Kernel Methods and Machine Learning

Offering a fundamental basis in kernel-based learning theory, this book covers both statistical and algebraic principles. It provides over 30 major theorems for kernel-based supervised and unsupervised learning models. The first of the theorems establishes a condition, arguably necessary and sufficient, for the kernelization of learning models. In addition, several other theorems are devoted to proving mathematical equivalence between seemingly unrelated models.

With nearly 30 closed-form and iterative algorithms, the book provides a step-by-step guide to algorithmic procedures and analyzing which factors to consider in tackling a given problem, enabling readers to improve specifically designed learning algorithms and to build models for new application paradigms such as green IT and big data learning technologies.

Numerous real-world examples and over 200 problems, several of which are MATLAB-based simulation exercises, make this an essential resource for undergraduate and graduate students in computer science, and in electrical and biomedical engineering. It is also a useful reference for researchers and practitioners in the field of machine learning. Solutions to some problems and additional resources are provided online for instructors.

S. Y. KUNG is a Professor in the Department of Electrical Engineering at Princeton University. His research areas include VLSI array/parallel processors, system modeling and identification, wireless communication, statistical signal processing, multimedia processing, sensor networks, bioinformatics, data mining, and machine learning. He is a Fellow of the IEEE.

Kernel Methods and Machine Learning

S. Y. KUNG
Princeton University

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314-321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi - 110025, India

103 Penang Road, #05-06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107024960

First published 2014

A catalogue record for this publication is available from the British Library

Library of Congress Cataloging in Publication data
Kung, S. Y. (Sun Yuan)
Kernel methods and machine learning / S.Y. Kung, Princeton University, New Jersey.
pages cm
ISBN 978-1-107-02496-0 (hardback)
1. Support vector machines. 2. Machine learning. 3. Kernel functions. I. Title.
Q325.5.K86 2014
006.3´10151252–dc23 2014002487

ISBN 978-1-107-02496-0 Hardback

Additional resources for this publication at www.cambridge.org/9781107024960

To Jaemin, Soomin, Timmy, and Katie, who have been our constant source of joy and inspiration

Contents

Preface *page* xvii

Part I Machine learning and kernel vector spaces 1

1 Fundamentals of kernel-based machine learning 3
1.1 Introduction 3
1.2 Feature representation and dimension reduction 4
1.2.1 Feature representation in vector space 6
1.2.2 Conventional similarity metric: Euclidean inner product 8
1.2.3 Feature dimension reduction 8
1.3 The learning subspace property (LSP) and "kernelization" of learning models 9
1.3.1 The LSP 9
1.3.2 Kernelization of the optimization formulation for learning models 13
1.3.3 The LSP is necessary and sufficient for kernelization 14
1.4 Unsupervised learning for cluster discovery 15
1.4.1 Characterization of similarity metrics 15
1.4.2 The LSP and kernelization of K-means learning models 16
1.4.3 The LSP and kernelization of ℓ_2 elastic nets 18
1.5 Supervised learning for linear classifiers 19
1.5.1 Learning and prediction phases 20
1.5.2 Learning models and linear system of equations 21
1.5.3 Kernelized learning models for under-determined systems 23
1.5.4 The vital role of the ℓ_2-norm for the LSP 24
1.5.5 The LSP condition of one-class SVM for outlier detection 25
1.6 Generalized inner products and kernel functions 25
1.6.1 Mahalanobis inner products 26
1.6.2 Nonlinear inner product: Mercer kernel functions 27
1.6.3 Effective implementation of kernel methods 30
1.7 Performance metrics 31
1.7.1 Accuracy and error rate 31
1.7.2 Sensitivity, specificity, and precision 32
1.7.3 The receiver operating characteristic (ROC) 33

1.8 Highlights of chapters 35
1.9 Problems 38

2 Kernel-induced vector spaces 44
2.1 Introduction 44
2.2 Mercer kernels and kernel-induced similarity metrics 45
2.2.1 Distance axioms in metric space 45
2.2.2 Mercer kernels 46
2.2.3 Construction of Mercer kernels 50
2.2.4 Shift-invariant kernel functions 50
2.3 Training-data-independent intrinsic feature vectors 50
2.3.1 Intrinsic spaces associated with kernel functions 52
2.3.2 Intrinsic-space-based learning models 56
2.4 Training-data-dependent empirical feature vectors 60
2.4.1 The LSP: from intrinsic space to empirical space 61
2.4.2 Kernelized learning models 63
2.4.3 Implementation cost comparison of two spaces 66
2.5 The kernel-trick for nonvectorial data analysis 67
2.5.1 Nonvectorial data analysis 68
2.5.2 The Mercer condition and kernel tricks 70
2.6 Summary 72
2.7 Problems 72

