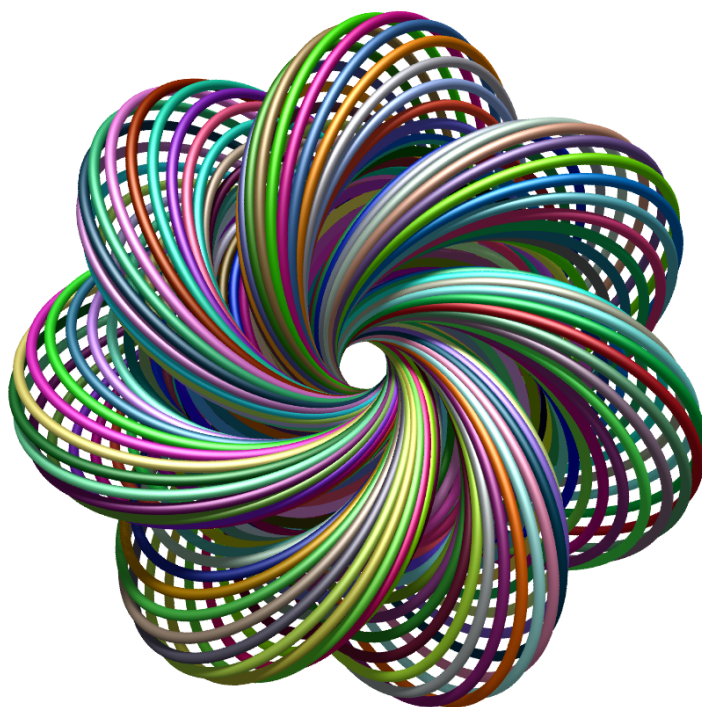


Essentials of Mathematical Methods:

Vol-1 Foundations and Principles
Vol-2 Statistical Learning and Deep Learning

Yuguang Yang

version 4.0



God used beautiful mathematics in creating the world. –Paul Dirac

*Dedicated to
those who appreciate the power of mathematical methods
and enjoy learning it.*

License statement

You are free to redistribute the material in any medium or format under the following terms:

- **Attribution:** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **NonCommercial:** You may not use the material for commercial purposes.

- * The licensor cannot revoke these freedoms as long as you follow the license terms.
- * This license is created via creative commons (<https://creativecommons.org>)
- * If you have any questions regarding the license, please contact the author.

Preface

Objective

Today, mathematical methods, models, and computational algorithms are playing increasingly significant roles in addressing major challenges arising from scientific research and technological developments. Although many novel methods and algorithms, such as deep learning and artificial intelligence, are emerging and reshaping various areas at an unprecedented pace, their core ideas and working mechanisms are inherently related to and deeply rooted in some essential mathematical foundations and principles. By performing an in-depth survey on the underlying foundations, principles, and algorithms, this book aims to navigate the vast landscape of mathematical methods widely used in diverse scientific and engineering domains.

This book starts with a survey of mathematical foundations, including essential concepts and theorems in real analysis, linear algebra, and related fundamentals. Then it examines a broad spectrum of applied mathematical methods, ranging from traditional ones such as optimizations and dynamical system modeling, to state-of-the-art such as machine learning, deep learning, and reinforcement learning. The emphasis is placed on methods for stochastic and dynamical system modeling, optimal decision-making, and statistical learning. For each topic, this book organizes fundamental definitions, theorems, methods, and algorithms in a logical and illuminating way.

Features and Highlights

- Comprehensive, essential, and self-contained.
- Concepts, theorems, and discussions are developed to suit real-world applications.
- Key references and resources are provided on each topic.
- Comparisons and discussions on similar definitions and theorems.
- An evolving book with regular updates on [Github](#).

Acknowledgment

This book evolved from my study notes during my PhD studies at the Johns Hopkins University (JHU). I want to thank the following professors at JHU for their courses and valuable discussion: Daniel Robinson, Teresa Lebar, Andrea Prosperetti, Gregory Chirikjian, Michael Kahadan, James C. Spall, Marin Kobilarov, Suchi Saria, Michael Dimitz, Sean Sun, Ari Turner, Gregory Eyink, Amitabh Basu, John Miller and David Audley. I also want to thank Rachael Zhang for her editorial assistance.

Yuguang Yang, Fall 2019
yangyutu123@gmail.com

Notations

- \mathbb{R} : real numbers.
- \mathbb{R}_+ : nonnegative real numbers.
- \mathbb{R}_{++} : positive real numbers.
- $\bar{\mathbb{R}}$: extended real numbers.
- \mathbb{C} : complex numbers.
- \mathbb{F} : real or complex numbers.
- \mathbb{Q} : rational numbers.
- \mathbb{Z} : integer numbers.
- \mathbb{P} : positive numbers.
- \mathcal{P}_n : polynomial of degree of n .
- \mathbb{N} : natural numbers.
- $\mathcal{R}(A)$: the range of matrix A .
- $\mathcal{N}(A)$: the null space of matrix A .
- V : vector space.
- $\det(A)$: the determinant of matrix A .
- $\text{rank}(A)$: the rank of matrix A .
- $\|\cdot\|_2$: Euclidean 2 norm of a vector of a matrix.
- $\|\cdot\|_F$: Frobenius norm of a matrix.
- $\rho(A)$: the spectral radius of matrix A .
- $\text{Tr}(A)$: the trace of matrix A .
- $L^2[a, b]$: Lebesgue integrable function on $[a, b]$.
- $L^1[a, b]$: Lebesgue integrable function on $[a, b]$.
- $N(0, 1)$: standard Gaussian distribution.
- $N(\mu, \sigma^2)$: Gaussian distribution with mean μ and variance σ^2 .
- $MN(\mu, \Sigma)$: multivariate Gaussian distribution with mean vector μ and covariance matrix Σ .
- $\mathbf{1}(x), I(x)$ indicator function.
- $E[X], \mathbb{E}[X], \mathbb{E}[X]$ expectation of random variable X .
- $\text{Var}[X]$ variance of random variable X .

CONTENTS

i mathematical foundations

- 1 SETS, SEQUENCES, AND SERIES 2
- 2 METRIC SPACE 37
- 3 ADVANCED CALCULUS 57
- 4 LINEAR ALGEBRA I: VECTOR SPACE AND LINEAR MAPS 116
- 5 LINEAR ALGEBRA II: MATRIX ANALYSIS 162
- 6 BASIC FUNCTIONAL ANALYSIS 249

ii mathematical optimization methods

- 7 UNCONSTRAINED NONLINEAR OPTIMIZATION 297
- 8 CONSTRAINED NONLINEAR OPTIMIZATION 340
- 9 LINEAR OPTIMIZATION 386
- 10 CONVEX ANALYSIS AND CONVEX OPTIMIZATION 409

iii classical statistical methods

- 11 PROBABILITY THEORY 458
- 12 STATISTICAL DISTRIBUTIONS 567
- 13 STATISTICAL ESTIMATION THEORY 625
- 14 MULTIVARIATE STATISTICAL METHODS 693
- 15 LINEAR REGRESSION ANALYSIS 737
- 16 MONTE CARLO METHODS 836

iv dynamics modeling methods

- 17 MODELS AND ESTIMATION IN LINEAR SYSTEMS 874
- 18 ESTIMATION IN DYNAMICAL SYSTEMS 939

19	STOCHASTIC PROCESS	953
20	STOCHASTIC CALCULUS	992
21	MARKOV CHAIN AND RANDOM WALK	1042
22	TIME SERIES ANALYSIS	1092

v statistical learning methods

23	SUPERVISED LEARNING PRINCIPLES	1174
24	LINEAR MODELS FOR REGRESSION	1200
25	LINEAR MODELS FOR CLASSIFICATION	1220
26	GENERATIVE MODELS	1281
27	K NEAREST NEIGHBORS	1301
28	TREE METHODS	1308
29	ENSEMBLE AND BOOSTING METHODS	1333
30	UNSUPERVISED STATISTICAL LEARNING	1365
31	PRACTICAL STATISTICAL LEARNING	1431

vi deep learning methods

32	FOUNDATIONS OF DEEP LEARNING	1459
33	NETWORK TRAINING AND OPTIMIZATION	1488
34	CONVOLUTIONAL NEURAL NETWORKS	1512
35	RECURRENT NEURAL NETWORKS	1543

vii optimal control and reinforcement learning

36	CLASSICAL OPTIMAL CONTROL THEORY	1571
37	REINFORCEMENT LEARNING	1590

viii applications

38	NATURAL LANGUAGE PROCESSING I: FOUNDATION	1676
39	NATURAL LANGUAGE PROCESSING II: TASKS	1722
40	DEEP LEARNING FOR AUTOMATIC SPEECH RECOGNITION	1765

41 DEEP LEARNING FOR SPEAKER RECOGNITION 1804

ix appendix

A SUPPLEMENTAL MATHEMATICAL FACTS 1867

Alphabetical Index 1898

LIST OF ALGORITHMS

1	A generic line search algorithm	306
2	Backtracking-Armijo line search algorithm	317
3	Steepest decent Backtracking-Armijo line search algorithm	320
4	Modified Newton Backtracking-Armijo line search algorithm	320
5	Quasi Newton with Wolfe line search algorithm	320
6	A generic trust-region algorithm	322
7	A linear conjugate algorithm	330
8	Iteratively reweighted least squares for p norm least square	333
9	Gauss-Newton method for nonlinear least-square algorithm	335
10	Levenberg-Marquardt method for nonlinear least-square algorithm	336
11	Newton method for root finding	337
12	Primal active-set method for strictly convex quadratic programming	359
13	First order gradient projection algorithm	362
14	The Simplex algorithm (non-degenerate system)	400
15	Primal-dual long-step path-following algorithm	405
16	A generic subgradient algorithm	443
17	Gradient projection algorithm with constant step size	450
18	Gradient projection algorithm with adaptive size	451
19	Proximal gradient algorithm	452
20	Iterative Shrinkage-Thresholding Algorithm with constant step size for L ₁ optimization	454
21	Iterative reweighed estimation for multivariate normal distribution	701
22	EM algorithm for least square with nonconstant variance	801
23	Accept-Reject algorithm for random number generation.	840
24	Importance sampling for Monte Carlo integration.	850

25	MCMC Metropolis-Hasting algorithm	854
26	MCMC Gibbs sampling algorithm	856
27	Recursive linear least square estimation of dynamical systems.	943
28	Recursive nonlinear least square of dynamical systems	945
29	Kalman filtering	949
30	Extended Kalman filter	950
31	Coordinate descent for Lasso regression.	1211
32	Iteratively reweighted least squares for logistic regression	1227
33	Perceptron learning algorithm	1258
34	Soft margin SVM algorithm	1267
35	Multinomial Naive Bayes classification	1287
36	KNN classification and regression algorithm	1302
37	A generic decision tree generation algorithm	1316
38	ID3 classification decision tree algorithm	1321
39	A regression tree growth algorithm	1329
40	A basic bagging algorithm	1338
41	Generic Adaboost classifier algorithm	1344
42	Adaboost regressor algorithm	1348
43	A generic additive model algorithm	1350
44	Generic gradient boosting algorithm	1353
45	Gradient tree boosting algorithm	1355
46	XGBoost algorithm	1361
47	Iterative reweighted least square PCA with outliers algorithm	1378
48	Random sample consensus PCA with outliers algorithm	1378
49	Orthogonal Matching Pursuit	1380
50	K-SVD for dictionary learning.	1382
51	Online dictionary learning	1383
52	Stochastic gradient descent for matrix factorization based recommender systems.	1393
53	Isomap algorithm	1401

54	Kernel PCA algorithm	1402
55	Laplacian Eigenmap algorithm	1407
56	Diffusion map algorithm	1410
57	K-means algorithm.	1415
58	DBSCAN algorithm	1418
59	General spectral clustering algorithm	1420
60	Gaussian mixture model EM algorithm	1424
61	Solve forward backward stochastic differential equation via deep learning. .	1486
62	Full batch gradient descent algorithm	1491
63	Minibatch stochastic gradient descent algorithm	1492
64	Adam stochastic gradient descent algorithm.	1497
65	The policy iteration algorithm for MDP	1599
66	Value iteration algorithm for a finite state MDP	1601
67	First-visit MC value function estimation	1608
68	MC-based reinforcement learning control	1610
69	TD(o) estimation of a value function.	1611
70	SARSA learning	1612
71	Q-learning algorithm	1613
72	TD(n) estimation of a value function.	1616
73	n -step SARSA learning	1617
74	A generic batch policy-gradient algorithm (REINFORCE)	1628
75	A basic Monte-Carlo policy-gradient algorithm	1629
76	Actor-Critic policy-gradient method	1631
77	Policy-gradient method with a value function baseline	1636
78	Neural Fitted Q Iteration (NFQ)	1641
79	Deep Q-learning with experience replay	1643
80	Asynchronous Deep Q-Learning for each thread	1648
81	Deep Q-learning with universal value function approximator	1649
82	Deep deterministic policy gradient algorithm (DDPG)	1652
83	Twin-delayed deep deterministic policy gradient (TD3)	1654

84	Trust Region Policy Optimization	1657
85	Proximal Policy Optimization	1660
86	Soft Actor-Critic (SAC) policy optimization	1663
87	Isotropic multivariate Gaussian evolution strategies for reinforcement learning	1665
88	Isotropic multivariate Gaussian parallelized evolution strategies for reinforcement learning	1665
89	Q learning with prioritized experience replay	1668
90	Deep Q-learning with hindsight experience replay	1669
91	Online greedy MAP decoding for UIS-RNN.	1852

LIST OF FIGURES

Figure 1.1.1	An illustration of set union and intersection.	5
Figure 1.1.2	An illustration of set difference.	5
Figure 1.1.3	Illustrating inclusion exclusion principle via intersections of multiple sets.	7
Figure 1.2.1	Illustration of function mappings. f is not onto Y ; g is both one-to-one and onto Y	9
Figure 1.3.1	Nested intervals on the real line.	15
Figure 1.4.1	Illustration of a convergent sequence.	16
Figure 2.2.1	The set shown in green, not including the dashed boundary, is an open subset of \mathbf{R}^2 , with its usual metric, because it includes an open ball around each of its points.	43
Figure 3.1.1	An illustration of function continuity. (left) A continuous function example. (right) A function with discontinuity at $x = 4, 6, 10$. This function is not a continuous function.	59
Figure 3.1.2	At $x = a$, $\lim_{x \rightarrow a} f(x) = L$. When $\lim_{x \rightarrow a} f(x) = f(a)$, $f(x)$ is continuous at $x = a$ (left); otherwise $f(x)$ is discontinuous at $x = a$ (right).	60
Figure 3.1.4	Illustration of intermediate value theorem. For every y between $f(a)$ and $f(b)$, there exists at least one value c between a, b such that $f(c) = y$.	63
Figure 3.1.5	Extreme values in an example function defined on the interval $[1, 4]$. Here point 1 is global maximum, point 4 is global minimum, point 2 is local minimum, and points 3 and 5 are local maximum values.	64
Figure 3.9.1	An example 3D curve generated by $z = t, x = t \cos(t), y = t \sin(t)$.	109
Figure 3.9.2	An example smooth surface generated by $z = x \exp(-2x^2 - y^2)$	112
Figure 5.3.1	Demonstration of SVD for matrices of different shapes. The dashed lines highlight the compact form SVD.	188
Figure 5.6.1	Illustration of different quadratic forms.	209
Figure 6.2.1	Contraction mapping $f : X \rightarrow X$ shrinks distances between arbitrary two points in a normed space X .	257

