Advances in Pattern Recognition

For further volumes:
http://www.springer.com/series/4205

Shigeo Abe

Support Vector Machines for Pattern Classification

Second Edition

Springer

Prof. Dr. Shigeo Abe
Kobe University
Graduate School of Engineering
1-1 Rokkodai-cho
Nada-ku
Kobe
657-8501 Japan
abe@kobe-u.ac.jp

Series Editor
Prof. Sameer Singh, PhD
Research School of Informatics
Loughborough University
Loughborough, UK

ISSN 1617-7916
ISBN 978-1-84996-097-7 e-ISBN 978-1-84996-098-4
DOI 10.1007/978-1-84996-098-4
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2010920369

© Springer-Verlag London Limited 2005, 2010
Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.
The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.
The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Preface to the Second Edition

Since the introduction of support vector machines, we have witnessed the huge development in theory, models, and applications of what is so-called *kernel-based methods*: advancement in generalization theory, kernel classifiers and regressors and their variants, various feature selection and extraction methods, and wide variety of applications such as pattern classification and regressions in biology, medicine, chemistry, as well as computer science.

In *Support Vector Machines for Pattern Classification, Second Edition*, I try to reflect the development of kernel-based methods since 2005. In addition, I have included more intensive performance comparison of classifiers and regressors, added new references, and corrected many errors in the first edition. The major modifications of, and additions to, the first edition are as follows:

Symbols: I have changed the symbols of the mapping function to the feature space from $\mathbf{g}(\mathbf{x})$ to more commonly used $\boldsymbol{\phi}(\mathbf{x})$ and its associated kernel from $H(\mathbf{x}, \mathbf{x}')$ to $K(\mathbf{x}, \mathbf{x}')$.

1.3 Data Sets Used in the Book: I have added publicly available two-class data sets, microarray data sets, multiclass data sets, and regression data sets.

1.4 Classifier Evaluation: Evaluation criteria for classifiers and regressors are discussed.

2.3.2 Kernels: Mahalanobis kernels, graph kernels, etc., are added.

2.3.6 Empirical Feature Space: The high-dimensional feature space is treated implicitly via kernel tricks. This is an advantage and also a disadvantage because we treat the feature space without knowing its structure. The empirical feature space is equivalent to the feature space, in that it gives the same kernel value as that of the feature space. The introduction of the empirical feature space greatly enhances the interpretability and manipulability of the feature space.

2.8.4 Effect of Model Selection by Cross-Validation: In realizing high generalization ability of a support vector machine, selection of kernels and their parameter values, i.e., model selection, is very important. Here I discuss how cross-validation, which is one of the most well-used model selection methods, works to generate a support vector machine with high generalization ability.

3.4 I have deleted the section "Sophisticated Architecture" because it does not work.

4.3 Sparse Support Vector Machines: Based on the idea of the empirical feature space, sparse support vector machines, which realize smaller numbers of support vectors than those of support vector machines are discussed.

4.4 Performance Comparison of Different Classifiers: Performance of some types of support vector machines is compared using benchmark data sets.

4.8 Learning Using Privileged Information: Incorporating prior knowledge into support vector machines is very useful in improving the generalization ability. Here, one such approach proposed by Vapnik is explained.

4.9 Semi-supervised Learning: I have explained the difference between semi-supervised learning and transductive learning.

4.10 Multiple Classifier Systems: Committee machines in the first edition are renamed and new materials are added.

4.11 Multiple Kernel Learning: A weighted sum of kernels with positive weights is also a kernel and is called a *multiple kernel*. A learning method of support vector machines with multiple kernels is discussed.

5.6 Steepest Ascent Methods and Newton's Methods: Steepest ascent methods in the first edition are renamed as Newton's methods and steepest ascent methods are explained in Section 5.6.1.

5.7 Batch Training by Exact Incremental Training: A batch training method based on incremental training is added.

5.8 Active Set Training in Primal and Dual: Training methods in the primal or dual form by variable-size chunking are added.

5.9 Training of Linear Programming Support Vector Machines: Three decomposition techniques for linear programming support vector machines are discussed.

6 Kernel-Based Methods: Chapter 8 Kernel-Based Methods in the first edition is placed just after Chapter 5 Training methods and kernel discriminant analysis is added.

11.5.3 Active Set Training: Active set training discussed in Section 5.8 is extended to function approximation.

11.7 Variable Selection: Variable selection for support vector regressors is added.

Preface to the First Edition

I was shocked to see a student's report on performance comparisons between support vector machines (SVMs) and fuzzy classifiers that we had developed with our best endeavors. Classification performance of our fuzzy classifiers was comparable, but in most cases inferior, to that of support vector machines. This tendency was especially evident when the numbers of class data were small. I shifted my research efforts from developing fuzzy classifiers with high generalization ability to developing support vector machine-based classifiers.