Part II Dimension-reduction: PCA/KPCA and feature selection 77

3 PCA and kernel PCA 79
3.1 Introduction 79
3.2 Why dimension reduction? 79
3.3 Subspace projection and PCA 81
3.3.1 Optimality criteria for subspace projection 81
3.3.2 PCA via spectral decomposition of the covariance matrix 82
3.3.3 The optimal PCA solution: the mean-square-error criterion 83
3.3.4 The optimal PCA solution: the maximum-entropy criterion 87
3.4 Numerical methods for computation of PCA 89
3.4.1 Singular value decomposition of the data matrix 90
3.4.2 Spectral decomposition of the scatter matrix 90
3.4.3 Spectral decomposition of the kernel matrix 91
3.4.4 Application studies of the subspace projection approach 94
3.5 Kernel principal component analysis (KPCA) 95
3.5.1 The intrinsic-space approach to KPCA 95
3.5.2 The kernelization of KPCA learning models 99
3.5.3 PCA versus KPCA 105
3.5.4 Center-adjusted versus unadjusted KPCAs 106
3.5.5 Spectral vector space 110

3.6 Summary 113
3.7 Problems 113

4 Feature selection 118
4.1 Introduction 118
4.2 The filtering approach to feature selection 119
4.2.1 Supervised filtering methods 120
4.2.2 Feature-weighted linear classifiers 122
4.2.3 Unsupervised filtering methods 124
4.2.4 Consecutive search methods 124
4.3 The wrapper approach to feature selection 127
4.3.1 Supervised wrapper methods 127
4.3.2 Unsupervised wrapper methods 129
4.3.3 The least absolute shrinkage and selection operator 130
4.4 Application studies of the feature selection approach 131
4.5 Summary 134
4.6 Problems 134

Part III Unsupervised learning models for cluster analysis 139

5 Unsupervised learning for cluster discovery 141
5.1 Introduction 141
5.2 The similarity metric and clustering strategy 141
5.3 K-means clustering models 144
5.3.1 K-means clustering criterion 144
5.3.2 The K-means algorithm 146
5.3.3 Monotonic convergence of K-means 148
5.3.4 The local optimum problem of K-means 151
5.3.5 The evaluation criterion for multiple trials of K-means 152
5.3.6 The optimal number of clusters 152
5.3.7 Application examples 152
5.4 Expectation-maximization (EM) learning models 153
5.4.1 EM clustering criterion 153
5.4.2 The iterative EM algorithm for basic GMM 155
5.4.3 Convergence of the EM algorithm with fixed σ 156
5.4.4 Annealing EM (AEM) 158
5.5 Self-organizing-map (SOM) learning models 159
5.5.1 Input and output spaces in the SOM 161
5.5.2 The SOM learning algorithm 162
5.5.3 The evaluation criterion for multiple-trial SOM 165
5.5.4 Applications of SOM learning models 166

5.6 Bi-clustering data analysis 169
5.6.1 Coherence models for bi-clustering 170
5.6.2 Applications of bi-clustering methods 171
5.7 Summary 173
5.8 Problems 174

6 Kernel methods for cluster analysis 178
6.1 Introduction 178
6.2 Kernel-based K-means learning models 179
6.2.1 Kernel K-means in intrinsic space 180
6.2.2 The K-means clustering criterion in terms of kernel matrix 181
6.3 Kernel K-means for nonvectorial data analysis 183
6.3.1 The similarity matrix for nonvectorial training datasets 184
6.3.2 Clustering criteria for network segmentation 185
6.3.3 The Mercer condition and convergence of kernel K-means 187
6.4 K-means learning models in kernel-induced spectral space 190
6.4.1 Discrepancy on optimal solution due to spectral truncation 191
6.4.2 Computational complexities 193
6.5 Kernelized K-means learning models 194
6.5.1 Solution invariance of spectral-shift on the kernel matrix 194
6.5.2 Kernelized K-means algorithms 195
6.5.3 A recursive algorithm modified to exploit sparsity 197
6.6 Kernel-induced SOM learning models 201
6.6.1 SOM learning models in intrinsic or spectral space 201
6.6.2 Kernelized SOM learning models 202
6.7 Neighbor-joining hierarchical cluster analysis 204
6.7.1 Divisive and agglomerative approaches 204
6.7.2 An NJ method that is based on centroid update 206
6.7.3 Kernelized hierarchical clustering algorithm 207
6.7.4 Case studies: hierarchical clustering of microarray data 212
6.8 Summary 213
6.9 Problems 215

Part IV Kernel ridge regressors and variants 219

7 Kernel-based regression and regularization analysis 221
7.1 Introduction 221
7.2 Linear least-squares-error analysis 222
7.2.1 Linear-least-MSE and least-squares-error (LSE) regressors 223
7.2.2 Ridge regression analysis 225
7.3 Kernel-based regression analysis 225
7.3.1 LSE regression analysis: intrinsic space 227
7.3.2 Kernel ridge regression analysis: intrinsic space 228