Figure 7.1.1	Demonstration of local minimizer (red, green, and blue), strict local minimizer (red and blue), and global minimizer (blue). 300
Figure 7.1.2	A complex objective function in unconstrained optimization. 301
Figure 7.1.3	Illustration of different cases in unconstrained quadratic optimization. 305
Figure 7.2.1	Drawbacks of steepest gradient descent. 308
Figure 7.2.2	Demonstration on the step choices on the iterative algorithm. (a) Large step size. (b) Small step size. 313
Figure 7.2.3	Armijo sufficient decrease condition. 317
Figure 7.3.1	Demonstration of the dogleg path as an approximation to the exact solution path in the trust-region subproblem. 325
Figure 7.4.1	Demonstration of coordinate descent procedures when A is diagonal and non-diagonal. 328
Figure 8.1.1	Demonstration of KKT condition at a local minimal for $f(x_1, x_2) = x_1^2 + x_2^2$ under constraint $x_1 + x_2 = 1$. 344
Figure 8.2.1	Demonstration of KKT condition at a local minimal for $f(x_1, x_2) = x_1^2 + x_2^2$ under constraint $x_1 + x_2 \geq 1$. 354
Figure 9.2.1	The geometry of linear programming. (a) The feasible region is an open space extending to infinity if A is not full column rank. (b-d) Example feasible regions if $\text{rank}(A) = n, m \geq n$. Red arrows are direction of $-c$. When moving along $-c$ in the feasible region, the objective function will decrease. 390
Figure 9.2.2	Demonstration on multiple minimizers forming a convex set. 392
Figure 9.3.1	Overview of geometry approach to linear programming. 394
Figure 10.1.1	Example 2D affine hull and 3D affine hull. 411
Figure 10.1.2	Affine subspace and linear subspace. 413
Figure 10.2.1	(left) A convex set. (right) A non-convex set. 417
Figure 10.2.2	(left) The affine hull of two points in a plane is a line passing through them. (right) The convex hull of two points in a plane is a line segment containing them. 419
Figure 10.2.3	An illustration of separating hyperplane theorem for two convex bodies. 421
Figure 10.2.4	An illustration of Farkas' lemma. (left) When b lies outside the cone (that is, $Ax = b, x \geq 0$ has no solution), there exists a hyperplane, characterized by normal vector y , separating b and the cone. (right) When b lies inside the cone (that is, $Ax = b, x \geq 0$ has a solution), there does not exist a hyperplane, characterized by normal vector y , separating b and the cone. 423

- Figure 10.2.5 An illustration of Farkas' lemma variant where the cone is open set. (left) When b lies outside the cone (that is, $Ax = b, x > 0$ has no solution), there exists a hyperplane, characterized by normal vector y , separating b and the cone. (right) When b lies inside the cone (that is, $Ax = b, x > 0$ has a solution), there does not exist a hyperplane, characterized by normal vector y , separating b and the cone. [424](#)
- Figure 10.3.1 Demonstration of convex functions. (a) A convex function satisfying $f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2)$. (b) A non-convex function where the green points does not satisfy the relation. [425](#)
- Figure 10.3.2 The epigraph (green area) of a convex function. [427](#)
- Figure 10.3.3 Illustration of linear underestimator. [430](#)
- Figure 10.5.1 Demonstration of optimality condition $\nabla f(x^*)^T(x - x^*) \geq 0, \forall x \in X$ when x^* lies on the boundary of X . [440](#)
- Figure 11.1.1 An illustration of measurable functions. [467](#)
- Figure 11.11.1 Visualization of central limit theorem. Samples are drawn from uniform and lognormal distributions. Sample means \bar{X}_n converge to normal distribution in distribution when n is large. [536](#)
- Figure 11.14.1 Entropy $H(q)$ of a Bernoulli distribution as a function of parameter q . [553](#)
- Figure 12.1.1 Comparison of Laplace distribution and normal distribution. [578](#)
- Figure 12.1.2 Density of $LN(0,1)$ and $LN(0,0.5)$. Note the positive skewness. [586](#)
- Figure 12.1.3 Density of $NLN(0,1)$ and $NLN(0,0.5)$. Note the negative skewness. [587](#)
- Figure 12.2.1 Distributions with left-skewness (black) and right-skewness (red). [608](#)
- Figure 12.2.2 Distributions with zero excess Kurtosis (Normal distribution, black), positive excess kurtosis (Laplace distribution, red), and negative excess Kurtosis (Uniform distribution, blue). [609](#)
- Figure 12.2.3 Percentile points at $\alpha = 0.1, 0.2, \dots, 0.9$ for a standard normal distribution. [611](#)
- Figure 12.2.4 QQ plot of different sample distributions against standard normal distribution, including standard normal $N(0,1)$, shifted-scaled normal $N(50,5)$, Student's t with degree 1, and lognormal $LN(0,1)$. Red solid lines are the fitted linear lines. [613](#)
- Figure 13.1.1 Statistical estimation and inference scheme. [628](#)
- Figure 13.1.2 An example of biased estimator with smaller variance than unbiased estimator [633](#)
- Figure 13.1.3 Visualization of log-likelihood function for normal distributed samples. [641](#)

Figure 13.7.1	Demonstration for rejection regions for upper-tailed one-sided hypothesis (a), two-sided hypothesis(b), and lower-tailed one-sided hypothesis (c). 676
Figure 13.8.1	QQ plot with different sample distributions, including normal, Laplace ($b = 4$), Uniform $U([-1, 1])$, Lognormal $LN(0, 1)$. Red solid lines are the fitted linear lines. 685
Figure 14.2.1	Principal components for 2D samples. 704
Figure 14.2.2	PCA eigenface analysis. 712
Figure 14.2.3	PCA eigen-digit analysis for MNIST dataset. 713
Figure 14.2.4	Demonstration of interest rate curve dynamics. 715
Figure 14.2.5	Demonstration of first three dominating PCA factor in the swap rate curve daily change. 715
Figure 14.3.1	2D data x ($N = 1000$ samples) can be generated from a latent linear model. $Z \sim N(0, 1)$, $X Z \sim MN(Wz + \mu, I)$, $\mu = (5, 7)^T$, $W = [1.0, 2.0]^T$. 720
Figure 14.3.2	Correlation structure for the Gaussian copula implied by the factor model. 728
Figure 14.3.3	Factor model using (a) external factors or (b) internal factors. 729
Figure 14.3.4	Scatter plot of AAPL return vs. market excess return, SMB excess return and HML excess return. 733
Figure 15.1.1	Demonstration of simple linear regression model $y = \beta_1 x + \beta_0 + \epsilon$ and multiple linear regression model $y = \beta_1 x_1 + \beta_2 x_2 + \beta_0 + \epsilon$. Scatter points are observed data. The solid line in the left and the plane in the right are the mean responses. 741
Figure 15.3.1	Demo of heteroskedasticity in linear regression. The noises are larger at larger x values. 799
Figure 15.3.2	Demonstrations on linear regression with auto-correlated error. Observations are generated by $y_i = x_i + \epsilon_i$, $\epsilon_i = \rho \epsilon_{i-1} + z$, $z \sim N(0, 1)$. 804
Figure 15.3.3	Illustration of an outlier, a high-leverage point, and a influential point. Left subfigure shows a red-colored outlier, which does not have high leverage and large influence on the regression result. Middle subfigure shows a red-colored high-leverage point, which is not an outlier or influential point due to its weak influence on the regression result. Right subfigure shows an influential point that is both an outlier and a high-leverage point. 809
Figure 15.3.4	Visual scatter and box plots to identify outliers. 810
Figure 15.3.5	Different function choice for M-estimation linear regression 815
Figure 15.3.6	Common linear regression diagnosis plots 818
Figure 15.4.1	Diagnosis plots for a toy linear regression example 821
Figure 15.4.2	Diagnosis plots for the Boston Housing example 823

Figure 16.4.1	Brownian motion interpolation Demo. 862
Figure 19.1.1	An illustration of a random walk mapping a sample point, ω , to a trajectory parameterized by time, where red trajectory sample point HHT, and blue trajectory has sample point THT. 957
Figure 19.3.1	Sample trajectories of Brownian motion process. 962
Figure 19.4.1	Variance of X_t in a Brownian bridge 972
Figure 19.5.1	A typical realized trajectory from the Poisson process with jumps at T_1, T_2 , and T_3 . 977
Figure 20.4.1	The variance function $Var[X(t)]$ for Brownian motion (red) and OU process(black) with $a = 0.5, \sigma = 1$. 1027
Figure 20.4.2	Representative trajectories from three OU processes with different k . k has the unit of inverse year. Mean level $\mu = 50$ and volatility $\sigma = 20$. 1029
Figure 21.1.1	Example Markov chains. Arrows and numbers are transition directions and probabilities. 1045
Figure 21.1.2	Markov chain diagram for a random walk on the state space \mathbb{Z} . 1046
Figure 21.2.1	Demonstration accessibility in a Markov chain. In chain (a), states A and B are accessible to each other or they can communicate. In chain (b), state A can access B but B cannot access A. 1048
Figure 21.2.2	Demonstration of partitioning state space by communicating classes. Green and orange states belong to different communicating classes. Note that a communicating class can consist of only one state. 1049
Figure 21.2.3	Classification of communicating classes into recurrent class and state space by communicating classes. Green states form a communicating class belonging to the transient class. Orange states form a communicating class belonging to the recurrent/closed/adsorbing class. 1056
Figure 21.2.4	Example periodic and aperiodic Markov chains. 1058
Figure 22.1.1	Example time series including a white noise process (upper left), a seasonable time series with periodicity 20, the US new privately owned housing [source], and the US GDP time series[source]. 1096
Figure 22.1.2	Demonstration on using STL to decompose an example CO2 concentration time series. 1101
Figure 22.2.1	Example trajectories of AR(1) models ($X_t = aX_{t-1} + Z_t$) with different choices of a . 1106
Figure 22.2.2	Example trajectories of MA(1) models (upper) and MA(2) models (lower). 1113

Figure 22.2.3	Representative simulated trajectories for AR(1) process with coefficient 1, which forms a unit-root process, and with coefficient -1. 1122
Figure 22.2.4	The ACF and PACF correlogram for a white noise process. 1129
Figure 22.2.5	The ACF and PACF correlogram for an AR(1) process with coefficient 0.8. 1129
Figure 22.2.6	The ACF and PACF correlogram for an MA(1) process. 1130
Figure 22.2.7	The ACF and PACF correlogram for an ARMA(1,1) process. 1130
Figure 22.2.8	Diagnosis plot of residuals for AR(2) model estimation. 1139
Figure 22.2.9	Diagnosis plot of residuals for AR(1) model fitted to time series generated by AR(2) ground truth model. 1140
Figure 22.4.1	Stock index SP500 daily return between 2014 and 2019. 1153
Figure 22.4.2	Simulated representative trajectories from ARCH(1) model with coefficients $\alpha = 0.9$ and $\alpha = 0.5$. 1155
Figure 23.1.1	Scheme of a supervised learning task. Training samples are fed into a learning system to obtain an optimized model, which will be further used in a prediction system for regression and classification tasks. 1176
Figure 23.1.2	Two major types of supervised learning tasks: classification and regression. A classification task can be viewed as drawing a decision boundary in the input feature space. A regression task can be viewed as extrapolating an observed trend in the training data. 1177
Figure 23.1.3	Input feature data type examples. 1178
Figure 23.2.1	A simple regression problem illustrating underfitting and overfitting. Solid lines are models, and points are samples. 1181
Figure 23.2.2	The commonly observed phenomenon of overfitting and underfitting in machine learning. 1182
Figure 23.3.1	Illustrating the impact bias and variance on model performance evaluated on unseen dataset. 1183
Figure 23.3.2	Intuition of bias and variance illustrated in polynomial regression. A linear model suffers from large bias; that is, the predictions averaged over models trained from different training sets cannot approximate the true value well. A high-order polynomial model suffers from large variance; that is, a prediction differs a lot from the expectation of prediction. 1186
Figure 23.3.3	Performance of models with different bias and variance balance. 1187
Figure 23.4.1	Performance of models with different bias and variance balance. 1191

Figure 23.5.1	Regression loss functions: MSE Loss, MAE Loss, Huber loss ($\delta = 0.1, 1, 3$), and Log-Cosh Loss. 1195
Figure 23.5.2	Common classification loss functions. 1197
Figure 24.1.1	Correlation among features and the label MEDV. 1203
Figure 24.1.2	Pair plot among features and the label. 1204
Figure 24.2.1	Comparison different penalization: Lasso $L1$, elastic net, and Ridge $L2$. Orange regions are admissible set for parameter β . Contours are objective function value as a function of parameter β . Black solid circles are the minimizers $\hat{\beta}$ when there are no penalties, and red solid circles are the minimizers when penalties are applied. 1213
Figure 24.2.2	Penalized regression path in a toy regression problem. 1214
Figure 24.3.1	Linear regression with non-linear high order terms. The samples are generated via $y = 2x_1 + x_2 - 0.8x_1x_2 + 0.5x_1^2 + \epsilon$, where ϵ is noise. 1217
Figure 24.3.2	Linear model enhancement flowchart. 1217
Figure 25.1.1	Logistic regression for classification on the Iris data set. 1223
Figure 25.1.2	Loss function value when $y = 1$ and $y = 0$. 1224
Figure 25.1.3	Pair plot analysis results on South Africa heart disease problem. 1232
Figure 25.1.4	$L1$ regularization path for South Africa heart disease problem. 1233
Figure 25.1.5	Logistic regression result with $L1$ penalty. 1234
Figure 25.1.6	Balanced accuracy score vs. inverse regularization strength in credit card fraud detection problem. 1235
Figure 25.1.7	Logistic regression coefficients corresponding to each class. 1236
Figure 25.2.1	Geometry of decision boundary in linear Gaussian discriminant model. 1240
Figure 25.2.2	Comparison of Gaussian LDA and Gaussian QDA on binary classification. Decision boundary of LDA is simply a line in 2D input space and unable to discriminate difficult cases. Gaussian discrimination can have richer decision boundary geometry. LDA using polynomial features is a special case of GDA. 1244
Figure 25.2.3	The decision boundary geometry of LDA and GDA can be understood via their decision functions 1245
Figure 25.3.1	Comparison and PCA and LDA. PCA seeks low dimensional representations that preserves maximum variance; LDA seeks low dimensional presentation representations that maximize class separation. 1246
Figure 25.3.2	The linear discriminant w that maximizes the separability for 2D sample points belonging to two classes. 1247

Figure 25.3.3	Fisher linear discriminate will fail to achieve class separability for complex data structures. 1251
Figure 25.3.4	Comparison of 2D low-dimensional embedding obtained via PCA vs. LDA. 1256
Figure 25.4.1	Scheme of a hyperplane. 1257
Figure 25.4.2	Binary classification using the Perceptron learning algorithm. The hyperplane learned separates the two clusters. 1259
Figure 25.5.1	Left: existence of multiple separating hyperplanes in 2D binary classification problem. Right: hyperplanes with maximum margin. 1261
Figure 25.5.2	SVM classification with different regularization strength. Small C tends to emphasize the margin and ignore the outliers in the training data, while large C may tend to overfit the training data. 1264
Figure 25.5.3	SVM classification using Gaussian kernel. The original problem cannot be separated by linear kernel. 1269
Figure 25.5.4	Comparison of classification loss functions. 1271
Figure 26.2.1	Histogram of the features group by class label (fraud vs. genuine). 1291
Figure 26.2.2	Feature density similarity and predictive performance of model 1292
Figure 27.2.1	Binary classification via KNN algorithm with different choices $K = 1, 3, 5, 7$. Scattered points are training examples classified into two different classes (red and blue). Colored regions are corresponding decision boundaries. 1306
Figure 28.2.1	Demonstration of decision trees. 1315
Figure 28.2.2	Different types of impurity measure for binary classification (p is the probability of the outcome taking label 1). Entropy function, Gini function and classification error function. 1317
Figure 28.2.3	Different splitting strategy when variables taking more than two discrete values. 1319
Figure 28.2.4	The visualization of the decision tree classifier for Iris data set. The tree grows until all examples are classified correctly. The splitting criterion is Gini impurity. 1324
Figure 28.2.5	The visualization of the decision tree classifier for Iris data set. The tree grows until examples in each node is smaller than 10. The splitting criterion is Gini impurity. 1325
Figure 28.2.6	The visualization of the decision tree classifier for Iris data set. The tree grows until examples in each node is smaller than 10. The splitting criterion is entropy and information gain. 1325
Figure 28.3.1	Demonstration of a tree and input space partitioning. 1328