This book focuses on the application of support vector machines to pattern classification. Specifically, we discuss the properties of support vector machines that are useful for pattern classification applications, several multiclass models, and variants of support vector machines. To clarify their applicability to real-world problems, we compare the performance of most models discussed in the book using real-world benchmark data. Readers interested in the theoretical aspect of support vector machines should refer to books such as [1–4].

Three-layer neural networks are universal classifiers in that they can classify any labeled data correctly if there are no identical data in different classes [5, 6]. In training multilayer neural network classifiers, network weights are usually corrected so that the sum-of-squares error between the network outputs and the desired outputs is minimized. But because the decision boundaries between classes acquired by training are not directly determined, classification performance for the unknown data, i.e., the generalization ability, depends on the training method. And it degrades greatly when the number of training data is small and there is no class overlap.

On the other hand, in training support vector machines the decision boundaries are determined directly from the training data so that the separating margins of decision boundaries are maximized in the high-dimensional space called *feature space*. This learning strategy, based on statistical learning theory developed by Vapnik [1, 2], minimizes the classification errors of the training data and the unknown data.

Therefore, the generalization abilities of support vector machines and other classifiers differ significantly, especially when the number of training data is small. This means that if some mechanism to maximize the margins of decision boundaries is introduced to non-SVM-type classifiers, their performance degradation will be prevented when the class overlap is scarce or nonexistent.[1]

In the original support vector machine, an n-class classification problem is converted into n two-class problems, and in the ith two-class problem we determine the optimal decision function that separates class i from the remaining classes. In classification, if one of the n decision functions classifies

[1] To improve generalization ability of a classifier, a regularization term, which controls the complexity of the classifier, is added to the objective function.

an unknown data sample into a definite class, it is classified into that class. In this formulation, if more than one decision function classify a data sample into definite classes or if no decision functions classify the data sample into a definite class, the data sample is unclassifiable.

Another problem of support vector machines is slow training. Because support vector machines are trained by solving a quadratic programming problem with the number of variables equal to the number of training data, training is slow for a large number of training data.

To resolve unclassifiable regions for multiclass support vector machines we propose fuzzy support vector machines and decision-tree-based support vector machines.

To accelerate training, in this book, we discuss two approaches: selection of important data for training support vector machines before training and training by decomposing the optimization problem into two subproblems.

To improve generalization ability of non-SVM-type classifiers, we introduce the ideas of support vector machines to the classifiers: neural network training incorporating maximizing margins and a kernel version of a fuzzy classifier with ellipsoidal regions [6, pp. 90–93, 119–139].

In Chapter 1, we discuss two types of decision functions: direct decision functions, in which the class boundary is given by the curve where the decision function vanishes, and the indirect decision function, in which the class boundary is given by the curve where two decision functions take on the same value.

In Chapter 2, we discuss the architecture of support vector machines for two-class classification problems. First we explain hard-margin support vector machines, which are used when the classification problem is linearly separable, namely, the training data of two classes are separated by a single hyperplane. Then, introducing slack variables for the training data, we extend hard-margin support vector machines so that they are applicable to inseparable problems. There are two types of support vector machines: L1 soft-margin support vector machines and L2 soft-margin support vector machines. Here, L1 and L2 denote the linear sum and the square sum of the slack variables that are added to the objective function for training. Then we investigate the characteristics of solutions extensively and survey several techniques for estimating the generalization ability of support vector machines.

In Chapter 3, we discuss some methods for multiclass problems: one-against-all support vector machines, in which each class is separated from the remaining classes; pairwise support vector machines, in which one class is separated from another class; the use of error-correcting output codes for resolving unclassifiable regions; and all-at-once support vector machines, in which decision functions for all the classes are determined at once. To resolve unclassifiable regions, in addition to error-correcting codes, we discuss fuzzy support vector machines with membership functions and decision-tree-based support vector machines. To compare several methods for multiclass prob-

lems, we show performance evaluation of these methods for the benchmark data sets.

Since support vector machines were proposed, many variants of support vector machines have been developed. In Chapter 4, we discuss some of them: least-squares support vector machines whose training results in solving a set of linear equations, linear programming support vector machines, robust support vector machines, and so on.