7.3.3 The learning subspace property (LSP): from intrinsic to empirical space 228
7.3.4 KRR learning models: empirical space 228
7.3.5 Comparison of KRRs in intrinsic and empirical spaces 230
7.4 Radial basis function (RBF) networks for regression analysis 230
7.4.1 RBF approximation networks 230
7.4.2 The Nadaraya–Watson regression estimator (NWRE) 232
7.4.3 Back-propagation neural networks 234
7.5 Multi-kernel regression analysis 240
7.6 Summary 244
7.7 Problems 244

8 Linear regression and discriminant analysis for supervised classification 248
8.1 Introduction 248
8.2 Characterization of supervised learning models 249
8.2.1 Binary and multiple classification 249
8.2.2 Learning, evaluation, and prediction phases 250
8.2.3 Off-line and inductive learning models 251
8.2.4 Linear and nonlinear learning models 252
8.2.5 Basic supervised learning strategies 252
8.3 Supervised learning models: over-determined formulation 253
8.3.1 Direct derivation of LSE solution 254
8.3.2 Fisher's discriminant analysis (FDA) 258
8.4 Supervised learning models: under-determined formulation 263
8.5 A regularization method for robust classification 266
8.5.1 The ridge regression approach to linear classification 266
8.5.2 Perturbational discriminant analysis (PDA): an extension of FDA 268
8.5.3 Equivalence between RR and PDA 269
8.5.4 Regularization effects of the ridge parameter ρ 270
8.6 Kernelized learning models in empirical space: linear kernels 273
8.6.1 Kernelized learning models for under-determined systems 273
8.6.2 Kernelized formulation of KRR in empirical space 276
8.6.3 Comparison of formulations in original versus empirical spaces 277
8.7 Summary 278
8.8 Problems 278

9 Kernel ridge regression for supervised classification 282
9.1 Introduction 282
9.2 Kernel-based discriminant analysis (KDA) 284
9.3 Kernel ridge regression (KRR) for supervised classification 287
9.3.1 KRR and LS-SVM models: the intrinsic-space approach 287
9.3.2 Kernelized learning models: the empirical-space approach 288

9.3.3 A proof of equivalence of two formulations 289
9.3.4 Complexities of intrinsic and empirical models 290
9.4 Perturbational discriminant analysis (PDA) 290
9.5 Robustness and the regression ratio in spectral space 292
9.5.1 The decision vector of KDA in spectral space 293
9.5.2 Resilience of the decision components of KDA classifiers 293
9.5.3 Component magnitude and component resilience 298
9.5.4 Regression ratio: KDA versus KRR 299
9.6 Application studies: KDA versus KRR 300
9.6.1 Experiments on UCI data 300
9.6.2 Experiments on microarray cancer diagnosis 301
9.6.3 Experiments on subcellular localization 302
9.7 Trimming detrimental (anti-support) vectors in KRR learning models 303
9.7.1 A pruned-KRR learning model: pruned PDA (PPDA) 304
9.7.2 Case study: ECG arrhythmia detection 306
9.8 Multi-class and multi-label supervised classification 307
9.8.1 Multi-class supervised classification 307
9.8.2 Multi-label classification 310
9.9 Supervised subspace projection methods 313
9.9.1 Successively optimized discriminant analysis (SODA) 313
9.9.2 Trace-norm optimization for subspace projection 318
9.9.3 Discriminant component analysis (DCA) 325
9.9.4 Comparisons between PCA, DCA, PC-DCA, and SODA 331
9.9.5 Kernelized DCA and SODA learning models 333
9.10 Summary 335
9.11 Problems 336

Part V Support vector machines and variants 341

10 Support vector machines 343
10.1 Introduction 343
10.2 Linear support vector machines 344
10.2.1 The optimization formulation in original vector space 345
10.2.2 The Wolfe dual optimizer in empirical space 345
10.2.3 The Karush–Kuhn–Tucker (KKT) condition 348
10.2.4 Support vectors 349
10.2.5 Comparison between separation margins of LSE and SVM 351
10.3 SVM with fuzzy separation: roles of slack variables 353
10.3.1 Optimization in original space 354
10.3.2 The learning subspace property and optimization in empirical space 354
10.3.3 Characterization of support vectors and WEC analysis 356

10.4 Kernel-induced support vector machines 358
10.4.1 Primal optimizer in intrinsic space 359
10.4.2 Dual optimizer in empirical space 359
10.4.3 Multi-class SVM learning models 361
10.4.4 SVM learning softwares 362
10.5 Application case studies 362
10.5.1 SVM for cancer data analysis 362
10.5.2 Prediction performances w.r.t. size of training datasets 364
10.5.3 KRR versus SVM: application studies 364
10.6 Empirical-space SVM for trimming of support vectors 365
10.6.1 ℓ_1-Norm SVM in empirical space 365
10.6.2 ℓ_2-Norm SVM in empirical space 365
10.6.3 Empirical learning models for vectorial and nonvectorial data analysis 367
10.6.4 Wrapper methods for empirical learning models 369
10.6.5 Fusion of filtering and wrapper methods 373
10.7 Summary 374
10.8 Problems 375