Figure 28.3.2	2D input space partitions cannot be represented by a regression tree. 1328
Figure 28.3.3	Regression tree demonstration in a toy example. 1330
Figure 28.3.4	Variable importance from regression tree in the Boston Housing Pricing problem. 1331
Figure 29.2.1	The correctness probability of a majority vote is greater than the correctness probability of individual votes when individual accuracy probability is greater than 0.5. 1336
Figure 29.4.1	Illustration of adaptive boosting where sample weights are adjusted iteratively based on the classification error. 1343
Figure 29.4.2	Illustration of Adaboost learning process. (top row) Base classifiers trained from weighted samples at different iterations. (bottom row) Final classifier as a weighted sum of base classifiers. 1345
Figure 30.1.1	Demonstration of SVD for matrices of two different shapes. The dashed lines highlight the compact form SVD. 1369
Figure 30.1.2	Principal components for 2D samples. 1373
Figure 30.2.1	Singular value spectrum of LSA on 20-news-group text data. 1388
Figure 30.2.2	SVD for collaborative filtering. 1391
Figure 30.2.3	A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist. [9] 1393
Figure 30.3.1	Triangle and Tetrahedron reconstructed from distance matrix by MDS method. 1399
Figure 30.3.2	Application of MDS, based on Euclidean distance, to Swiss Roll data set cannot fully reveal of the global structure. 1400
Figure 30.3.3	Isomap analysis of MNIST dataset. 1411
Figure 30.3.4	Isomap analysis of digit '5' in MNIST dataset 1412
Figure 30.4.1	Demonstration of k-means clustering on a data set with two blobs. 1414
Figure 30.4.2	K-means performance can be affected by a number of factors, including incorrect number of clusters/blobs, non-spherical clusters/blobs, clusters/blobs with unequal variance, and bad initial cluster centers. 1416
Figure 30.4.3	Clustering comparison between Kmeans and Kmeans++. 1417
Figure 30.4.4	DBSCAN demo with different ϵ . 1419
Figure 30.4.5	Spectral clustering demo. 1421
Figure 30.4.6	Clustering comparison between K-means and GMM. 1425
Figure 30.4.7	K-means application to image segmentation. 1427
Figure 31.1.1	Scheme for ROC curves diagram. A and B demote ROC curves of different model. 1437
Figure 31.2.1	Scheme for cross-validation error calculation procedure. 1441

Figure 31.2.2	Hyperparameter search via turning regularization parameter λ . At large λ , heavy regularization causes underfitting; at small λ , insufficient regularization causes overfitting. 1443
Figure 31.3.1	Left: A sample of nine real-world time series reveals a diverse range of temporal patterns [4, 5]. Right: Examples of different classes of methods for quantifying the different types of structure, such as those seen in time series on the left: (i) distribution (the distribution of values in the time series, regardless of their sequential ordering); (ii) autocorrelation properties (how values of a time series are correlated to themselves through time); (iii) stationarity (how statistical properties change across a recording); (iv) entropy (measures of complexity or predictability of the time series quantified using information theory); and (v) nonlinear time-series analysis (methods that quantify nonlinear properties of the dynamics).[7] 1454
Figure 32.1.1	Scheme of an artificial neuron. 1462
Figure 32.1.2	Common activation functions in artificial neural networks. 1463
Figure 32.1.3	Scheme for an artificial neural network. 1464
Figure 32.1.4	A four-layer feed-forward neural network. 1465
Figure 32.1.5	A three-layer feed-forward neural network with three output units. 1465
Figure 32.1.6	A four-layer feed-forward neural network. 1468
Figure 32.2.1	Simple neural network architecture for linear regression and classification application. 1473
Figure 32.2.2	Visualization of first layer weight for a one-layer linear multi-class classification neural network 1476
Figure 32.2.3	Polynomial regression with degree $d = 1, 3, 6, 10$. 1477
Figure 32.2.4	Multi-layer feed-forward neural network for nonlinear regression. 1478
Figure 32.2.5	Example images from the Fashion MNIST dataset. 1480
Figure 32.2.6	The confusion matrix from fashion MNIST classification results. 1481
Figure 32.2.7	Classification results for a set of randomly selected samples. 1482
Figure 32.2.8	Feed-forward neural network for rating matrix decomposition in a recommender system. 1483
Figure 33.1.1	An example saddle point at $(0, 0, 0)$, which locally minimizes the x direction but maximizes the y direction.. 1491
Figure 33.1.2	SGD without momentum and with momentum. SGD with momentum can accumulate gradient/velocity in horizontal direction and move faster towards the minimum located at the center. 1495
Figure 33.2.1	Impact batch normalization on internal data distribution and learning process. Code 1502

Figure 33.2.2	Dropout technique for a simple feed-forward neural networks. The original network (left) and the neural network after some neurons being dropped out (right). 1506
Figure 33.2.3	Impact of dropout on model generalization. Code 1507
Figure 34.1.1	Comparison of receptive fields in fully-connected layer and local-connected layer in CNN. Credit 1513
Figure 34.1.2	Demo for one kernel 'convoluting' with an input image. The values (across all channels) in the dashed box will convolve with the kernel to produce a value in the output. 1514
Figure 34.1.3	Max-Pooling layer illustration. Depending on the kernel size and stride, we will have different output shape. 1516
Figure 34.1.4	Gradient distribution during back-propagation for mean pooling and max pooling (both have kernel size 2). 1517
Figure 34.1.5	A typical CNN architecture for classification or regression tasks. 1518
Figure 34.1.6	1×1 convolution (stride 1 and zero padding) can be used to increase channel size (upper) or decrease channel size (lower) without changing the spatial size. 1519
Figure 34.1.7	Illustration of regular 3×3 convolution and dilated convolution. (Upper) Regular convolution act on contiguous patches in the input and use pooling to increase receptive field for layers in the back. (Lower) Dilated convolution insert <i>holes</i> in the convolution computation and expand the receptive field without pooling. 1520
Figure 34.2.1	Scheme and architecture for LeNet[2]. 1521
Figure 34.2.2	Architecture of AlexNet. 1522
Figure 34.2.3	A typical VGG architecture: VGG-19 scheme. 1524
Figure 34.2.4	Training error (left) and testing error (right) on CIFAR-10 with 20-layer and 56-layer vanilla CNN networks. The deeper network has both higher training error and testing error.[5] 1525
Figure 34.2.5	Scheme for a residual block without (left) and with (right) 1×1 convolution. 1×1 convolution is used to match the channel number before and after the convolutional layers. 1526
Figure 34.2.6	Scheme of a 34-layer ResNet 1527
Figure 34.2.7	Scheme for a residual bottleneck block. 1528
Figure 34.3.1	Example images from CIFAR10 image dataset. 1529
Figure 34.4.1	Visualization of convolution layer. 1531
Figure 34.4.2	Grad-CAM method applied to understand image classification. Middle and right are class-discriminative localization map superimposed onto the original image. 1532

- Figure 34.5.1 A CNN based autoencoder. An autoencoder consists of an encoder that transforms high-dimensional image into a low-dimensional code and a decoder that unfolds the code to reconstruct the image. [1534](#)
- Figure 34.5.2 Comparison of reconstruction performance on random samples from MNIST data set. Top row is the original data. Middle row is autoencoder result based on a 49 dimensional code. Bottom row is PCA result based on a 50 dimensional code. [1534](#)
- Figure 34.5.3 Denoising autoencoders applied to remove the noise in the MNIST dataset. [1535](#)
- Figure 34.6.1 Demonstration of neural style transfer with Van Gogh painting style. [1537](#)
- Figure 34.6.2 Demonstration of neural style transfer with Picasso painting style. [1538](#)
- Figure 34.7.1 Application of CNN in deep reinforcement learning in Q learning approach and Actor-Critic approach. Two streams of sensory inputs are fed to the neural network, including a pixel image of the robot's neighborhood fed into a convolutional layer and the target's position fed into a fully connected layer. [1540](#)
- Figure 35.1.1 Scheme of recurrent units in a neural network (left). Recurrent neural network can be unrolled (right) [1545](#)
- Figure 35.1.2 Scheme for backpropagation through time in a simple RNN. Red arrows are backpropagation directions. [1546](#)
- Figure 35.2.1 Scheme of an LSTM cell. [1549](#)
- Figure 35.2.2 Scheme of a GRU cell. Modified from [2, p. 152]. [1552](#)
- Figure 35.3.1 Different types of units in RNNs: (a) Vanilla RNN cell. (b) LSTM cell. (c) GRU cell. [Credit 1553](#)
- Figure 35.3.2 Stacked RNN and bidirectional RNN. [1554](#)
- Figure 35.3.3 Stacked bidirectional RNN [1555](#)
- Figure 35.3.4 Typical RNN connection to the output layer. [1556](#)
- Figure 35.3.6 LSTM cell with a projection layer (purple box) added after the hidden output h_t . [1557](#)
- Figure 35.3.5 Three places to add Dropout to a RNN: at the input, at the recurrent connection, and at the output. [1557](#)
- Figure 35.4.1 RNN architecture for time series prediction . (left) In the training phase, RNN are updated by minimizing the next step prediction error. (right) In the prediction phase, trained RNN is used to sequentially predict next step state value based on preceding predicted state value. [1560](#)
- Figure 35.4.2 RNN one-step forward and multiple forward prediction performance for Sine time series. [1561](#)

- Figure 35.4.3 RNN architecture for time series prediction with covariates. (left) In the training phase, RNN are updated by minimizing the next step prediction error. (right) In the prediction phase, trained RNN is used to sequentially predict next step state value based on preceding predicted state value. Note that covariate time series are assumed available for all time steps. [1562](#)
- Figure 35.4.4 DeepAR model architecture for time series prediction. Outputs are the parameters characterizing the condition distribution of x_{t+1} conditioned on histories of x_t and z_t . (a) In the training phase, RNN are updated by minimizing the negative log likelihood function. (b) In the prediction phase, a predicted \hat{x}_{t+1} are sampled from predicted conditional distribution and then used to predict next step conditional distribution. [1563](#)
- Figure 35.4.5 A RNN architecture for MNIST image recognition. [1565](#)
- Figure 35.4.6 A RNN for character-level word classification. [1566](#)
- Figure 35.4.7 A RNN for character-level language model. (left) During the training session, one-hot coded characters are directly fed into RNN and predict the next character in the word. (right) A RNN for character-level language model. During the word generation session, the network starts with a user input character and continues the generation process with predicted character from the previous step. [1567](#)
- Figure 37.1.1 Policy iteration involves iteratively carrying out policy evaluation and policy improvement procedures. [1595](#)
- Figure 37.2.1 One core component in reinforcement learning is agent environment interaction. The agent takes actions based on observations on the environment and a decision-making module that maps observations to action. The environment model updates system state and provides rewards according to the action [1603](#)
- Figure 37.2.2 Scheme of the Atari game *breakout*. [1604](#)
- Figure 37.2.3 Policy evaluation and policy improvement framework in the context reinforcement learning. [1605](#)
- Figure 37.3.1 Neural network parameterization for a Gaussian policy (a) and (b) a generic stochastic policy. [1633](#)
- Figure 37.4.1 A typical feed-forward neural network used in NFQ to approximate the Q function. [1640](#)
- Figure 37.4.2 The network architecture for canonical deep Q learning. The network takes a state or an observation, denoted by s , as the input, and outputs multiple values corresponding $Q(s, a)$. The number of outputs [1642](#)

- Figure 37.4.3 A typical single stream Q-network (top) in deep Q learning and a dueling Q-network (bottom). The dueling network has two streams to separately estimate (scalar) state-value and the advantage function for each action; the green output module synthesizes the Q value from two streams. Both networks output Q-values for each action. [10]. 1645
- Figure 37.4.4 In a DQRN, recurrent layers are usually placed on the last layer before output. Unlike canonical DQN, we need to feed sequence of observation into the network and finally output Q values. Above scheme only unfolds to two steps.[11] 1646
- Figure 37.4.5 An example architecture that implements the asynchronous reinforcement learning paradigm. Multiple agents interact with multiple instances of environments in parallel. Agents collect experiences to train a globally shared network that learns control policies. 1647
- Figure 37.4.6 A comparison between a typical deep Q learning network (left) and a typical universal value function approximator network (right). 1649
- Figure 37.4.7 A typical deep neural network architecture for DDPG reinforcement learning. The actor network outputs control policy; the critic network outputs estimate of Q function. Through back-propagation, the critic network improves estimation accuracy and the actor network improves policy. 1651
- Figure 37.5.1 Illustration of planning curriculum on a swiss roll manifold. Red targets can be generated along the path connecting from an intended start point to the target goal position. 1670
- Figure 37.5.2 One representative trajectory on the curved surface via a control policy learned via curriculum learning on a low-dimensional manifold. 1671
- Figure 38.1.1 Modern NLP tasks and their application areas. Source from [1]. 1679
- Figure 38.2.1 (a) Embedding layer maps large, sparse one-hot vectors to short, dense vectors. (b) Example of low dimensional embeddings that capture semantic meanings. 1683
- Figure 38.2.2 (left) Example of co-occurrence matrix constructed from corpus "I love math" and "I like NLP". The context window size of 2. (right) We can obtain lower-dimensional word embeddings from SVD truncated factorization of the co-occurrence matrix. Such low-dimensional embeddings capture important semantic and syntactic information in the co-occurrence matrix. 1685

Figure 38.2.3	(a) The CBOW architecture that predicts the central word given its surrounding context words. (b) The Skip-gram architecture that predicts surrounding words given the central word. The embedding layer is represented by a $V \times D$ weight matrix that performs look-up for each word token integer index, where V is the vocabulary size and D is the dimensionality of the dense vector. The linear output layer is also represented by a $V \times D$ weight matrix that is used to compute the logit for each token label as sort of classification over the vocabulary. 1687
Figure 38.2.4	Visualization of neighboring words of <i>apple</i> in a 2D low-dimensional space (first two components via PCA). Image from Tensorflow projector . 1691
Figure 38.3.1	Illustrating basic language modeling tasks: assigning probability to a sentence (left); and predict the next word based on preceding context (right). 1695
Figure 38.3.2	Feedforward neural network based language model. 1700
Figure 38.3.3	Recurrent neural network based language model. 1701
Figure 38.4.1	Static word embedding approach vs. contextualized word embedding approach. 1704
Figure 38.4.2	A generic neural contextual embedding encoder. 1704
Figure 38.4.3	BERT model architecture 1705
Figure 38.4.4	Input embedding in BERT, which consists of token embedding, segment embedding and positional embedding. 1706
Figure 38.4.6	The multi-head attention architecture 1708
Figure 38.4.5	Example position encodings of dimensionality 256 for position index from 1 to 256. 1708
Figure 38.4.7	In ELMO, static word embeddings (e_1, \dots, e_N) are contextualized by stacked bidirectional LSTM as (h_1, \dots, h_N) . 1711
Figure 38.4.8	BERT pre-training and downstream task fine tuning framework. Image from [17] . 1712
Figure 38.4.9	BERT architecture configuration for different downstream tasks. 1716
Figure 39.1.1	A typical workflow for text classification tasks. 1724
Figure 39.1.2	Fasttext classification architecture. 1730
Figure 39.1.3	Demonstration of a CNN filter applying to a sentence to produce a one-dimensional feature vector. 1731
Figure 39.1.4	2D CNN for sentence classification. 1732
Figure 39.1.5	1D CNN for sentence classification. 1733
Figure 39.1.6	Character level CNN for text classification tasks. 1734
Figure 39.1.7	RNN architecture for textual classification. 1735
Figure 39.2.1	A basic window classification model for the NER task. 1737

- Figure 39.2.2 Linear conditional random field architectures. (upper) Each output label only connects with the input token at the same step and its immediate left and right label. (lower) Each output label connects with all input tokens. [1742](#)
- Figure 39.2.3 BiLSTM for sequence labeling tasks. [1745](#)
- Figure 39.2.4 BiLSTM with an additional linear CRF for sequence labeling tasks. [1746](#)
- Figure 39.2.5 The BERT model architecture for sequence labeling has a token classification head on top (a linear layer on top of the hidden-states output). [1747](#)
- Figure 39.3.1 Siamese type of training scheme to learning sentence embedding from natural language inference task. The sentence encoder is bi-directional LSTM architecture with max pooling and it takes 300 dimension of Glove word embeddings of constituents as the input. [1750](#)
- Figure 39.3.2 **(left)** Sentence-BERT architecture with classification objective function for training. The two BERT networks are pre-trained have tied weights (Siamese network structure). **(right)** Sentence-BERT architecture at inference. The similarity between two input sentences can be computed as the similarity score between two sentence embeddings. [1751](#)
- Figure 39.4.1 Seq2seq modeling for language transformation. The input and output sequences might have different lengths, and not in synchrony. [1753](#)
- Figure 39.4.2 The Encoder-decoder architecture for seq2seq language modeling. The input sequence is fed into the encoder RNN and terminated by an explicit <EOS> symbol. Then the decoder RNN starts with the context vector and the final prediction of the encoder to generate an output sequence until an explicit <EOS> symbol is produced. [1754](#)
- Figure 39.4.3 The Encoder-decoder architecture with attention mechanism for seq2seq modeling. During the encoding phase, all hidden states, rather than the final one, are saved to construct different context vectors via linear combination for the decoding stage. During the decoding phase, relevant context vectors are constructed and fed into the each hidden states in the decoder. [1757](#)
- Figure 39.4.4 A bidirectional RNN encoder system with attention mechanism. [\[11\]](#) [1758](#)