In Chapter 5, we discuss some training methods for support vector machines. Because we need to solve a quadratic optimization problem with the number of variables equal to the number of training data, it is impractical to solve a problem with a huge number of training data. For example, for 10,000 training data, 800 MB memory is necessary to store the Hessian matrix in double precision. Therefore, several methods have been developed to speed training. One approach reduces the number of training data by preselecting the training data. The other is to speed training by decomposing the problem into two subproblems and repeatedly solving the one subproblem while fixing the other and exchanging the variables between the two subproblems.

Optimal selection of features is important in realizing high-performance classification systems. Because support vector machines are trained so that the margins are maximized, they are said to be robust for nonoptimal features. In Chapter 7, we discuss several methods for selecting optimal features and show, using some benchmark data sets, that feature selection is important even for support vector machines. Then we discuss feature extraction that transforms input features by linear and nonlinear transformation.

Some classifiers need clustering of training data before training. But support vector machines do not require clustering because mapping into a feature space results in clustering in the input space. In Chapter 8, we discuss how we can realize support vector machine-based clustering.

One of the features of support vector machines is that by mapping the input space into the feature space, nonlinear separation of class data is realized. Thus the conventional linear models become nonlinear if the linear models are formulated in the feature space. They are usually called *kernel-based methods*. In Chapter 6, we discuss typical kernel-based methods: kernel least squares, kernel principal component analysis, and the kernel Mahalanobis distance.

The concept of maximum margins can be used for conventional classifiers to enhance generalization ability. In Chapter 9, we discuss methods for maximizing margins of multilayer neural networks and in Chapter 10 we discuss maximum-margin fuzzy classifiers with ellipsoidal regions and polyhedral regions.

Support vector machines can be applied to function approximation. In Chapter 11, we discuss how to extend support vector machines to function approximation and compare the performance of the support vector machine with that of other function approximators.

Acknowledgments

I am grateful to those who are involved in the research project, conducted at the Graduate School of Engineering, Kobe University, on neural, fuzzy, and support vector machine-based classifiers and function approximators, for their efforts in developing new methods and programs. Discussions with Dr. Seiichi Ozawa were always helpful. Special thanks are due to them and current graduate and undergraduate students: T. Inoue, K. Sakaguchi, T. Takigawa, F. Takahashi, Y. Hirokawa, T. Nishikawa, K. Kaieda, Y. Koshiba, D. Tsujinishi, Y. Miyamoto, S. Katagiri, T. Yamasaki, T. Kikuchi, K. Morikawa, Y. Kamata, M. Ashihara, Y. Torii, N. Matsui, Y. Hosokawa, T. Nagatani, K. Morikawa, T. Ishii, K. Iwamura, T. Kitamura, S. Muraoka, S. Takeuchi, Y. Tajiri, and R. Yabuwaki; and a then PhD student T. Ban.

I thank A. Ralescu for having used my draft version of the book as a graduate course text and having given me many useful comments. Thanks are also due to V. Kecman, H. Nakayama, S. Miyamoto, J. A. K. Suykens, F. Anouar, G. C. Cawley, H. Motoda, A. Inoue, F. Schwenker, N. Kasabov, B.-L. Lu, and J. T. Dearmon for their valuable discussions and useful comments.

This book includes my students' and my papers published in journals and proceedings of international conferences. I must thank many anonymous reviewers who reviewed our papers for their constructive comments and pointing out errors and missing references in improving the papers and consequently my book.

The Internet was a valuable source of information in writing the book. In preparing the second edition, I extensively used SCOPUS (https://www.scopus.com/home.url) to check the well-cited papers and tried to include those papers that are relevant to my book. Most of the papers listed in References were obtained from the Internet, from either authors' home pages or free downloadable sites such as

ESANN: http://www.dice.ucl.ac.be/esann/proceedings
/electronicproceedings.htm
JMLR: http://jmlr.csail.mit.edu/papers/

CiteSeer: http://citeseer.ist.psu.edu/
NIPS: http://books.nips.cc/

I enjoyed reading papers with innovating ideas. In understanding and including the papers into my book, they were especially helpful if they were well written in that in the abstract and the introduction their proposed methods were explained not by their characteristics but by their approaches and ideas and in the main text the ideas behind the algorithms were explained before their detailed explanation.

Kobe, October 2004, October 2009 *Shigeo Abe*

References

1. V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.
2. V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, New York, 1998.
3. R. Herbrich. *Learning Kernel Classifiers: Theory and Algorithms.* MIT Press, Cambridge, MA, 2002.
4. B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, 2002.
5. S. Young and T. Downs. CARVE—A constructive algorithm for real-valued examples. *IEEE Transactions on Neural Networks*, 9(6):1180–1190, 1998.
6. S. Abe. *Pattern Classification: Neuro-Fuzzy Methods and Their Comparison.* Springer-Verlag, London, UK, 2001.