11 Support vector learning models for outlier detection 380
11.1 Introduction 380
11.2 Support vector regression (SVR) 381
11.3 Hyperplane-based one-class SVM learning models 383
11.3.1 Hyperplane-based ν-SV classifiers 383
11.3.2 Hyperplane-based one-class SVM 385
11.4 Hypersphere-based one-class SVM 389
11.5 Support vector clustering 392
11.6 Summary 393
11.7 Problems 393

12 Ridge-SVM learning models 395
12.1 Introduction 395
12.2 Roles of ρ and C on WECs of KRR and SVM 396
12.2.1 Roles of ρ and C 396
12.2.2 WECs of KDA, KRR, and SVM 397
12.3 Ridge-SVM learning models 399
12.3.1 Ridge-SVM: a unifying supervised learning model 401
12.3.2 Important special cases of Ridge-SVM models 401
12.3.3 Subset selection: KKT and the termination condition 402
12.4 Impacts of design parameters on the WEC of Ridge-SVM 404
12.4.1 Transition ramps and the number of support vectors 404
12.4.2 Effects of ρ and $C_{\min}$ on the transition ramp 404
12.4.3 The number of support vectors w.r.t. $C_{\min}$ 408

12.5 Prediction accuracy versus training time 408
12.5.1 The tuning of the parameter C 409
12.5.2 The tuning of the parameter $C_{\min}$ 409
12.5.3 The tuning of the parameter ρ 411
12.6 Application case studies 412
12.6.1 Experiments on UCI data 412
12.6.2 Experiments on microarray cancer diagnosis 413
12.6.3 Experiments on subcellular ocalization 414
12.6.4 Experiments on the ischemic stroke dataset 415
12.7 Summary 416
12.8 Problems 417

Part VI Kernel methods for green machine learning technologies 419

13 Efficient kernel methods for learning and classification 421
13.1 Introduction 421
13.2 System design considerations 423
13.2.1 Green processing technologies for local or client computing 423
13.2.2 Cloud computing platforms 423
13.2.3 Local versus centralized processing 424
13.3 Selection of cost-effective kernel functions 424
13.3.1 The intrinsic degree J 426
13.3.2 Truncated-RBF (TRBF) kernels 428
13.4 Classification complexities: empirical and intrinsic degrees 430
13.4.1 The discriminant function in the empirical representation 432
13.4.2 The discriminant function in the intrinsic representation 433
13.4.3 Tensor representation of discriminant functions 436
13.4.4 Complexity comparison of RBF and TRBF classifiers 438
13.4.5 Case study: ECG arrhythmia detection 438
13.5 Learning complexities: empirical and intrinsic degrees 439
13.5.1 Learning complexities for KRR and SVM 439
13.5.2 A scatter-matrix-based KRR algorithm 440
13.5.3 KRR learning complexity: RBF versus TRBF kernels 440
13.5.4 A learning and classification algorithms for big data size N 440
13.5.5 Case study: ECG arrhythmia detection 442
13.6 The tradeoff between complexity and prediction performance 444
13.6.1 Comparison of prediction accuracies 444
13.6.2 Prediction–complexity tradeoff analysis 446
13.7 Time-adaptive updating algorithms for KRR learning models 447
13.7.1 Time-adaptive recursive KRR algorithms 448
13.7.2 The intrinsic-space recursive KRR algorithm 449
13.7.3 A time-adaptive KRR algorithm with a forgetting factor 452

13.8 Summary 453
13.9 Problems 453

Part VII Kernel methods and statistical estimation theory 457

14 Statistical regression analysis and errors-in-variables models 459
14.1 Introduction 459
14.2 Statistical regression analysis 460
14.2.1 The minimum mean-square-error (MMSE) estimator/regressor 461
14.2.2 Linear regression analysis 462
14.3 Kernel ridge regression (KRR) 463
14.3.1 Orthonormal basis functions: single-variate cases 463
14.3.2 Orthonormal basis functions: multivariate cases 466
14.4 The perturbation-regulated regressor (PRR) for errors-in-variables models 467
14.4.1 MMSE solution for errors-in-variables models 468
14.4.2 Linear perturbation-regulated regressors 470
14.4.3 Kernel-based perturbation-regulated regressors 471
14.5 The kernel-based perturbation-regulated regressor (PRR): Gaussian cases 472
14.5.1 Orthonormal basis functions: single-variate cases 472
14.5.2 Single-variate Hermite estimators 473
14.5.3 Error–order tradeoff 475
14.5.4 Simulation results 477
14.5.5 Multivariate PRR estimators: Gaussian distribution 480
14.6 Two-projection theorems 482
14.6.1 The two-projection theorem: general case 483
14.6.2 The two-projection theorem: polynomial case 485
14.6.3 Two-projection for the PRR 486
14.6.4 Error analysis 486
14.7 Summary 487
14.8 Problems 488