- Figure 39.4.5 The model architecture of Google’s Neural Machine Translation system, which consists of an encoder module (left), an attention module (middle), a decoder module (right). The bottom encoder layer is bi-directional and other layers are uni-directional. Residual connections are used from the third layer in the encoder and in all layers in the decoder. There are total 8 LSTM layers in both encoder and decoder and the model is partitioned into multiple GPUs to speed up training. Image from [13]. 1759
- Figure 39.4.6 The residual connection and bi-directional LSTM structure in the model. Image from [13]. 1760
- Figure 40.1.1 (a) 100 second audio signal (sampling rate 16kHz), with a sampling rate 16kHz and total samples of 950353. (b) The magnified view of the audio signal in (a) within 1 s. 1768
- Figure 40.1.2 Speech audio signals are decomposed into overlapping frames, which are further processed into features for machine learning applications. 1769
- Figure 40.1.3 (a) Hamming window function. (b) A signal frame before and after hamming windowing. 1770
- Figure 40.1.4 Image from source. 1772
- Figure 40.1.5 Demonstration of DFT/FFT on the windowed frame in Figure 40.1.3. This example frame has 400 data points. For FFT, we zero padded the frame to have 512 sample points. The frequency k runs to 200 and 256 in DFT and FFT, respectively. 1774
- Figure 40.1.6 FFT magnitudes corresponding to 256 frequencies for each frame (horizontal axis, 500 frames in total) in the signal. 1776
- Figure 40.1.7 Mel scale transformation for physically measure frequency. 1776
- Figure 40.1.8 Computation of filter bank features. 1777
- Figure 40.1.9 Mel spectrum at 16k sampling rate and 8k sampling rate. 1778
- Figure 40.1.10 The computational flow for MFCC features. 1779
- Figure 40.1.11 An example MFCC feature. 1780
- Figure 40.2.1 A traditional end-to-end speech recognition system 1784
- Figure 40.2.2 A monophone acoustic generation is modeled by three phases, and each phase is an HMM state. 1787
- Figure 40.3.1 Demonstration of SpecAugment technique. Upper is the original spectrogram, middle and bottom are two augmented spectrogram with frequency masking (rows) and time masking (columns) 1790
- Figure 40.4.1 Listen, Attend and Spell (LAS) architecture. The listener is a pyramidal bidirectional LSTM encoding input sequence (x_1, \dots, x_T) into high level features $h = (h_1, \dots, h_U)$. The speller is an attention-based decoder generating the y characters from h . Image from [8]. 1794

- Figure 40.4.2 Alignment, as indicated by attention, between character outputs and audio signal produced by the Listen, Attend and Spell (LAS) model for the utterance “how much would a woodchuck chuck”. Image from [8]. 1795
- Figure 40.4.3 (left)Handwriting recognition: The input can be (x,y)(x,y) coordinates of a pen stroke or pixels in an image. (right) Speech recognition: The input can be a spectrogram or some other frequency based feature extractor. Image from [9]. 1796
- Figure 40.4.4 Typical acoustic encoder with CTC loss. 1797
- Figure 40.4.5 Illustrating valid alignments and the summation of path probability. 1798
- Figure 40.4.6 Multilayer RNN as the feature extractor of Deep Speech architecture.[11] 1800
- Figure 40.4.7 A typical transducer architecture. 1801
- Figure 41.1.1 Speaker recognition system applications. 1808
- Figure 41.1.2 Audio features of utterances from a male and a female speaking *Hey, Alexa*. Code 1809
- Figure 41.1.3 A generic computational flow for speaker verification 1810
- Figure 41.1.4 Computation of filter bank features. 1813
- Figure 41.2.1 GMM-UBM scheme for speaker identification. 1817
- Figure 41.2.2 i-vector computational flow. 1820
- Figure 41.3.1 Two phases (enrollment and verification) of using deep learning methods for speaker verification. 1822
- Figure 41.3.2 Mel-spectrums or filter bank energies of a speech is fed into the neural network as the embedding extractor. The embedding is refined by minimizing either classification loss or metric learning loss, which corresponding to two different paradigms. The speaker embedding extracted are named d-vector by Google or x vector by Hopkins 1823
- Figure 41.3.3 Computation in TDNN with sub-sampling (red) and without sub-sampling (blue+red) 1824
- Figure 41.3.4 Pairwise contrastive loss encourages the learning of representations that positive pairs are close and negative pairs are far apart. 1829
- Figure 41.3.5 Prototypical loss (left) and angular prototypical loss. Utterances in the support set are used to compute class centroids. Losses are measured between class centroids and query examples. 1830
- Figure 41.3.6 Decision boundaries in a binary classification problem using Softmax. For simplicity, we assume bias terms are zero. Decision boundaries are perpendicular to $w_1 - w_2$. When x falls into orange region, it belongs to class 1; when x falls into green region, it belongs to class 2. 1833

- Figure 41.3.7 Visualization of 2D decision boundary resulted from Softmax loss and normalized Softmax loss for binary classification. When x falls into orange region, it belongs to class 1; when x falls into green region, it belongs to class 2. [1835](#)
- Figure 41.3.8 Visualization of 2D decision boundary resulted from normalized Softmax loss and AM-Softmax loss for binary classification. When x falls into orange region, it belongs to class 1; when x falls into green region, it belongs to class 2. [1837](#)
- Figure 41.4.1 A speaker diarization system aims to partition an audio into homogeneous segments corresponding to different speakers. [1839](#)
- Figure 41.4.2 Schematics for calculating diarization error. [1839](#)
- Figure 41.4.3 Conventional clustering approach pipeline for speaker diarization. [1841](#)
- Figure 41.4.4 Supervised clustering method can help reduce ambiguity in unsupervised clustering. The blobs in left figure can be one cluster or two clusters, depending on the algorithm parameter and model assumptions. In a supervised learning setting, training data examples (middle) provide additional clues on how to cluster data when ambiguity occurs (right). [1842](#)
- Figure 41.4.5 A generic end-to-end neural diarization pipeline for speaker diarization. [1843](#)
- Figure 41.4.6 Bottom up scheme of agglomerative hierarchical clustering [1845](#)
- Figure 41.4.7 A LSTM speaker diarization system. Image from [29]. [1846](#)
- Figure 41.4.8 Training examples consisting of input feature sequence and the ground truth label sequence. [1847](#)
- Figure 41.4.9 Generative process of UIS-RNN. Colors indicate labels for speaker segments. There are four options for y_7 given $x_{[6]}, y_{[6]}$. Image from [34]. [1851](#)
- Figure 41.4.10 A general end-to-end neural speaker diarization system with permutation invariant training loss. [1854](#)
- Figure 41.4.11 Two-speaker end-to-end neural speaker diarization (EEND) model trained with the permutation invariant loss and the deep clustering loss. Image from [30]. [1856](#)
- Figure 41.4.12 Two-speaker self-attentive end-to-end neural speaker diarization model trained with permutation-free loss. Image from [31]. [1857](#)
- Figure 41.4.13 Multihead self-attention calculation scheme. [1858](#)

LIST OF TABLES

Table 13.8.1	Test on mean with known variance σ^2	684
Table 13.8.2	Test on mean with unknown variance σ^2	686
Table 13.8.3	Test on variance	686
Table 13.8.4	Test on variance comparison between two samples	687
Table 14.2.1	Eigenvectors and eigenvalues for swap rate daily change	716
Table 14.3.1	statistics on Fama-French 3 factors from July 1963 to Dec. 1991.	732
Table 14.3.2	AAPL stock return modeled by the Fama-French 3 factor model.	734
Table 21.2.1	Summary of Markov chain state property[3, p. 140]	1060
Table 22.2.1	Summary of PACF and ACF for AR, MA, and ARMA processes.	1129
Table 23.1.1	Common application examples of supervised learning.	1177
Table 24.1.1	Linear regression results.	1205
Table 25.1.1	Logistic regression results on South Africa heart disease problem.	1232
Table 30.2.1	Most frequent words in the top 8 topics	1389
Table 37.2.1	Estimating cumulative rewards $G_t^{(n)}$ of different steps n as the target for value function V . $G^{(1)}$ corresponds to temporal-difference TD(0) and $G^{(\infty)}$ corresponds to Monte-Carlo estimation. If the process terminates at K and $K < n$, then we use $G_t^{(n)} = G_t^{(K)}$. Trajectories are generated under policy π .	1615
Table 38.4.1	Comparison of model parameter size for different configurations of BERT and ALBERT models.	1718
Table 39.2.1	Common used entity category in NER tasks.	1737
Table 39.2.2	Universal Dependencies part-of-speech tags	1739
Table 39.3.1	<i>Entailment</i> , <i>contradiction</i> , and <i>neural</i> examples in a natural language inference task.	1749
Table 40.2.1	The CMU Pronouncing Dictionary	1786
Table A.9.1	Closed Newton-Cotes Formula	1888

CONTENTS

i mathematical foundations

1	SETS, SEQUENCES, AND SERIES	2
1.1	Sets	4
1.1.1	Basic concepts and definitions	4
1.1.2	Basic properties of sets	6
1.1.3	Set disjoint and partition	8
1.2	Functions	9
1.2.1	Basic concepts	9
1.3	Real numbers	11
1.3.1	Rational and irrational numbers	11
1.3.2	Dense subset	11
1.3.3	Axiom of completeness	12
1.3.4	Nested interval property	15
1.4	Sequences in \mathbb{R}	16
1.4.1	Basics	16
1.4.2	Algebraic properties of limits	17
1.4.3	Sequence characterization of a dense subset	19
1.5	Monotone sequence	21
1.5.1	Fundamentals	21
1.5.2	Applications	21
1.6	Subsequences and limits	24
1.6.1	Subsequence	24
1.6.2	Bolzano-Weierstrass theorem	25
1.7	Cauchy sequence	26
1.8	Infinite series	28
1.8.1	Series and convergence	28
1.8.2	Absolute convergence	29
1.8.3	Tests for convergence	30
1.8.4	Inequalities and l_2 series	32
1.8.4.1	Holder's and Minkowski's inequality	32
1.8.4.2	Cauchy-Schwarz inequality	33
1.9	Notes on bibliography	35
2	METRIC SPACE	37
2.1	Metric space	38

2.1.1	Definitions	38
2.1.2	Examples	38
2.2	Open sets & closed sets in metric space	43
2.3	Open and closed sets in \mathbb{R}^n	46
2.3.1	Sequences in \mathbb{R}^n	46
2.3.2	Open and closed sets	47
2.4	Compact sets	49
2.4.1	Compact sets in a metric space	49
2.4.2	Compact sets in \mathbb{R}^N	51
2.5	Completeness of metric space	52
2.5.1	Sequence and completeness	52
2.5.2	Completeness of \mathbb{R}^n	53
2.6	Notes on bibliography	55
3	ADVANCED CALCULUS	57
3.1	Continuous functions on \mathbb{R}	59
3.1.1	Definitions	59
3.1.2	Basic properties of function continuity	61
3.1.3	Boundness of continuous functions	62
3.1.4	Intermediate value theorem	62
3.1.5	Extreme value theorem	64
3.2	Uniform continuity	66
3.2.1	Uniform continuity on real line	66
3.2.1.1	Concepts	66
3.2.1.2	Implications of uniform continuity	68
3.2.1.3	Lipschitz continuity	69
3.2.2	Locally and globally Lipschitz continuous	69
3.3	Continuous function on metric space	71
3.3.1	Boundness and extreme values	72
3.3.2	Uniform continuity on metric space	73
3.4	Differentiation	75
3.4.1	Differential function concept	75
3.4.2	Lipschitz continuity and differentiability	76
3.4.3	Differential rules	77
3.4.4	Mean value theorem	79
3.5	Function sequence and series	83
3.5.1	Pointwise convergence, uniform convergence	83
3.5.2	Properties of uniform convergence	84
3.5.2.1	Uniform convergence preserve continuity	84
3.5.2.2	Exchange limits and integration	85
3.5.2.3	Exchange limits and differential	85
3.5.2.4	Linearity of uniform convergence	85

3.6	Power series	87
3.6.1	Fundamentals	87
3.6.2	Term-by-term operation	89
3.6.3	Power series and analytic function	89
3.6.4	Approximation by polynomials	90
3.7	Taylor polynomial and Taylor series	92
3.7.1	Taylor polynomial and approximation	92
3.7.2	Taylor series and Taylor's theorem	94
3.7.3	Common Taylor series	96
3.7.4	Useful approximations	98
3.8	Riemann Integral	100
3.8.1	First Fundamental Theorem of Calculus	100
3.8.2	Second Fundamental Theorem of Calculus	100
3.8.2.1	Fundamentals	100
3.8.2.2	Differentiating definite integrals	102
3.8.2.3	Application to differential equation	104
3.8.3	Essential theorems	104
3.8.4	Integration rules	106
3.8.5	Improper Riemann integrals	106
3.9	Curves and surfaces	109
3.9.1	Curvature	111
3.9.2	Surfaces	112
3.10	Notes on bibliography	114
4	LINEAR ALGEBRA I: VECTOR SPACE AND LINEAR MAPS	116
4.1	Vector space theory	119
4.1.1	Vector space	119
4.1.2	Subspace	120
4.1.3	Sum and direct sum	121
4.1.4	Basis and dimensions	123
4.1.5	Complex vector space vs. real vector space	125
4.2	Linear maps & linear operators	127
4.2.1	Basic concepts of linear maps	127
4.2.2	Fundamental theorem of linear maps	128
4.2.3	Isomorphism	130
4.2.4	Coordinate map properties	132
4.2.5	Change of basis and similarity	133
4.2.5.1	Change of basis for coordinate vector	133
4.2.5.2	Change of basis for linear maps	133
4.2.6	Linear maps and matrices	133
4.2.6.1	Similarity	135
4.3	Fundamental theorems of ranks and linear algebra	136

4.3.1	Basics of ranks	136
4.3.2	Fundamental theorem of ranks	137
4.3.3	Fundamental theorem of linear algebra	138
4.4	Complementary subspaces and projections	140
4.4.1	General complementary subspaces	140
4.4.2	Orthogonal complementary spaces and projections	142
4.4.3	Decomposition of orthogonal projectors	146
4.5	Orthonormal basis and projections	149
4.5.1	Gram-Schmidt Procedure	149
4.5.2	Orthogonal-triangular decomposition	149
4.6	Application to systems of linear equations	151
4.6.1	Overview	151
4.6.2	Homogeneous systems	151
4.6.3	Non-homogeneous systems	152
4.6.4	Overdetermined vs. underdetermined systems	153
4.6.5	Solution methods	154
4.6.6	Error bounds in numerical solutions	158
4.6.6.1	Condition number	158
4.6.6.2	Error bounds	159
4.7	Notes on bibliography	160
5	LINEAR ALGEBRA II: MATRIX ANALYSIS	162
5.1	Eigenvectors and eigenvalues of Matrices: general theory	165
5.1.1	Existence and properties of eigenvalues	165
5.1.2	Properties of eigenvectors	167
5.1.3	Right and left eigenvectors	168
5.1.4	Diagonalizable matrices	169
5.2	Eigenvalue and eigenvectors of matrices: case studies	172
5.2.1	Real diagonalizable matrix	172
5.2.2	Real symmetric matrix	173
5.2.2.1	Spectral properties	173
5.2.2.2	Rayleigh quotients	175
5.2.2.3	Pointcare inequality	178
5.2.3	Hermitian matrix	179
5.2.4	Matrix congruence	181
5.2.5	Complex symmetric matrix	182
5.2.6	Unitary, orthonormal & rotation matrix	182
5.2.7	Generalized eigenvalue problem	183
5.3	Singular Value Decomposition theory	186
5.3.1	SVD fundamentals	186
5.3.2	SVD and matrix norm	188
5.3.3	SVD vs. eigendecomposition	189

