	487
Algorithm 7.4	Backpropagation algorithm for the basic autoencoder.....	531
Algorithm 7.5	Extreme learning machine (ELM) algorithm	546
Algorithm 7.6	ELM algorithm for regression and binary classification....	549
Algorithm 7.7	ELM multiclass classification algorithm	551
Algorithm 7.8	DeepWalk(G, w, d, γ, t)	580
Algorithm 7.9	SkipGram($\Phi, \mathcal{W}_{v_i}, w$)	581
Algorithm 7.10	GraphSAGE embedding generation algorithm.....	583
Algorithm 7.11	Batch renormalization, applied to activation \mathbf{y} over a mini-batch.....	595
Algorithm 7.12	Minibatch stochastic gradient descent training of GANs....	601
Algorithm 8.1	PSVM binary classification algorithm.....	650
Algorithm 8.2	SVM-recursive feature elimination (SVM-RFE) algorithm.....	653
Algorithm 8.3	LS-SVM multiclass classification algorithm	658
Algorithm 8.4	PSVM multiclass classification algorithm	660
Algorithm 8.5	Gaussian process regression algorithm	666
Algorithm 9.1	Roulette wheel fitness selection algorithm	700
Algorithm 9.2	Fast-nondominated-sort(P)	701
Algorithm 9.3	Crowding-distance-assignment (I)	703
Algorithm 9.4	Single-objective simulated annealing.....	717
Algorithm 9.5	Multiobjective simulated annealing	721
Algorithm 9.6	Archived multiobjective simulated annealing (AMOSa) algorithm	722
Algorithm 9.7	Genetic algorithm with gene rearrangement (GAGR) clustering algorithm	730
Algorithm 9.8	Nondominated sorting genetic algorithm (NSGA)	738
Algorithm 9.9	(1 + 1) evolutionary algorithm	741
Algorithm 9.10	Multiobjective evolutionary algorithm based on decomposition (MOEA/D).....	748
Algorithm 9.11	MOEA/D-DE algorithm for Solving (9.9.5)	750
Algorithm 9.12	SPEA algorithm	752
Algorithm 9.13	Improved strength Pareto evolutionary algorithm (SPEA2).....	755

Algorithm 9.14 Differential evolution (DE) 764

Algorithm 9.15 Framework of artificial bee colony (ABC) algorithm..... 779

Algorithm 9.16 Particle swarm optimization (PSO) algorithm 783

Algorithm 9.17 Genetic learning PSO (GL-PSO) algorithm..... 785

Algorithm 9.18 Hybrid particle swarm optimization with local search
(HPSO-LS) 787

Algorithm 9.19 Opposition-based differential evolution (ODE)..... 791