15 Kernel methods for estimation, prediction, and system identification 494
15.1 Introduction 494
15.2 Kernel regressors for deterministic generation models 495
15.3 Kernel regressors for statistical generation models 500
15.3.1 The prior model and training data set 500
15.3.2 The Gauss–Markov theorem for statistical models 501
15.3.3 KRR regressors in empirical space 507
15.3.4 KRR regressors with Gaussian distribution 509
15.4 Kernel regressors for errors-in-variables (EiV) models 510
15.4.1 The Gauss–Markov theorem for EiV learning models 511
15.4.2 EiV regressors in empirical space 515
15.4.3 EiV regressors with Gaussian distribution 517
15.4.4 Finite-order EiV regressors 518

15.5 Recursive KRR learning algorithms 521
15.5.1 The recursive KRR algorithm in intrinsic space 522
15.5.2 The recursive KRR algorithm in empirical space 524
15.5.3 The recursive KRR algorithm in intrinsic space with a forgetting factor 525
15.5.4 The recursive KRR algorithm in empirical space with a forgetting factor and a finite window 527
15.6 Recursive EiV learning algorithms 529
15.6.1 Recursive EiV learning models in intrinsic space 529
15.6.2 The recursive EiV algorithm in empirical space 530
15.7 Summary 531
15.8 Problems 531

Part VIII Appendices 537

Appendix A Validation and testing of learning models 539
A.1 Cross-validation techniques 539
A.2 Hypothesis testing and significance testing 541
A.2.1 Hypothesis testing based on the likelihood ratio 542
A.2.2 Significance testing from the distribution of the null hypothesis 545
A.3 Problems 547

Appendix B *k*NN, PNN, and Bayes classifiers 549
B.1 Bayes classifiers 550
B.1.1 The GMM-based-classifier 551
B.1.2 The basic Bayes classifier 552
B.2 Classifiers with no prior learning process 554
B.2.1 *k* nearest neighbors (*k*NN) 554
B.2.2 Probabilistic neural networks (PNN) 555
B.2.3 The log-likelihood classifier (LLC) 557
B.3 Problems 559

References 561
Index 578

Preface

Machine learning is a research field involving the study of theories and technologies to adapt a system model using a training dataset, so that the learned model will be able to generalize and provide a correct classification or useful guidance even when the inputs to the system are previously unknown. Machine learning builds its foundation on linear algebra, statistical learning theory, pattern recognition, and artificial intelligence. The development of practical machine learning tools requires multi-disciplinary knowledge including matrix theory, signal processing, regression analysis, discrete mathematics, and optimization theory. It covers a broad spectrum of application domains in multimedia processing, network optimization, biomedical analysis, etc.

Since the publication of Vapnik's book entitled *The Nature of Statistical Learning Theory* (Springer-Verlag, 1995) and the introduction of the celebrated support vector machine (SVM), research on kernel-based machine learning has flourished steadily for nearly two decades. The enormous amount of research findings on unsupervised and supervised learning models, both theory and applications, should already warrant a new textbook, even without considering the fact that this fundamental field will undoubtedly continue to grow for a good while.

The book first establishes algebraic and statistical foundations for kernel-based learning methods. It then systematically develops kernel-based learning models both for unsupervised and for supervised scenarios.

- The secret of success of a machine learning system lies in finding an effective representation for the objects of interest. In a basic representation, an object is represented as a feature vector in a finite-dimensional vector space. However, in numerous machine learning applications, two different types of modified representations are often employed: one involving dimension reduction and another involving dimension expansion.

  Dimension reduction. Dimension reduction is vital for visualization because of humans' inability to see objects geometrically in high-dimensional space. Likewise, dimension reduction may become imperative because of a machine's inability to process computationally demanding data represented by an extremely huge dimensionality. Subspace projection is a main approach to dimension reduction. This book will study principal component analysis (PCA) and discriminant component analysis (DCA), two such projection methods for unsupervised and supervised learning scenarios, respectively.

Dimension expansion. In other application scenarios, the dimensionality of the original feature space may be too small, which in turn limits the design freedom of any linear methods, rendering them ineffective for classifying datasets with complex data distributions. In this case, dimension expansion offers a simple and effective solution. One of the most systematic approaches to dimension expansion is the kernel methods, which are based on polynomial or Gaussian kernels. The higher the order of the kernel functions the more expanded the new feature space.

As shown later, the kernel methods, when applied to PCA or DCA, will lead to kernel PCA and kernel DCA, respectively. Likewise, the same methods may be used to derive various kernelized learning models both for unsupervised and for supervised scenarios.