5.3.4	SVD low rank approximation	190
5.3.4.1	Frobenius norm low rank approximation	190
5.3.4.2	Two-norm low rank approximation	192
5.4	Generalized eigenvectors and Jordan normal forms	194
5.4.1	Generalized eigenvectors	194
5.4.2	Upper triangle matrix and nilpotent matrix	197
5.4.3	Jordan normal forms	199
5.5	Matrix factorization	203
5.5.1	Orthogonal-triangular decomposition	203
5.5.2	LU decomposition	204
5.5.3	Cholesky decomposition	204
5.6	Positive definite matrices and quadratic forms	206
5.6.1	Quadratic forms	206
5.6.2	Real symmetric non-negative definite matrix	207
5.6.2.1	Characterization	207
5.6.2.2	Decomposition and transformation	210
5.6.2.3	Matrix square root	211
5.6.2.4	Maximization of quadratic forms	212
5.6.2.5	Gramian matrix	214
5.6.3	Completing the square	215
5.7	Matrix norm and spectral estimation	216
5.7.1	Basics	216
5.7.2	Singularity from matrix norm and spectral radius	217
5.7.3	Gerschgorin theorem	218
5.7.4	Irreducible matrix and stronger results	219
5.8	Pseudoinverse of matrix	220
5.8.1	Pseudoinverse for full rank system	220
5.8.2	Pseudoinverse for general matrix	222
5.8.3	Application in linear systems	224
5.9	Multilinear forms	227
5.9.1	Bilinear forms	227
5.9.2	Multilinear forms	228
5.10	Determinant	231
5.10.1	Basic properties	231
5.10.2	Vandermonde matrix and determinant	237
5.11	Numerical iteration analysis	239
5.11.1	Numerical linear equation solution	239
5.11.1.1	Goals and general principles	239
5.11.1.2	Jacobi algorithm	239
5.11.1.3	Gauss Seidel algorithm	240
5.11.2	Power method for eigen-decomposition	240

5.12	Appendix: supplemental results for polynomials	243
5.12.1	Basics	243
5.12.2	Factorization of polynomial over \mathbb{C}	244
5.12.3	Factorization of polynomial over \mathbb{R}	245
5.13	Notes on bibliography	247
6	BASIC FUNCTIONAL ANALYSIS	249
6.1	Normed vector space	251
6.1.1	Basic properties	251
6.1.2	Equivalence of norms	253
6.2	Contraction mapping and fixed point theorems	255
6.2.1	Complete normed space (Banach space)	255
6.2.2	Contraction mapping	256
6.2.3	Banach fixed point theorem	257
6.2.4	Applications in root finding	259
6.2.5	Application to numerical linear equations	259
6.2.6	Applications to integral and differential equations	260
6.3	Inner product space and Hilbert space	263
6.3.1	Inner product space (pre-Hilbert space) and Hilbert space	263
6.3.1.1	Foundations	263
6.3.2	Hilbert spaces	265
6.3.2.1	Basics	265
6.3.3	Orthogonal decomposition of Hilbert spaces	266
6.3.3.1	Orthogonality	266
6.3.4	Projection and orthogonal decomposition	267
6.4	Approximations in Hilbert space	270
6.4.1	Approximation via projection	270
6.4.2	Application examples	271
6.4.2.1	Orthogonal projection and normal equations in \mathbb{R}^n	271
6.4.2.2	Approximation by continuous polynomials	273
6.4.2.3	Legendre polynomial via Gram-Schmidt process	274
6.5	Orthonormal systems	275
6.5.1	Basic definitions	275
6.5.2	Gram-Schmidt process	275
6.5.3	Properties of orthonormal systems	275
6.5.4	Orthonormal expansion in Hilbert space	277
6.5.5	Complete orthonormal system	278
6.5.5.1	Weierstrass approximation theorem for polynomials	280
6.5.5.2	Examples of complete orthonormal function set	280
6.6	Theory for trigonometric Fourier Series	282
6.6.1	Basic definitions	282
6.6.2	Completeness of Fourier series	284

6.6.3	Complex representation	285
6.7	Fourier transform	287
6.7.1	Definitions and basic concepts	287
6.7.2	Convolution theorem	289
6.7.3	Fourier transform and Fourier series	290
6.7.4	Discrete Fourier transform	291
6.7.4.1	Properties	291
6.8	Notes on bibliography	294

ii mathematical optimization methods

7	UNCONSTRAINED NONLINEAR OPTIMIZATION	297
7.1	Optimality conditions	299
7.1.1	Optimality concepts	299
7.1.2	Necessary and sufficient conditions	301
7.1.3	Special case: unconstrained quadratic programming	304
7.2	Line search method	306
7.2.1	A generic algorithm	306
7.2.2	Theory and computation of descent directions	307
7.2.2.1	Gradient descent direction and properties	307
7.2.2.2	Curvature-modified descent direction	308
7.2.2.3	Quasi-Newton method	310
7.2.2.4	Subspace optimization in quadratic forms	312
7.2.3	Theory and computation of step length	313
7.2.3.1	Overview	313
7.2.3.2	Lipschitz bounded convex functions	313
7.2.3.3	Backtracking-Armijo step size search	316
7.2.3.4	Wolfe condition	318
7.2.4	Complete algorithms	319
7.3	Trust region method	321
7.3.1	Motivation and the framework	321
7.3.2	Cauchy point method	322
7.3.3	Exact solution method	324
7.3.4	Approximate method	324
7.4	Conjugate gradient method	327
7.4.1	Motivating problems	327
7.4.2	Theory conjugate direction	328
7.4.3	Linear conjugate gradient algorithm	329
7.5	Least square problems	331
7.5.1	Linear least square theory and algorithm	331
7.5.1.1	Linear least square problems	331
7.5.1.2	SVD methods	332

7.5.1.3	Extension to L^p norm optimization	332
7.5.2	nonlinear least square problem	333
7.5.3	Line search Gauss-Newton method	334
7.5.4	Trust region method	336
7.5.5	Application: roots for nonlinear equation	336
7.6	Notes on bibliography	338
8	CONSTRAINED NONLINEAR OPTIMIZATION	340
8.1	Quadratic optimization I: equality constraints	342
8.1.1	Problem formulation	342
8.1.2	Optimality condition	342
8.1.2.1	General case	342
8.1.2.2	Positive semi-definitive quadratic programming	345
8.1.3	Solving KKT systems	346
8.1.3.1	Factorization approach	346
8.1.3.2	Range space approach	347
8.1.4	Linear least square with linear constraints	347
8.1.4.1	Least norm problem	347
8.1.5	Application: Markovitz Portfolio Optimization Model	349
8.2	Quadratic optimization II: inequality constraints	352
8.2.1	Problem formulation	352
8.2.2	Optimality conditions	352
8.2.2.1	Pure inequality case	352
8.2.2.2	General constrained optimization	354
8.2.2.3	Positive semi-definitive quadratic programming	355
8.2.3	Primal active-set method	356
8.2.4	Gradient projection method	361
8.2.5	Dual convex quadratic programming	362
8.3	General equality constrained optimization	364
8.3.1	Feasible path and optimality	364
8.3.2	Constraint qualification and Lagrange theory	365
8.3.3	Second order condition	368
8.4	General inequality constrained optimization	372
8.4.1	Feasible path and optimality	372
8.4.2	Constraint qualifications and KKT conditions	373
8.4.3	Second order conditions	377
8.5	Envelope theorem and sensitive analysis	381
8.6	Notes on bibliography	384
9	LINEAR OPTIMIZATION	386
9.1	Equality constrained linear programming	387
9.2	Inequality constrained linear programming	389
9.2.1	Linear optimization with inequality constraints	389

9.2.2	Geometry of linear programming	389
9.2.3	Optimality property and condition	391
9.2.4	Standard form of linear programming	392
9.2.5	Application examples	393
9.3	Linear programming geometry and simplex algorithm	394
9.3.1	Geometrical approach to linear programming	394
9.3.1.1	Overview	394
9.3.1.2	Vertex and optimality	394
9.3.1.3	Descent direction at a vertex	398
9.3.1.4	Stepping along a descent direction	399
9.3.2	The simplex algorithm	400
9.4	Interior point method	401
9.4.1	Optimality condition	401
9.4.2	Newton step and perturbed system	403
9.4.3	Algorithms	405
9.5	Notes on bibliography	407
10	CONVEX ANALYSIS AND CONVEX OPTIMIZATION	409
10.1	Affine sets	411
10.1.1	Basic concepts	411
10.1.2	Affine independence and dimensions	413
10.2	Convex sets and properties	417
10.2.1	Concepts of convex sets	417
10.2.2	Projection theorems	419
10.2.3	Separation theorems	420
10.2.3.1	Separating hyperplane theorem	420
10.2.3.2	Farka's lemma	422
10.3	Convex functions	425
10.3.1	Basic concepts	425
10.3.2	Connection to convex set	427
10.3.3	Strongly convex functions	427
10.3.4	Operations preserve convexity	428
10.3.5	Convexity and derivatives	429
10.3.6	Subgradient	431
10.4	Duality theory	433
10.5	Convex optimization and optimality conditions	437
10.5.1	Local optimality vs. global optimality	437
10.5.2	Unconstrained optimization optimality conditions	438
10.5.3	Constrained optimization optimality conditions	438
10.6	Subgradient methods	443
10.6.1	A generic algorithm for unconstrained problem	443
10.6.2	Convergence under Lipschitz smoothness	445

10.6.3	Projected gradient methods	448
10.6.3.1	Foundations	448
10.6.3.2	Algorithms	450
10.6.4	Proximal gradient methods	451
10.6.4.1	Foundations	451
10.6.4.2	Algorithms	452
10.6.4.3	Case study: sparsity regularization problem	453
10.7	Notes on bibliography	455

iii classical statistical methods

11	PROBABILITY THEORY	458
11.1	σ algebra	462
11.1.1	σ algebra concepts	462
11.1.2	Generation of sigma algebra	462
11.1.3	Partition of sample space	463
11.1.4	Filtration & information	463
11.1.5	Borel σ algebra	464
11.1.6	Measurable set and measurable space	465
11.2	Probability space	468
11.2.1	Event, sample point and sample space	468
11.2.2	Probability space	468
11.2.3	Properties of probability measure	470
11.2.4	Conditional probability	471
11.2.4.1	Basics	471
11.2.4.2	Bayes' theorem	472
11.2.4.3	Independence of events and sigma algebra	473
11.3	Measurable map and random variable	476
11.3.1	Random variable	476
11.3.2	σ algebra of random variables	477
11.3.3	Independence of random variables	478
11.4	Distributions of random variables	480
11.4.1	Basic concepts	480
11.4.1.1	Probability mass function	480
11.4.1.2	Distributions on \mathbb{R}^n	480
11.4.1.3	Probability density function	481
11.4.1.4	Conditional distributions	482
11.4.1.5	Bayes law	483
11.4.2	Independence	484
11.4.3	Conditional independence	486
11.4.4	Transformations	486
11.4.4.1	Transformation for univariate distribution	486

11.4.4.2	Location-scale transformation	487
11.4.4.3	Transformation for multivariate distribution	488
11.5	Expectation, variance, and covariance	492
11.5.1	Expectation	492
11.5.2	Expectation in the Lebesgue framework	493
11.5.3	Properties of expectation	495
11.5.4	Variance and covariance	495
11.5.5	Conditional variance	496
11.5.6	Delta method	497
11.6	Moment generating functions and characteristic functions	499
11.6.1	Moment generating function	499
11.6.2	Characteristic function	502
11.6.3	Joint moment generating functions for random vectors	503
11.6.4	Cumulants	504
11.7	Conditional expectation	506
11.7.1	General intuitions	506
11.7.2	Formal definitions	506
11.7.3	Different versions of conditional expectation	508
11.7.3.1	Conditioning on an event	508
11.7.3.2	Conditioning on a discrete random variable as a new random variable	508
11.7.3.3	Condition on random variable vs. event vs σ algebra	509
11.7.4	Properties	509
11.7.4.1	Linearity	509
11.7.4.2	Taking out what is known	510
11.7.4.3	Law of total expectation	510
11.7.4.4	Law of iterated expectations	512
11.7.4.5	Conditioning on independent random variable/ σ algebra	512
11.7.4.6	Least Square minimizing property	513
11.8	The Hilbert space of random variables	514
11.8.1	Definitions	514
11.8.2	Subspaces, projections, and approximations	514
11.8.3	Connection to conditional expectation	519
11.9	Probability inequalities	522
11.9.1	Chebychev inequalities	522
11.9.2	Jensen's inequality	523
11.9.3	Holder's, Minkowski, and Cauchy-Schwarz inequalities	524
11.9.4	Popoviciu's inequality for variance	526
11.10	Convergence of random variables	528
11.10.1	Different levels of equivalence among random variables	528

11.10.2	Convergence almost surely	528
11.10.3	Convergence in probability	529
11.10.4	Mean square convergence	530
11.10.5	Convergence in distribution	531
11.11	Law of Large Number and Central Limit theorem	533
11.11.1	Law of Large Numbers	533
11.11.2	Central limit theorem	535
11.12	Finite sampling models	538
11.12.1	Counting principles	538
11.12.2	Matching problem	541
11.12.3	Birthday problem	543
11.12.4	Coupon collection problem	544
11.12.5	Balls into bins model	545
11.13	Order statistics	548
11.14	Information theory	552
11.14.1	Concept of entropy	552
11.14.2	Entropy maximizing distributions	554
11.14.3	KL divergence	558
11.14.4	Conditional entropy and mutual information	559
11.14.5	Cross-entropy	561
11.15	Notes on bibliography	564
12	STATISTICAL DISTRIBUTIONS	567
12.1	Common distributions and properties	569
12.1.1	Overview	569
12.1.2	Bernoulli distribution	569
12.1.3	Poisson distribution	569
12.1.4	Geometric distribution	571
12.1.5	Binomial distribution	572
12.1.6	Normal distribution	574
12.1.7	Half-normal distribution	577
12.1.8	Laplace distribution	577
12.1.9	Multivariate Gaussian/normal distribution	578
12.1.9.1	Basic definitions	578
12.1.9.2	Affine transformation and its consequences	580
12.1.9.3	Marginal and conditional distribution	581
12.1.9.4	Box Muller transformation	584
12.1.10	Lognormal distribution	584
12.1.10.1	Univariate lognormal distribution	584
12.1.10.2	Extension to univariate lognormal distribution	586
12.1.10.3	Multivariate lognormal distribution	588
12.1.11	Exponential distribution	589

12.1.12	Gamma distribution	590
12.1.13	Hypergeometric distribution	592
12.1.14	Beta distribution	592
12.1.15	Multinomial distribution	594
12.1.16	Dirichlet distribution	596
12.1.17	χ^2 -distribution	597
12.1.17.1	Basic properties	597
12.1.17.2	Noncentral chi-squared distribution	598
12.1.18	Wishart distribution	599
12.1.19	t -distribution	600
12.1.19.1	Standard t distribution	600
12.1.19.2	classical t distribution	601
12.1.19.3	Multivariate t distribution	601
12.1.20	F -distribution	602
12.1.21	Empirical distributions	603
12.1.22	Heavy-tailed distributions	604
12.1.22.1	Basic characterization	604
12.1.22.2	Pareto and power distribution	604
12.1.22.3	Student t distribution family	605
12.1.22.4	Gaussian mixture distributions	606
12.2	Characterizing distributions	608
12.2.1	Skewness and kurtosis	608
12.2.2	Percentiles and quantiles	610
12.2.2.1	Basics	610
12.2.2.2	Cornish-Fisher expansion	612
12.3	Moment matching approximation methods	614
12.4	Gaussian quadratic forms	616
12.4.1	Quadratic forms and chi-square distribution	616
12.4.2	Applications	619
12.5	Notes on bibliography	623
13	STATISTICAL ESTIMATION THEORY	625
13.1	Parameter estimators	628
13.1.1	Overview	628
13.1.2	Statistic and Estimator	629
13.1.2.1	Statistic	629
13.1.2.2	Estimator properties	630
13.1.2.3	Variance-bias decomposition	632
13.1.2.4	Consistence	634
13.1.2.5	Efficiency	636
13.1.2.6	Robust statistics	637
13.1.3	Method of moments	638