- **Unsupervised learning models.** The book presents conventional unsupervised learning models for clustering analysis. They include K-means, expectation-maximization (EM), self-organizing-map (SOM), and neighbor-joining (NJ) methods. All these unsupervised learning models can be formulated as ℓ_2-based optimizers, thus they satisfy a critical learning subspace property (LSP). This in turn assures the existence of their kernelized counterparts, i.e. kernelized learning models. The latter models are formulated in terms of pairwise similarities between two objects, as opposed to the representative feature vectors for individual objects. Hence kernelized learning models are naturally applicable to non-vectorial data analysis, such as network segmentation.
- **Supervised learning models.** The book also presents conventional supervised learning models for classfication. They include least-squares error (LSE), Fisher discriminant analysis (FDA), ridge regression (RR) and linear SVM. All these supervised learning models can be formulated as ℓ_2-based optimizers, thus they satisfy the LSP condition, which in turn leads to their respective kernelized formulations, such as kernel RR (KRR) and kernel SVM. The combination of KRR and SVM further yields a hybrid classifier, named Ridge-SVM. The Ridge-SVM is endowed with a sufficient set of design parameters to embrace existing classifiers as its special cases, including KDA, KRR, and SVM. With properly adjusted parameters, again, all these kernelized supervised learning models are naturally applicable to nonvectorial data analysis, such as subcellular protein-sequence prediction.

In the book, the presentation of these topics and their extensions will be subdivided into the following parts:

(i) Part I: Machine learning and kernel vector spaces
(ii) Part II: Dimension-reduction: PCA/KPCA and feature selection
(iii) Part III: Unsupervised learning models for cluster analysis
(iv) Part VI: Kernel ridge regressors and variants
(v) Part V: Support vector machines and variants
(vi) Part VI: Kernel methods for green machine learning technologies
(vii) Part VII: Kernel methods for statistical estimation theory
(viii) Part VIII: Appendices.

The table of contents provides a more detailed description of the scope of the book.

From the perspective of new feature representation

The study of kernel-based machine learning involves a natural extension of the linear methods into their nonlinear counterparts. This book starts by devoting much of the discussion to establishing formally the linear learning models so as to make sure that students are given an opportunity to acquire a solid grasp of the underlying linear algebra and statistical principles of the learning models. The mathematical principle of kernel methods, instead of linear methods, hinges upon replacing the conventional pairwise similarity metric by a nonlinear kernel function. This ultimately leads to the nonlinear (and more flexible) decision boundaries for pattern classification. In summary, this basic mapping approach is conceptually simple. It involves (1) mapping the original representative vectors to the (dimension-expanded) intrinsic space, resulting in a training-data-independent feature representation; and (2) applying the same linear methods to the new and higher-dimensional feature vectors to yield a kernel-based learning model, which is defined over the intrinsic space.

From the perspective of the kernel trick

If the LSP holds, the above two-step mapping procedure can ultimately lead to a kernelized learning model, defined over the "empirical space" with a training-data-dependent feature representation. In the literature, the tedious two-step re-mapping process has often been replaced by a shortcut, nicknamed the "kernel trick." Most authors present the kernel trick as an elegant and simple notion. However, as evidenced by the following two aspects, a deeper understanding will prove essential to fully appreciating the limitation/power of the kernel trick.

- **The pre-requisite of applying the kernel trick.** First of all, note that not all linear learning models are amenable to the kernel trick. Let us briefly explain the precondition for applying the kernel trick. Conceptually, machine learning methods are built upon the principle of learning from examples. Algebraically, the range of the training vectors forms a learning subspace prescribing the subspace on which the solution is most likely to fall. This leads to a formal condition named the learning subspace property (LSP). It can be shown that the kernel trick is applicable to a linear learning model if and only if the LSP holds for the model. In other words, the LSP is the pre-requisite for the kernelizability of a linear learning model.
- **The interplay between two kernel-induced representations.** Given the kernelizability, we have at our disposal two learning models, defined over two different kernel-induced vector spaces. Now let us shift our attention to the interplay between two kernel-induced representations. Even though the two models are theoretically equivalent, they could incur very different implementation costs for learning and prediction. For cost-effective system implementation, one should choose the lower-cost representation, irrespective of whether it is intrinsic or empirical. For example, if the dimensionality of the empirical space is small and manageable, an empirical-space learning model will be more appealing. However, this will not be so if the number of

training vectors is extremely large, which is the case for the "big-data" learning scenario. In this case, one must give serious consideration to the intrinsic model, whose cost can be controlled by properly adjusting the order of the kernel function.

Presentation style and coverage of the book

For an introductory textbook, it would be wise to keep the mathematics to a minimum and choose materials that are easily accessible to beginning students and practitioners. After all, one of the overriding reasons for my undertaking of this project is because the original book by Vapnik is mathematically so deep that it is accessible only to the most able researchers.

Moreover, an editor keenly reminded me of the famous cliché that "for every equation in the book the readership would be halved." To be fair, my original intention was indeed to write a mathematically much simpler textbook. The book can hardly be considered a success by this measure – having included nearly a thousand equations, thirty or so algorithms, and almost as many theorems.

From another viewpoint, however, such heavy use of equations does serve some very useful purposes.

- This book includes nearly sixty numerical examples, many with step-by-step descriptions of an algorithmic procedure. Concrete examples with numerical equations may go a long way towards clarifying the mathematical algorithm or theorem. They provide a tangible, and much less abstract, illustration of the actual procedure.
- This book contains equations specifying the bounds of computational complexities or estimates of prediction performance associated with a learning model, each of which could serve as a preliminary and quantitative guideline on the effectiveness of the learning model for specific applications.
- The book aims at demonstrating how machine learning models can be integrated into a recognition application system. Some theorems and equations in the book are devoted to establishing connections between equivalent learning models, paving a way to avoid redundant experiments on equivalent (and thus predictable) models. In short, the mathematical equivalence both improves the understanding of the models and prevents repetitive coding efforts.
- Compared with natural language or computer language (e.g. pseudocodes), the mathematics and equations provide a more concise descriptive language. With somewhat casual mathematical language, the semi-formal presentation style of this book should help beginning readers to more easily appreciate the power of the linear algebra and statistical theory behind the machine learning tools.

Comprehensiveness versus cohesiveness

Since machine learning covers a vast range of subjects, the selection of materials for this book inevitably involves a tradeoff between comprehensiveness and cohesiveness. Admittedly, the coverage of the book is far from being comprehensive. The constraint on space was certainly an important factor. On the other hand, there is already a large volume of publications on SVM and its variants. In order to save space, it was necessary

to leave out many SVM-related subjects, knowing that several excellent presentations of SVM are already available in textbook form.

What sets the book apart from others is unlikely to be its scope of coverage; rather, it may very well be the cohesive presentation and novel results.

- **Cohesive presentation.** The book aims at offering a cohesive, organized, and yet balanced presentation with natural flow between sections. This streamlined approach facilitates the presentation of key ideas in a single flow, without digression into the analytical details. Moreover, the streamlined approach also reflects a personal (and subjective) viewpoint on how to relate the loosely connected subjects.
- **Novel results.** Some significant novel results have been introduced here for the first time in textbook form. For example, under the supervised scenario, DCA for optimal subspace projection will outperform PCA, which is meant for use in unsupervised scenarios. A hybrid learning model of KRR and SVM, named Ridge-SVM, covers many existing classifiers as special cases, including KDA, KRR, and SVM. With properly adjusted parameters, it has been shown to deliver improved generalization and prediction capability. The book also establishes the theoretical foundation linking kernel methods and the rich theory in estimation, prediction, and system identification. Curiously, the presentation of these novel ideas seemed to fall naturally into appropriate places in their respective chapters.

Finally, due to its emphasis being placed on a cohesive and streamlined presentation of key ideas, the book necessarily had to forgo some otherwise important research results. I would like to take this opportunity to express my most sincere apologies and profound regret to researchers whose contributions have inadvertently been omitted here.

Readership of the book

The book was designed for senior and graduate students with a diversity of educational experiences in computer science, electrical engineering, financial engineering, applied statistics, etc. The main focus of the book aims at taking a beginning student, with some prior exposure to linear algebra, statistical theory, and convex optimization, through an integrated understanding of the underlying principles and potential applications of kernel-based learning models. In addition, the book should provide enough material for it to be used either as a textbook for classroom instruction or as a reference book for self-study.

- **As a textbook for machine learning course.** The book may be adopted for one-semester senior or graduate courses in machine learning in, say, electrical engineering and computer science departments. For example, by carefully picking some fundamental materials from Chapters 1 through 13, it should be possible to find enough material to be organized into a one-semester course that covers feature representations, and unsupervised and supervised learning models, with balanced yet rigorous treatments in statistics and linear algebra.

Just like in other textbooks, exercises are included at the end of each chapter. They should be useful for self-study and for probing into some of the more intricate aspects of the subjects treated in the text.

- **As a recommended or supplementary reference for courses on artificial intelligence.** The scope of the materials covered here is sufficiently broad to allow it to be re-structured for many other educational purposes. For example, the book may be adopted as a recommended reference for artificial intelligence and machine learning. It may also be adopted as a textbook/reference for a two-semester course. In this case, the first semester can be devoted to fundamental concepts, with the second semester covering advanced research areas such as big-data learning and kernel-based statistical estimation. For the latter area, Chapters 14 and 15 present statistical estimation techniques with errors-in-variables methods, Gauss–Markov theorems, and kernel methods for time-series analysis.
- **As a reference book for research and development.** The book is also intended for professional engineers, scientists, and system integrators who want to learn systematic ways of implementing machine learning systems. Throughout the book, application examples are provided to motivate the learning model developed. The book provides practitioners with basic mathematical knowledge so that they know how to apply off-the-shelf machine learning codes to solve new problems. In addition, efforts have been made to make the book relatively self-contained. For example, some basic matrix algebra and statistical theory are included in the book, making the book more accessible to newcomers from other fields and to those who have become rusty with some aspects of their undergraduate curriculum.