13.1.4	Maximum likelihood estimation	639
13.1.4.1	Basic concepts	639
13.1.4.2	MLE examples	641
13.1.4.3	Bias and consistence of MLE	644
13.2	Information and efficiency	647
13.2.1	Fisher information	647
13.2.2	Cramer-Rao lower bound	651
13.2.2.1	Preliminary: information inequality	651
13.2.2.2	Cramer-Rao lower bound: univariate case	652
13.2.2.3	Cramer-Rao lower bound: multivariate case	653
13.2.3	Efficient estimators	655
13.2.4	Asymptotic normality and efficiency of MLE	656
13.3	Sufficiency and data reduction	658
13.3.1	Sufficient estimators	658
13.3.2	Factorization theorem	659
13.4	Bayesian estimation theory	662
13.4.1	Overview	662
13.4.2	Fundamentals	662
13.4.2.1	Basics	662
13.4.2.2	Bayesian prediction	664
13.4.3	Bayesian estimation for Gaussians	665
13.4.3.1	Univariate Bayesian estimation	665
13.4.3.2	Multivariate Bayesian estimation	668
13.4.4	Estimating variance with known mean	669
13.5	Bootstrap method	671
13.6	Expectation Maximization Algorithm	673
13.7	Hypothesis testing theory	674
13.7.1	Basics	674
13.7.2	Characterizing errors and power	677
13.7.3	Power of a statistical test	679
13.7.4	Common statistical tests	680
13.7.4.1	Chi-square goodness-of-fit test	680
13.7.4.2	Chi-square test for statistical independence	683
13.7.4.3	Kolmogorov-Smirnov goodness-of-fit test	683
13.8	Hypothesis testing on normal distributions	684
13.8.1	Normality test	684
13.8.2	Sample mean with known variance	684
13.8.3	Sample mean with unknown variance	686
13.8.4	Variance test	686
13.8.5	Variance comparison test	687
13.8.6	Person correlation t test	687

13.8.7	Two sample tests	688
13.8.7.1	Two-sample z test	688
13.8.7.2	Two-sample t test	688
13.8.7.3	Paired Data	689
13.8.8	Interval estimation for normal distribution	689
13.9	Notes on bibliography	691
14	MULTIVARIATE STATISTICAL METHODS	693
14.1	Multivariate data and distribution	695
14.1.1	Sample statistics	695
14.1.2	Multivariate Gaussian distribution	696
14.1.3	Estimation methods	698
14.1.3.1	Maximum likelihood estimation	698
14.1.3.2	Weighted estimation	701
14.2	Principal component analysis (PCA)	702
14.2.1	Statistical fundamentals of PCA	702
14.2.1.1	PCA for random vectors	702
14.2.1.2	Sample principal components	703
14.2.2	Geometric fundamentals of PCA	706
14.2.2.1	Optimization approach	706
14.2.2.2	Properties	708
14.2.3	Probabilistic PCA	709
14.2.4	Applications	711
14.2.4.1	Eigenfaces and eigendigits	711
14.2.4.2	Interest rate curve dynamics modeling	713
14.3	Covariance structure and factor analysis	717
14.3.1	The orthogonal factor model	717
14.3.1.1	Motivation and factor models	717
14.3.1.2	Covariance structure implied by factor model	717
14.3.2	Latent linear generative model	719
14.3.3	Parameter estimation	722
14.3.3.1	Data collection and preparation	722
14.3.3.2	PCA method	723
14.3.3.3	Maximum likelihood method	724
14.3.4	Factor score estimation	724
14.3.5	Application I: Joint default modeling	725
14.3.5.1	Single factor model	725
14.3.5.2	Multiple factor model	728
14.3.6	Application II: factor models for stock return	729
14.3.6.1	Overview	729
14.3.6.2	The Fama-French 3 factor model	731
14.4	Notes on Bibliography	735

15	LINEAR REGRESSION ANALYSIS	737
15.1	Linear regression analysis: basics	740
15.1.1	Linear regression models	740
15.1.2	Ordinary least square (OLS): fundamentals	743
15.1.2.1	Review on orthogonal projections	743
15.1.2.2	OLS results	743
15.1.2.3	OLS results with demeaned data	749
15.1.2.4	Gauss-Markov theorem	752
15.1.2.5	Variance decomposition	753
15.1.2.6	Residual and variance estimation	756
15.1.3	Ordinary least square (OLS): Additional topics	757
15.1.3.1	Orthogonal input and successive regression	757
15.1.3.2	Frisch-Waugh-Lovell(FWL) theorem and partial regression	759
15.1.3.3	Forecasting analysis with normality assumption	760
15.1.4	Hypothesis testing and analysis of variance	762
15.1.4.1	Distribution of coefficients	763
15.1.4.2	t test and normality test of single coefficients	765
15.1.4.3	F lack-of-fit test	767
15.1.4.4	χ^2 test for variance	769
15.1.5	Maximum likelihood method with normality assumption	770
15.1.6	Asymptotic properties of least square solutions	772
15.1.6.1	Asymptotic properties of standard OLS	772
15.1.6.2	Asymptotic efficiency of standard OLS	774
15.1.7	Partial and multiple correlation	774
15.1.7.1	Multiple correlation coefficient, R^2	774
15.1.7.2	Partial correlation coefficient	777
15.1.8	Generalized linear regression (GLR)	778
15.1.8.1	Linear regression with structural error	778
15.1.8.2	Generalized least square solution	779
15.1.8.3	Gauss-Markov theorem for GLR	781
15.1.8.4	Feasible GLS	782
15.1.9	Linear structure in joint distributions	782
15.2	Model specification and selection	785
15.2.1	Model order mis-specification	785
15.2.1.1	Omission of relevant regressors	785
15.2.1.2	Inclusion of irrelevant regressors	786
15.2.2	Model selection methods	788
15.2.2.1	Adjusted R square method	788
15.2.2.2	F test method	788
15.2.2.3	Information criterion methods	790

15.2.2.4	Bayesian information criterion (BIC)	791
15.2.3	Test for structure change	792
15.3	Linear regression analysis: diagnostics & solutions	794
15.3.1	Multi-collinearity	794
15.3.1.1	Detection and characterization	794
15.3.1.2	Regressor linear regression and variance inflation factor	794
15.3.1.3	Principal component linear regression (PCLR)	797
15.3.2	Rank deficiency and rigid regression	797
15.3.3	Heteroskedasticity	799
15.3.3.1	Test for heteroskedasticity	799
15.3.3.2	Heteroskedasticity robust estimator	800
15.3.3.3	Feasible weighted least square	800
15.3.4	Residual normality test	802
15.3.4.1	Jarque-Bera test	802
15.3.4.2	D'Agostino's K^2 test	802
15.3.5	Autocorrelation of errors	803
15.3.5.1	Motivation and general remarks	803
15.3.5.2	Test of autocorrelation of errors	804
15.3.5.3	Models with known autocorrelation	806
15.3.5.4	Transformation to generalized linear regression	807
15.3.6	Outliers analysis and robust linear regression	809
15.3.6.1	Outliers and influential points	809
15.3.6.2	Outlier impact analysis	810
15.3.6.3	Robust M-estimation linear regression	813
15.3.7	Visual diagnosis	816
15.4	Linear regression case studies	819
15.4.1	Standard linear regression	819
15.4.2	Boston Housing example	821
15.5	Multivariate multiple linear regression (MMLR)	824
15.5.1	Canonical MMLR	824
15.5.1.1	Motivation and model	824
15.5.1.2	Ordinary least square solution	825
15.5.2	Reduced rank regression	826
15.5.2.1	Fundamentals	826
15.5.2.2	Application in liquidity factor modeling	830
15.6	Notes on Bibliography	833
16	MONTE CARLO METHODS	836
16.1	Generating random variables	838
16.1.1	Inverse transform method	838
16.1.2	Box-Muller method for standard normal random variable	840

16.1.3	Acceptance-rejection method	840
16.1.4	Composition approach	842
16.1.5	Generate dependent continuous random variables	844
16.1.5.1	Multivariate normal and lognormal distribution	844
16.1.5.2	Multivariate student t distribution	844
16.1.5.3	General joint distribution	845
16.1.6	Generate discrete random variables	845
16.1.6.1	Generate single discrete random variables	845
16.1.6.2	Generate correlated discrete random variables	846
16.2	Monte Carlo integration	849
16.2.1	Naive approach	849
16.2.2	Importance sampling	850
16.3	Markov chain Monte Carlo	853
16.3.1	Basics	853
16.3.1.1	Markov chain Monte Carlo (MCMC)	853
16.3.2	Metropolis-Hasting algorithm	854
16.3.3	Gibbs sampling	856
16.4	Monte Carlo for random processes	857
16.4.1	Simulating stochastic differential equations	857
16.4.1.1	Simulating Brownian motion	857
16.4.1.2	Simulating linear arithmetic SDE	857
16.4.1.3	Simulating linear geometric SDE	858
16.4.1.4	Simulation mean-reversion(OU) process	858
16.4.2	Stochastic interpolation	859
16.4.2.1	Interpolating Gaussian processes	859
16.4.2.2	Interpolating one Dimensional Brownian motions	859
16.4.2.3	Interpolating multi-dimensional Brownian motions	862
16.5	Monte Carlo variance reduction	865
16.5.1	Antithetic sampling	865
16.5.1.1	Basic principles	865
16.5.1.2	Methods and analysis	866
16.5.2	Control variates	868
16.5.2.1	Basic principles	868
16.5.2.2	Multiple control variates	870
16.6	Notes on bibliography	871

iv dynamics modeling methods

17	MODELS AND ESTIMATION IN LINEAR SYSTEMS	874
17.1	Difference equation	876
17.1.1	Introduction	876
17.1.2	Solution structure of linear difference equations	877

17.1.3	Solution to non-homogeneous equation	879
17.1.4	Linear equations with constant coefficients	880
17.1.4.1	Basic case	880
17.1.4.2	General case	883
17.2	Differential equations	886
17.2.1	Linear differential equations	886
17.2.1.1	Concepts	886
17.2.1.2	Wronskian and linear independence	887
17.2.1.3	General solution theory	890
17.2.1.4	Existence & uniqueness of solution	893
17.2.2	Linear homogeneous differential equations with constant coefficients	894
17.2.2.1	The key identity	894
17.2.2.2	The case of real roots	896
17.2.2.3	The case of complex roots	896
17.2.2.4	The complete solution set	897
17.2.3	Solution to non-homogeneous ODEs	900
17.2.3.1	General principles	900
17.2.3.2	Key identity approach	901
17.2.4	First order linear differential equation	907
17.3	Linear system	909
17.3.1	Solution space for linear homogeneous system	909
17.3.2	Linear independence and the Wronskian	911
17.3.3	The fundamental system and solution method	913
17.3.4	The non-homogeneous linear equation	914
17.3.5	Conversion of linear differential/difference equation to linear systems	918
17.3.6	Solution method for discrete system	919
17.4	Linear system with constant coefficients	920
17.4.1	General solutions	920
17.4.2	System eigenvector method: continuous-time system	920
17.4.2.1	Diagonalizable system	920
17.4.2.2	two-by-two non-diagonalizable system	927
17.4.2.3	Non-diagonalizable system	928
17.4.3	Equilibrium point	930
17.4.3.1	Discrete-time system	930
17.4.3.2	Continuous-time system	931
17.4.4	Stability	932
17.4.5	Complex eigenvalues/eigenvectors	934
17.4.6	Boundedness of linear systems	934
17.4.7	One dimensional Nonlinear dynamical system analysis	935

17.5	Notes on bibliography	937
18	ESTIMATION IN DYNAMICAL SYSTEMS	939
18.1	Least square estimation of constant vectors	940
18.1.1	linear static estimation from single measurement with no prior information	940
18.1.2	linear static estimation from single measurement with prior information	941
18.1.3	Batch and recursive least square estimation with multiple measurements	942
18.1.4	Nonlinear least square estimation	944
18.2	Kalman filter	946
18.2.1	Preliminary: error propagation in linear systems	946
18.2.1.1	Discrete-time system	946
18.2.1.2	Continuous-time system	946
18.2.2	Batch estimation	947
18.2.3	From batch estimation to Kalman filter	948
18.2.4	Extended Kalman filter for nonlinear system	949
18.3	Application in robotics	951
18.3.1	Quaternions	951
19	STOCHASTIC PROCESS	953
19.1	Stochastic process	955
19.1.1	Basic definition and concepts	955
19.1.2	Stationarity	957
19.2	Gaussian process	960
19.2.1	Basic Gaussian process	960
19.2.2	Stationarity	961
19.3	Brownian motion (Wiener process)	962
19.3.1	Definition and elementary properties	962
19.3.2	Multi-dimensional Brownian motion	964
19.3.3	Asymptotic behaviors	965
19.3.4	The reflection principle	966
19.3.5	Quadratic variation	967
19.3.6	Discrete-time approximations and simulation	969
19.4	Brownian motion variants	970
19.4.1	Gaussian process generated by Brownian motion	970
19.4.2	Brownian bridge	971
19.4.2.1	Constructions	971
19.4.2.2	Applications	975
19.4.3	Geometric Brownian motion	975
19.5	Poisson process	977
19.5.1	Basics	977

19.5.2	Arrival and Inter-arrival Times	978
19.6	Martingale theory	980
19.6.1	Preliminaries: Filtration and adapted process	980
19.6.1.1	Basic concepts in filtration	980
19.6.1.2	Filtration for Brownian motion	982
19.6.2	Basics of martingales	982
19.6.3	Martingale transformation	985
19.7	Stopping time	986
19.7.1	Stopping time examples	986
19.7.1.1	First passage time	986
19.7.1.2	Trivial stopping time	986
19.7.1.3	Counter example: last exit time	987
19.7.2	Wald's equation	987
19.7.3	Optional stopping	988
19.7.4	martingale method for first hitting time	988
19.8	Notes on bibliography	989
20	STOCHASTIC CALCULUS	992
20.1	Ito stochastic integral	994
20.1.1	Motivation	994
20.1.2	Construction of Ito integral	994
20.1.2.1	Ito integral of a simple process	994
20.1.2.2	Ito integral of a general process	997
20.1.3	Ito integral with deterministic integrands (Wiener integral)	1001
20.1.3.1	Basics	1001
20.1.3.2	Integration by parts	1004
20.2	Stochastic differential equations	1007
20.2.1	Ito Stochastic differential equations	1007
20.2.2	Ito's lemma	1008
20.2.3	Useful results of Ito's lemma	1011
20.2.3.1	Product rule and quotient rule	1011
20.2.3.2	Logarithm and exponential	1012
20.2.3.3	Ito integral by parts	1012
20.2.3.4	Differentiate integrals of Ito process	1013
20.3	Linear SDE	1015
20.3.1	State-independent linear arithmetic SDE	1015
20.3.2	State-independent linear geometric SDE	1015
20.3.3	Multiple dimension extension	1017
20.3.4	Exact SDE	1019
20.3.5	Calculation mean and variance from SDE	1020
20.3.6	Integrals of Ito SDE	1021
20.4	Ornstein-Uhlenbeck(OU) process	1026

20.4.1	OU process	1026
20.4.1.1	Constant coefficient OU process	1026
20.4.1.2	Time-dependent coefficient OU process	1030
20.4.1.3	Integral of OU process	1032
20.4.2	Exponential OU process	1035
20.4.3	Parameter estimation for OU process	1037
20.4.4	Multiple factor extension	1037
20.5	Notes on bibliography	1040
21	MARKOV CHAIN AND RANDOM WALK	1042
21.1	Discrete-time Markov chain	1044
21.1.1	The model	1044
21.1.2	Evolution of discrete chain	1046
21.2	Classification of states	1048
21.2.1	accessibility and communicating classes	1048
21.2.2	Transient and recurrent states and classes	1050
21.2.2.1	Transient and recurrent states	1050
21.2.2.2	From states to classes	1054
21.2.2.3	Qualitative classification of recurrent and transient classes	1055
21.2.3	Periodicity	1056
21.2.4	Positive and null recurrent	1058
21.2.5	Summary	1060
21.3	Absorption analysis	1061
21.3.1	Matrix structure for adsorption analysis	1061
21.3.2	Absorbing Markov chains	1063
21.3.3	Hitting and return analysis	1065
21.3.4	Examples	1067
21.3.4.1	Consecutive coin toss game	1067
21.4	Limiting behavior & distributions	1069
21.4.1	Preliminary: eigenvalue propoties of stochastic matrices	1069
21.4.1.1	Preliminary: Frobenius-Perron matrix theory	1069
21.4.1.2	More general situations	1071
21.4.2	Limiting theorem	1073
21.4.2.1	Limiting distribution	1073
21.4.2.2	Extensions via Long run return analysis	1077
21.4.3	Application: PageRank algorithm	1079
21.5	Detailed balance and spectral properties	1081
21.6	Random walk	1083
21.6.1	Basic concepts and properties	1083
21.6.2	Persistent random walk	1083
21.6.3	Asymptotic properties	1085

21.6.4	Gambler's ruin problems	1085
21.7	Notes on bibliography	1090
22	TIME SERIES ANALYSIS	1092
22.1	Overview of time series analysis	1095
22.1.1	Introduction to time series	1095
22.1.2	Stationarity	1096
22.1.2.1	Stationarity concept	1096
22.1.2.2	Rolling analysis	1098
22.1.3	Remove trend and seasonality	1099
22.2	Linear stationary process theory	1102
22.2.1	Preliminaries: the lag operator and polynomial	1102
22.2.2	Linear process	1103
22.2.3	Autoregressive (AR) process	1105
22.2.3.1	Basics	1105
22.2.3.2	Stationarity and invertibility condition	1108
22.2.3.3	Forecasting	1109
22.2.4	Moving average (MA) process	1112
22.2.4.1	Basics	1112
22.2.4.2	Stationarity and invertibility	1115
22.2.4.3	Forecasting	1116
22.2.5	ARMA process	1120
22.2.5.1	Basic properties	1120
22.2.6	Unit root AR process	1121
22.2.6.1	Unit root process	1121
22.2.6.2	Trend stationarity vs. unit root process	1123
22.2.6.3	Unit root test	1123
22.2.6.4	Forecasting	1124
22.2.7	Correlation analysis	1124
22.2.7.1	Autocorrelation statistical analysis	1124
22.2.7.2	Partial autocorrelation function theory	1126
22.2.7.3	Correlogram analysis example	1128
22.2.8	Model analysis and calibration	1130
22.2.8.1	Order selection	1130
22.2.8.2	Yule-Walker equations and related methods	1131
22.2.8.3	Linear regression approach	1134
22.2.8.4	Maximum likelihood estimation	1136
22.2.8.5	Example: a toy example	1137
22.2.9	Wold Representation theorem	1140
22.3	Extensions to multivariate time series	1143
22.3.1	Introduction	1143
22.3.2	Vector autoregressive models	1144

22.3.2.1	VAR(1) model	1144
22.3.2.2	VAR(2) model	1146
22.3.2.3	VAR(p) model	1147
22.3.3	Vector moving-average model	1150
22.4	Autoregressive conditional heteroscedastic model	1153
22.4.1	ARCH models	1153
22.4.1.1	The motivation and the model	1153
22.4.1.2	Statistical properties	1155
22.4.1.3	Variance forecasting	1161
22.4.1.4	Detect ARCH effect	1164
22.4.1.5	Parameter estimation	1165
22.4.2	GARCH models	1165
22.4.2.1	The model	1165
22.4.2.2	Connecting GARCH to ARCH	1168
22.4.2.3	Variance forecasting	1168
22.5	Notes on Bibliography	1171

v statistical learning methods

23	SUPERVISED LEARNING PRINCIPLES	1174
23.1	The supervised learning problem	1175
23.1.1	Introduction	1175
23.1.2	Framework	1178
23.2	Underfitting and Overfitting	1180
23.3	Bias variance trade-off	1183
23.3.1	Introduction	1183
23.3.2	Variance-bias decomposition	1184
23.3.3	Estimating generalization error	1187
23.3.4	Bias variance analysis on linear regression	1188
23.4	Supervised learning algorithms	1190
23.4.1	Overview	1190
23.4.2	Inductive bias	1191
23.4.3	No free lunch (NFL) theorem	1192
23.5	Model loss functions	1194
23.5.1	Regression loss	1194
23.5.2	Classification loss	1196
23.6	Note on bibliography	1199
24	LINEAR MODELS FOR REGRESSION	1200
24.1	Standard linear regression	1201
24.1.1	Ordinary linear regression	1201
24.1.2	Application examples	1202
24.1.2.1	Boston housing prices	1202

24.2	Penalized linear regression	1206
24.2.1	Ridge regression	1206
24.2.1.1	Basics	1206
24.2.1.2	Dual form of ridge regression	1209
24.2.2	Lasso regression	1209
24.2.3	Elastic net	1212
24.2.4	Shrinkage Comparison	1212
24.2.5	Effective degree of freedom	1215
24.3	Basis function extension	1216
24.4	Note on bibliography	1218
25	LINEAR MODELS FOR CLASSIFICATION	1220
25.1	Logistic regression	1222
25.1.1	Logistic regression model	1222
25.1.2	Parameter estimation via maximum likelihood estimation	1223
25.1.3	Logistic regression with regularization	1227
25.1.4	Feature augmentation strategies	1229
25.1.5	Multinomial logistic regression	1230
25.1.6	Application examples	1231
25.1.6.1	South Africa heart disease	1231
25.1.6.2	Credit card fraud detection	1233
25.1.6.3	MNIST	1235
25.2	Gaussian discriminate analysis	1237
25.2.1	Linear Gaussian discriminant model	1237
25.2.1.1	The model	1237
25.2.1.2	Model parameter estimation	1238
25.2.1.3	Geometry of decision boundary	1238
25.2.2	Quadratic Gaussian discriminant model	1240
25.2.2.1	The model	1240
25.2.2.2	Model parameter estimation	1242
25.2.3	Application examples	1242
25.2.3.1	A toy example	1242
25.3	Fisher Linear discriminate analysis (Fisher LDA)	1246
25.3.1	One dimensional linear discriminant	1246
25.3.1.1	Basics	1246
25.3.1.2	Application in classification	1249
25.3.1.3	Possible issues	1251
25.3.2	Multi-dimensional linear discriminate	1252
25.3.2.1	Basics	1252
25.3.2.2	Application in classification	1253
25.3.3	Supervised dimensional reduction via Fisher LDA	1255
25.4	Separating hyperplane and Perceptron learning algorithm	1257

25.4.1	Basic geometry of hyperplanes	1257
25.4.2	The Perceptron learning algorithm	1258
25.5	Support vector machine classifier	1260
25.5.1	Motivation and formulation	1260
25.5.2	Optimality condition and dual form	1261
25.5.3	Soft margin SVM	1263
25.5.3.1	Basics	1263
25.5.3.2	Optimality condition for soft margin SVM	1264
25.5.3.3	Algorithm	1267
25.5.4	SVM with kernels	1268
25.5.5	A unified perspective from loss functions	1269
25.6	Kernel methods	1273
25.6.1	Basic concepts of kernels and feature maps	1273
25.6.2	Mercer's theorem	1273
25.6.3	Common kernels	1275
25.6.4	Kernel trick	1277
25.6.5	Elementary algorithms	1277
25.7	Note on bibliography	1279
26	GENERATIVE MODELS	1281
26.1	Naive Bayes classifier (NBC)	1282
26.1.1	Overview	1282
26.1.2	Binomial NBC	1282
26.1.3	Multinomial NBC	1284
26.1.4	Gaussian NBC	1287
26.1.5	Discussion	1289
26.2	Application	1290
26.2.1	Classifying documents using bag of words	1290
26.2.2	Credit card fraud prediction	1290
26.3	Supporting mathematical results	1293
26.3.1	Beta-binomial model	1293
26.3.1.1	The model	1293
26.3.1.2	Parameter inference	1294
26.3.2	Dirichlet-multinomial model	1296
26.3.2.1	The model	1296
26.3.2.2	Parameter inference	1299
27	K NEAREST NEIGHBORS	1301
27.1	Principles	1302
27.1.1	The algorithm	1302
27.1.2	Metrics and features	1303
27.2	Application examples	1305

28	TREE METHODS	1308
28.1	Preliminaries: entropy concepts	1309
28.1.1	Concept of entropy	1309
28.1.2	Conditional entropy and mutual information	1310
28.2	Classification tree	1314
28.2.1	Basic concepts of decision tree learning	1314
28.2.2	A generic tree-growth algorithm	1315
28.2.3	Splitting criterion	1316
28.2.4	Tree pruning	1320
28.2.5	Practical algorithms	1320
28.2.6	Examples	1323
28.2.6.1	Tree structures in Iris data classification	1323
28.3	Regression tree	1326
28.3.1	Basics	1326
28.3.2	Practical algorithms	1329
28.3.3	Examples	1329
28.3.3.1	A toy example	1329
28.3.3.2	Boston Housing prices	1330
29	ENSEMBLE AND BOOSTING METHODS	1333
29.1	Motivation and overview	1334
29.2	Voting	1335
29.3	Bagging Methods	1337
29.3.1	A basic bagging method	1337
29.3.2	Tree bagging	1338
29.3.3	Random Forest	1340
29.4	Adaboost	1343
29.4.1	Adaboost classifier	1343
29.4.2	Adaboost regressor	1348
29.4.3	Additive model framework	1349
29.4.3.1	Generic additive model algorithm	1349
29.4.3.2	Adaboost as a special additive model	1350
29.5	Gradient boosting machines	1352
29.5.1	Fundamental	1352
29.5.2	Gradient boosting tree	1354
29.5.2.1	The algorithm	1354
29.5.2.2	Regression loss	1355
29.5.2.3	Classification loss	1356
29.5.2.4	Practicals	1357
29.5.3	XGBoost	1358
29.6	Notes on Bibliography	1363
30	UNSUPERVISED STATISTICAL LEARNING	1365

30.1	Singular value decomposition (SVD) and matrix factorization	1367
30.1.1	SVD theory	1367
30.1.1.1	SVD fundamentals	1367
30.1.1.2	SVD and matrix norm	1369
30.1.1.3	SVD low rank approximation	1370
30.1.2	Principal component analysis (PCA)	1372
30.1.2.1	Statistical perspective of PCA	1372
30.1.2.2	Geometric fundamentals of PCA	1375
30.1.2.3	Robust PCA with outliers	1377
30.1.3	Sparse coding and dictionary learning	1379
30.1.3.1	Sparse coding	1379
30.1.3.2	Dictionary learning	1380
30.1.3.3	Online dictionary learning	1382
30.1.4	Non-negative matrix factorization	1384
30.2	Advanced applications of matrix factorization methods	1386
30.2.1	Latent semantic analysis	1386
30.2.2	Collaborative filtering in recommender systems	1389
30.3	Manifold learning	1395
30.3.1	Overview	1395
30.3.2	Preliminary: multidimensional scaling (MDS)	1395
30.3.2.1	Motivation	1395
30.3.2.2	Solution to classical MDS	1396
30.3.3	Isomap	1399
30.3.4	Kernel PCA	1401
30.3.5	Laplacian eigenmap	1403
30.3.5.1	Preliminary: graph Laplacian	1403
30.3.5.2	Laplacian eigenmap	1405
30.3.6	Diffusion map	1408
30.3.7	Application examples	1411
30.3.7.1	MNIST	1411
30.4	Clustering	1413
30.4.1	Overview	1413
30.4.2	K-means	1413
30.4.2.1	Canonical K-means	1413
30.4.2.2	K means++	1416
30.4.2.3	Kernel K means	1417
30.4.3	Density-based spatial clustering of applications with noise (DB-SCAN)	1418
30.4.4	Spectral clustering	1420
30.4.5	Gaussian mixture models (GMM)	1421

30.4.5.1	Preliminaries: Expectation Maximization (EM) algorithm	1421
30.4.5.2	The GMM model and algorithm	1423
30.4.6	Hierarchical clustering	1425
30.4.7	Application examples	1426
30.4.7.1	Image segmentation	1426
30.5	Notes on Bibliography	1428
31	PRACTICAL STATISTICAL LEARNING	1431
31.1	Model evaluation metrics	1433
31.1.1	Regression metrics	1433
31.1.2	Classification metrics	1434
31.1.3	ROC and PRC metrics	1436
31.1.4	Metrics for imbalanced data	1438
31.2	Model selection methods	1440
31.2.1	The training-validation-testing idea	1440
31.2.2	Cross-validation	1440
31.3	Data and feature engineering	1445
31.3.1	Data preprocessing	1445
31.3.1.1	Data standardization	1445
31.3.1.2	Data normalization	1446
31.3.1.3	Handle categorical data	1446
31.3.1.4	Handle missing values	1447
31.3.1.5	Dimensional reduction	1448
31.3.1.6	Centering kernel matrix	1448
31.3.2	Feature engineering I: basic routines	1448
31.3.2.1	Nonlinear transformation	1448
31.3.2.2	Polynomial features	1449
31.3.2.3	Binning	1449
31.3.3	Feature engineering II: feature selection	1449
31.3.3.1	Overview	1449
31.3.3.2	Filtering methods	1450
31.3.3.3	Recursive elimination methods	1451
31.3.3.4	Regularization methods	1451
31.3.4	Feature engineering III: feature extraction	1452
31.3.4.1	Text analytics	1452
31.3.4.2	Image	1453
31.3.4.3	Time series	1453
31.3.5	Imbalanced data	1454
31.3.5.1	Motivations	1454
31.3.5.2	Data resampling: undersampling	1455
31.3.5.3	Data resampling: upsampling	1455

31.3.5.4	Choice of loss functions, algorithms, and metrics	1456
31.4	Note on bibliography	1457

vi deep learning methods

32	FOUNDATIONS OF DEEP LEARNING	1459
32.1	Neural network fundamentals	1460
32.1.1	From machine learning to deep learning	1460
32.1.2	Neurons and neural networks	1461
32.1.2.1	Artificial neurons	1461
32.1.2.2	Artificial neural networks	1464
32.1.3	Universal approximation	1466
32.1.4	Training via backpropagation	1467
32.2	Feed-forward neural network	1473
32.2.1	Regression and classification	1473
32.2.1.1	Linear regression and classification	1473
32.2.1.2	Nonlinear extension	1476
32.2.1.3	Radial basis function network	1478
32.2.2	Image classification	1480
32.2.3	Recommender systems	1482
32.2.4	Approximating numerical partial differential equations	1483
33	NETWORK TRAINING AND OPTIMIZATION	1488
33.1	Optimization algorithms	1490
33.1.1	Motivation	1490
33.1.2	Full Batch gradient descent	1491
33.1.3	Minibatch stochastic gradient descent	1492
33.1.4	Adaptive gradient method	1493
33.1.4.1	Adaptive gradient (AdaGrad)	1493
33.1.4.2	RMSProp & AdaDelta	1493
33.1.5	Momentum method	1495
33.1.6	Combined together: adaptive momentum (Adam)	1496
33.2	Training and regularization techniques	1498
33.2.1	Choices of activation functions	1498
33.2.2	Weight initialization	1498
33.2.2.1	Motivation	1498
33.2.2.2	Xavier initialization	1499
33.2.2.3	He initialization	1500
33.2.3	Data normalization	1500
33.2.3.1	Initial data standardization	1500
33.2.3.2	Batch normalization	1501
33.2.3.3	Layer normalization	1503
33.2.3.4	Instance norm and batch norm	1504