Acknowledgements

I found this writing project to be expectedly difficult at the beginning, but surprisingly enjoyable towards the end. It was truly rewarding seeing so many old and new results fall so nicely into place together. I also came to the realization that I had been so very fortunate to be surrounded by many fine people, professors, colleagues, students, and friends. The emotional parting with a seven-year-long project is somewhat offset by the pleasure of being able to finally acknowledge this unique group of people who made it possible.

I am pleased to acknowledge the generous support of a gift grant from Mitsubishi (MERL), a research grant from Motorola, multiple research grants from the Hong Kong Research Grants Council, and the DARPA Research Program on active authentication. The project was also indirectly supported by various fellowships, received by some of my collaborators, from Princeton University, the Canadian Government, and Microsoft Inc.

I was fortunate to benefit from the outstanding professional support of many fine people at Cambridge University Press (CUP), including Phil Meyler, Sarah Marsh, Elizabeth Horne, Kirsten Bot, Jessica Murphy, Dr. Steven Holt, and numerous others. I wish to thank the anonymous CUP reviewer who kindly suggested the current title of the book.

During the period of the book project, I was a Distinguished Visiting Professor at the EEE Department of the University of Hong Kong for several summers. I am grateful for the kind hospitality, warm friendship, and stimulating exchange with C. Q. Chang, Fei Mai, Y. S. Hung, and many others.

This book was an outgrowth of many years of teaching and research on neural networks, biometric authentication, and machine learning. I am grateful to the Department of Electrical Engineering of Princeton University and my fellow colleagues for having created such a scholarly environment for teaching and research. In particular, I would like to acknowledge Sarah McGovern, Stacey Weber, and Lori Baily for their cheerful spirit and generous assistance.

I am much indebted to my Ph.D. students, former and current, for their participation in building my understanding of machine learning. They include Xinying Zhang, Yunnan Wu, C. L. Myers, Ilias Tagkopoulos, Yuhui Luo, Peiyuan Wu, and Yinan Yu (Princeton University), as well as Jian Guo, Shibiao Wan, and F. Tobar (outside Princeton University). Their research studies have provided an important foundation for this book. Moreover, they have helped develop this book in various ways.

I would like to acknowledge the invaluable contributions of all of the students in my class during the past six years, undergraduate and graduate, for their invaluable contribution to examples and exercises. In particular, I would like to mention Tiffany Tong, Chun-Yi Lee, Chia-Chun Lin, Dan Li, K. H. Lee, Si Chen, Yang Yang, Clement Canonne, Pei-yuan Wu, Zhang Zhuo, Xu Chen, Pingmei Xu, Shang Shang, Rasmus Rothe, Vincent Pham, and Jintao Zhang.

I express my sincere gratitude to my visiting professors for stimulating discussions and for their proofreading of numerous versions of the previous drafts when they visited Princeton University. They are Young-Shik Moon, Shang-Hung Lai, Shaikh Fattah, Jie Lin, Wei-Kuang Lai, Xiao-Dong Gu, Yu Liu, and K. Diamantaras. I also benefited greatly from the enlightening exchanges with many external collaborators, in particular, Professors J. Morris Chang, Y. K. Chen, Y. B. Kan, T. S. Lin, Mahesan Niranjan, D. Mandic, T. McKelvery, Jin-Shiuh Taur, Yue Wang, and Juan Zhou.

There is little doubt that I must have missed some important names of people whom I would like to thank, and to whom I wish to offer my most sincere apologies and profound regret in that regard.

It is always fun and brings back fond memories recalling my Stanford years, so I must express my special appreciation of Professor Thomas Kailath, my advisor and life-time mentor, for his constant inspiration and friendship. I am proud to be closely associated with a group of outstanding scholars including Professors Patrick DeWilde, Lenart Ljung, Bernard Levy, George Verghese, and Erik Verriest, among many others. Moreover, my first exposure to machine learning was a course taught by none other than Professor R. O. Duda, which was based on his now classical book *Pattern Classification and Scene Analysis* (John Wiley, 1973).

Their mention so far does not fully acknowledge the measure of the contributions by Professor Man-Wai Mak, Mr. Peiyuan Wu, and Miss Yinan Yu. For their invaluable and indispensable roles, they could conceivably have been named as co-authors of the book.

Finally, a book project of such scale would not have been possible without strong support from my parents, my wife, and all our children. In recent years, the center of attention of my (much extended) family seems to have been focused upon our four grandchildren. It is only fitting that the book is dedicated to them.

S. Y. KUNG
Princeton