33.2.4	Regularization	1505
33.2.4.1	L_p regularization	1505
33.2.4.2	Weight decay	1505
33.2.4.3	Early stopping	1506
33.2.4.4	Dropout	1506
33.2.4.5	Data augmentation	1507
33.2.4.6	Label smoothing	1508
33.3	Notes on Bibliography	1509
34	CONVOLUTIONAL NEURAL NETWORKS	1512
34.1	Foundations	1513
34.1.1	Convolutional connection	1513
34.1.2	Pooling and translational invariance	1516
34.1.3	Special convolution units	1518
34.1.3.1	The 1×1 convolution	1518
34.1.3.2	Dilated convolution	1519
34.2	CNN classical architectures	1521
34.2.1	LeNet	1521
34.2.2	AlexNet	1521
34.2.3	VGG	1523
34.2.4	ResNet	1524
34.3	Image recognition	1529
34.3.1	Image classification	1529
34.4	Visualizing CNN	1530
34.4.1	Visualizing filters	1530
34.4.2	Visualizing classification activation map	1531
34.5	Autoencoders and denoising	1533
34.5.1	Autoencoders	1533
34.5.2	Denoising autoencoder	1534
34.6	Neural style transfer	1536
34.7	Visual based deep reinforcement learning	1539
35	RECURRENT NEURAL NETWORKS	1543
35.1	Recurrent neural network	1544
35.1.1	Simple recurrent unit (SRU)	1544
35.1.2	Simple RNN and its approximation capability	1545
35.1.3	Backpropagation through time (BPTT)	1545
35.2	Recurrent unit variants	1548
35.2.1	Long short term memory (LSTM)	1548
35.2.2	Gated Recurrent Unit (GRU)	1551
35.3	Common RNN architectures	1553
35.3.1	Stacked RNN and bidirectional RNN	1553
35.3.2	Task dependent output layers	1555

- 35.3.3 Dropout 1556
- 35.3.4 LSTM with recurrent projection layer 1557
- 35.4 RNN application examples 1559
 - 35.4.1 Time series prediction 1559
 - 35.4.1.1 Simple RNN prediction 1559
 - 35.4.1.2 Deep autoregressive (DeepAR) model 1562
 - 35.4.1.3 Deep factor model 1563
 - 35.4.2 MNIST classification with sequential observation 1564
 - 35.4.3 Character-level language modeling 1565
 - 35.4.3.1 Word classification 1565
 - 35.4.3.2 Text generation 1566

vii optimal control and reinforcement learning

- 36 CLASSICAL OPTIMAL CONTROL THEORY 1571
 - 36.1 Basic problem 1572
 - 36.2 Controllability & observability 1573
 - 36.3 Dynamic programming principle 1574
 - 36.3.1 Principle of optimality 1574
 - 36.3.2 The Hamilton-Jacobi-Bellman equation (finite horizon) 1574
 - 36.3.3 The Hamilton-Jacobi-Bellman equation (infinite horizon) 1575
 - 36.4 Deterministic linear quadratic control 1578
 - 36.4.1 Linear quadratic control (finite horizon) 1578
 - 36.4.2 Linear quadratic control(infinite horizon) 1579
 - 36.5 Continuous-time stochastic optimal control 1581
 - 36.5.1 HJB equation for general nonlinear systems 1581
 - 36.5.2 Linear Gaussian quadratic system 1582
 - 36.6 Stochastic dynamic programming 1583
 - 36.6.1 Discrete-time Stochastic dynamic programming: finite horizon 1583
 - 36.6.2 Discrete-time stochastic dynamic programming: infinite horizon 1585
 - 36.6.2.1 Fundamentals 1585
 - 36.6.2.2 Convergence analysis 1586
 - 36.7 Notes on bibliography 1588
- 37 REINFORCEMENT LEARNING 1590
 - 37.1 Preliminaries 1593
 - 37.1.1 Notations 1593
 - 37.1.2 Finite state Markov decision process 1593
 - 37.1.3 Policy iteration and value iteration 1595
 - 37.1.3.1 Policy iteration 1595
 - 37.1.3.2 Value iteration 1599

37.2	Reinforcement learning theory	1603
37.2.1	Overview	1603
37.2.2	State-action Value function (Q function)	1605
37.2.3	Monte-Carlo method	1607
37.2.3.1	On-policy value estimation	1607
37.2.3.2	Off-policy value estimation	1609
37.2.3.3	MC-based reinforcement learning control	1609
37.2.4	TD(o) learning	1610
37.2.4.1	TD(o) for value estimation	1610
37.2.4.2	On-policy reinforcement learning control	1611
37.2.4.3	Off-policy reinforcement learning control	1612
37.2.5	TD(n) learning	1614
37.2.5.1	Motivation and concepts	1614
37.2.5.2	TD(n) for value estimation	1615
37.2.5.3	TD(n) for reinforcement learning control	1617
37.2.6	Standing challenges in reinforcement learning	1617
37.2.6.1	Curse of dimensionality	1617
37.2.6.2	Sample efficiency	1618
37.2.6.3	Exploration-exploitation dilemma	1618
37.2.6.4	Deadly triad	1618
37.3	Policy gradient learning	1619
37.3.1	Stochastic policy gradient fundamentals	1619
37.3.1.1	Preliminaries: derivative and expectation	1619
37.3.1.2	Theoretical framework based on finite-horizon trajectories	1620
37.3.1.3	Theoretical framework based on distributions*	1623
37.3.1.4	Estimate policy gradient and basic algorithms	1627
37.3.1.5	Bootstrap and Actor-Critic methods	1629
37.3.1.6	Common stochastic policies and their representations	1631
37.3.2	Advanced methods for policy gradient estimation	1633
37.3.2.1	Stochastic policy gradient with baseline	1633
37.3.2.2	Generalized advantage estimation	1636
37.3.2.3	Summary of stochastic gradient descent forms	1637
37.3.3	Deterministic policy gradient	1638
37.4	Algorithms zoo	1640
37.4.1	Neural Fitted Q Iteration (NFQ)	1640
37.4.2	Canonical deep Q learning	1641
37.4.3	DQN variants	1643
37.4.3.1	Overview	1643
37.4.3.2	Double Q learning	1644

37.4.3.3	Dueling network	1644
37.4.3.4	Deep Recurrent Q network (DRQN)	1645
37.4.3.5	Asynchronous Methods	1645
37.4.4	Universal value function approximator	1647
37.4.5	Deep deterministic policy gradient (DDPG) algorithm	1650
37.4.6	Twin-delayed deep deterministic policy gradient (TD3)	1652
37.4.7	Trust Region Policy Optimization (TRPO)	1655
37.4.7.1	TRPO	1655
37.4.7.2	Evaluating Hessian of KL-divergence	1657
37.4.8	Proximal Policy Optimization (PPO)	1659
37.4.9	Soft Actor-Critic(SAC)	1660
37.4.9.1	Entropy regulated reinforcement learning	1660
37.4.9.2	The SAC algorithm	1662
37.4.10	Evolution strategies	1663
37.5	Advanced training strategies	1667
37.5.1	Priority experience replay	1667
37.5.2	Hindsight experience generation	1668
37.5.3	Reverse goal generation	1669
37.5.4	Reverse goal generation on low-dimensional manifolds	1670
37.5.4.1	Key idea	1670
37.5.4.2	Example: navigation on a curved surface	1670
37.6	Notes on bibliography	1672

viii applications

38	NATURAL LANGUAGE PROCESSING I: FOUNDATION	1676
38.1	Introduction	1678
38.1.1	NLP tasks	1678
38.1.2	Challenges	1679
38.1.3	Approaches	1680
38.2	Word embeddings	1682
38.2.1	Overview	1682
38.2.2	SVD based word embeddings	1684
38.2.3	Word2Vec	1686
38.2.3.1	The model	1686
38.2.3.2	Optimization I: negative sampling	1689
38.2.3.3	Optimization II: subsampling of frequent words	1689
38.2.3.4	Visualization	1690
38.2.4	GloVe	1691
38.2.5	Subword model	1692
38.3	Language modeling	1695
38.3.1	Motivation	1695

38.3.2	<i>n</i> -gram statistical language model	1696
38.3.2.1	The baseline counting model	1696
38.3.2.2	Evaluation	1698
38.3.2.3	Smoothing technique	1698
38.3.3	Feed-forward neural language model	1699
38.3.4	Recurrent neural language model	1701
38.4	Contextualized word embeddings	1703
38.4.1	Introduction	1703
38.4.2	BERT	1705
38.4.2.1	Input embeddings	1706
38.4.2.2	Position encodings	1707
38.4.2.3	Multihead attention with marks	1709
38.4.2.4	Put it together	1710
38.4.2.5	Compared with ELMO	1711
38.4.3	Pre-training	1712
38.4.4	Downstream tasks: GLUE	1714
38.4.4.1	Single-sentence tasks	1714
38.4.4.2	Similarity and paraphrase tasks	1714
38.4.4.3	Inference tasks	1714
38.4.5	Fine-tuning and evaluation	1715
38.4.6	Other BERT family members	1715
38.4.6.1	RoBERTa	1715
38.4.6.2	ALBERT	1717
38.5	Notes on Bibliography	1719
38.5.1	Books and references	1719
38.5.2	Software	1719
39	NATURAL LANGUAGE PROCESSING II: TASKS	1722
39.1	Text classification and sentiment analysis	1724
39.1.1	Introduction	1724
39.1.2	Example tasks	1724
39.1.2.1	Sentiment analysis	1724
39.1.2.2	Document category	1725
39.1.3	Text preprocessing	1726
39.1.3.1	Basic steps	1726
39.1.3.2	Stop words removal	1726
39.1.3.3	Lemmatization and stemming	1727
39.1.4	Text classification models	1728
39.1.4.1	Bag of words baselines	1728
39.1.4.2	Fasttext text classification	1730
39.1.4.3	Word level CNN	1731
39.1.4.4	Character level CNN	1733

39.1.4.5	RNN	1734
39.2	Sequence labeling	1736
39.2.1	Sequence labeling tasks	1736
39.2.1.1	Named entity recognition	1736
39.2.1.2	Part-of-speech tagging	1738
39.2.2	HMM for POS tagging	1738
39.2.2.1	The model	1738
39.2.2.2	Decoding	1740
39.2.3	Conditional random field	1741
39.2.3.1	Formulation	1741
39.2.3.2	Feature function choices	1742
39.2.3.3	Training	1743
39.2.3.4	Inference on CRF	1743
39.2.4	Neural sequence labeling	1744
39.2.4.1	BiLSTM tagging	1744
39.2.4.2	BiLSTM-CRF tagging	1746
39.2.4.3	BERT labeling	1747
39.3	Sentence embeddings and its applications	1748
39.3.1	Introduction	1748
39.3.2	InferSent	1749
39.3.3	Sentence-BERT	1751
39.4	Sequence-to-sequence modeling	1752
39.4.1	Encoder decoder model	1752
39.4.1.1	work overflow	1754
39.4.2	Attention mechanism	1756
39.4.3	Google's Neural Machine Translation System	1758
39.5	Notes on Bibliography	1762
39.5.1	Books and references	1762
39.5.2	Software	1762
40	DEEP LEARNING FOR AUTOMATIC SPEECH RECOGNITION	1765
40.1	Speech signal characterization	1767
40.1.1	Speech signal representation	1767
40.1.2	Measurement of signal	1770
40.1.2.1	Amplitude and power	1770
40.1.2.2	Decibel	1771
40.1.2.3	Signal to noise ratio	1771
40.1.3	Signal transformation and representations	1773
40.1.3.1	Discrete Fourier Transform (DFT)	1773
40.1.3.2	Mel Filter bank features or Mel-spectrogram	1775
40.1.3.3	Mel Frequency Cepstral Coefficient (MFCC)	1778
40.2	Classical speech recognition model	1781

40.2.1	Introduction	1781
40.2.2	Mathematical formulation	1782
40.2.2.1	The Big picture	1782
40.2.2.2	Acoustic model component	1784
40.2.2.3	Lexicon model	1785
40.2.2.4	Language models	1785
40.2.3	Acoustic modeling framework	1786
40.2.3.1	Monophone system	1786
40.2.3.2	Context dependent triphone system	1787
40.2.4	Decoding	1788
40.3	Speech data augmentation	1789
40.3.1	Realistic noise and reverberation augmentation	1789
40.3.2	SpecAugment	1789
40.4	Deep learning models	1791
40.4.1	Motivation and overview	1791
40.4.2	Seq2Seq attention model	1792
40.4.2.1	Framework	1792
40.4.2.2	Listen, Attend, and Spell	1792
40.4.3	Acoustic encoder with CTC loss	1796
40.4.3.1	Fundamentals	1796
40.4.3.2	Path summation	1798
40.4.3.3	Decoding	1799
40.4.3.4	Deep Speech	1799
40.4.4	Transducer architecture	1801
41	DEEP LEARNING FOR SPEAKER RECOGNITION	1804
41.1	Introduction	1807
41.1.1	Background	1807
41.1.2	Speaker identification and verification	1810
41.1.3	Evaluation metrics	1811
41.1.3.1	Error types	1811
41.1.3.2	Detection cost function	1813
41.2	Classical methods: from GMM to i-vector	1815
41.2.1	Overview	1815
41.2.2	A simple GMM inference model	1815
41.2.3	GMM-UBM	1816
41.2.4	GMM-SVM	1817
41.2.5	Joint Factor Analysis	1818
41.2.6	i-vector method	1819
41.3	Deep learning for Speaker Recognition	1821
41.3.1	Introduction	1821
41.3.2	Architecture for embedding extractors	1824

41.3.2.1	Time-delayed neural network and x-vector	1824
41.3.2.2	LSTM and d-vector	1825
41.3.2.3	CNN	1826
41.3.3	Pooling strategies	1826
41.3.4	Contrastive learning	1828
41.3.4.1	Pairwise contrastive loss	1828
41.3.4.2	Prototypical loss and angular prototypical loss	1829
41.3.4.3	Generalized end-to-end (GE2E)	1831
41.3.5	Metric learning via classification	1832
41.3.5.1	Motivations	1832
41.3.5.2	Softmax loss	1832
41.3.5.3	Normalized Softmax loss	1834
41.3.5.4	AM-Softmax (CosFace)	1835
41.3.5.5	Other variants	1836
41.4	Speaker Diarization	1838
41.4.1	Introduction	1838
41.4.1.1	What is speaker diarization?	1838
41.4.1.2	Diarization evaluation	1838
41.4.1.3	Conventional clustering pipeline	1841
41.4.1.4	End-to-end neural diarization method	1842
41.4.2	Clustering approach	1844
41.4.2.1	Hierarchical clustering system	1844
41.4.2.2	LSTM based spectral clustering system	1845
41.4.3	Generative approach: unbounded interleaved-state RNN (UIS-RNN)	1847
41.4.3.1	Model overview	1847
41.4.3.2	Model structure	1848
41.4.3.3	Maximum likelihood estimation	1850
41.4.3.4	Decoding method	1851
41.4.4	Discriminative end-to-end neural speaker diarization	1852
41.4.4.1	Methodology overview	1852
41.4.4.2	Stacked BLSTM system	1855
41.4.4.3	Self-attentive Transformer system	1856
41.5	Notes on Bibliography	1860
41.5.1	Literature	1860
41.5.2	Dataset	1860

ix appendix

A	SUPPLEMENTAL MATHEMATICAL FACTS	1867
A.1	Basic logic for proof	1869
A.2	Some common limits	1870

A.3	Common series summation	1872
A.4	Some common spaces	1873
A.4.1	Notations on continuously differentiable functions	1873
A.5	Different modes of continuity	1875
A.5.1	continuity vs. uniform continuity	1876
A.6	Exchanges of limits	1877
A.6.1	Overall remark	1877
A.6.2	exchange limits with infinite summations	1877
A.6.3	Exchange limits with integration and differentiation	1877
A.6.4	Exchange differentiation with integration	1878
A.6.5	Exchange limit and function evaluations	1879
A.7	Useful inequalities	1880
A.7.1	Gronwall's inequality	1880
A.7.2	Inequality for norms	1880
A.7.3	Young's inequality for product	1881
A.8	Useful properties of matrix	1882
A.8.1	Matrix derivatives	1882
A.8.2	Matrix inversion lemma	1882
A.8.3	Block matrix	1883
A.8.4	Matrix trace	1884
A.8.5	Matrix elementary operator	1885
A.8.6	Matrix determinant	1887
A.9	Numerical integration	1888
A.9.1	Gaussian quadrature	1889
A.10	Vector calculus	1890
A.11	Numerical linear algebra computation complexity	1891
A.12	Distributions	1892
A.13	Common integrals	1893
A.14	Nonlinear root finding	1894
A.14.1	Bisection method	1894
A.14.2	Newton method	1894
A.14.3	Secant method	1894
A.15	Interpolation	1896
A.15.1	cubic interpolation	1896
Alphabetical Index		1898