Contents

Preface v
Acknowledgments xi
Symbols xix

1 Introduction 1
1.1 Decision Functions 2
1.1.1 Decision Functions for Two-Class Problems 2
1.1.2 Decision Functions for Multiclass Problems 4
1.2 Determination of Decision Functions 8
1.3 Data Sets Used in the Book 9
1.4 Classifier Evaluation 13
References 16

2 Two-Class Support Vector Machines 21
2.1 Hard-Margin Support Vector Machines 21
2.2 L1 Soft-Margin Support Vector Machines 28
2.3 Mapping to a High-Dimensional Space 31
2.3.1 Kernel Tricks 31
2.3.2 Kernels 33
2.3.3 Normalizing Kernels 43
2.3.4 Properties of Mapping Functions Associated with Kernels 44
2.3.5 Implicit Bias Terms 47
2.3.6 Empirical Feature Space 50
2.4 L2 Soft-Margin Support Vector Machines 56
2.5 Advantages and Disadvantages 58
2.5.1 Advantages 58
2.5.2 Disadvantages 59
2.6 Characteristics of Solutions 60
2.6.1 Hessian Matrix 60
2.6.2 Dependence of Solutions on C 62

2.6.3 Equivalence of L1 and L2 Support Vector Machines . . . 67
2.6.4 Nonunique Solutions . . . 70
2.6.5 Reducing the Number of Support Vectors . . . 78
2.6.6 Degenerate Solutions . . . 81
2.6.7 Duplicate Copies of Data . . . 83
2.6.8 Imbalanced Data . . . 85
2.6.9 Classification for the Blood Cell Data . . . 85
2.7 Class Boundaries for Different Kernels . . . 88
2.8 Developing Classifiers . . . 93
2.8.1 Model Selection . . . 93
2.8.2 Estimating Generalization Errors . . . 93
2.8.3 Sophistication of Model Selection . . . 97
2.8.4 Effect of Model Selection by Cross-Validation . . . 98
2.9 Invariance for Linear Transformation . . . 102
References . . . 106

3 Multiclass Support Vector Machines . . . 113
3.1 One-Against-All Support Vector Machines . . . 114
3.1.1 Conventional Support Vector Machines . . . 114
3.1.2 Fuzzy Support Vector Machines . . . 116
3.1.3 Equivalence of Fuzzy Support Vector Machines and Support Vector Machines with Continuous Decision Functions . . . 119
3.1.4 Decision-Tree-Based Support Vector Machines . . . 122
3.2 Pairwise Support Vector Machines . . . 127
3.2.1 Conventional Support Vector Machines . . . 127
3.2.2 Fuzzy Support Vector Machines . . . 128
3.2.3 Performance Comparison of Fuzzy Support Vector Machines . . . 129
3.2.4 Cluster-Based Support Vector Machines . . . 132
3.2.5 Decision-Tree-Based Support Vector Machines . . . 133
3.2.6 Pairwise Classification with Correcting Classifiers . . . 143
3.3 Error-Correcting Output Codes . . . 144
3.3.1 Output Coding by Error-Correcting Codes . . . 145
3.3.2 Unified Scheme for Output Coding . . . 146
3.3.3 Equivalence of ECOC with Membership Functions . . . 147
3.3.4 Performance Evaluation . . . 147
3.4 All-at-Once Support Vector Machines . . . 149
3.5 Comparisons of Architectures . . . 152
3.5.1 One-Against-All Support Vector Machines . . . 152
3.5.2 Pairwise Support Vector Machines . . . 152
3.5.3 ECOC Support Vector Machines . . . 153
3.5.4 All-at-Once Support Vector Machines . . . 153
3.5.5 Training Difficulty . . . 153
3.5.6 Training Time Comparison . . . 157
References . . . 158

4 Variants of Support Vector Machines 163
4.1 Least-Squares Support Vector Machines 163
4.1.1 Two-Class Least-Squares Support Vector Machines . . . 164
4.1.2 One-Against-All Least-Squares Support Vector Machines 166
4.1.3 Pairwise Least-Squares Support Vector Machines 168
4.1.4 All-at-Once Least-Squares Support Vector Machines . . 169
4.1.5 Performance Comparison 170
4.2 Linear Programming Support Vector Machines 174
4.2.1 Architecture 175
4.2.2 Performance Evaluation 178
4.3 Sparse Support Vector Machines 180
4.3.1 Several Approaches for Sparse Support Vector Machines 181
4.3.2 Idea 183
4.3.3 Support Vector Machines Trained in the Empirical Feature Space 184
4.3.4 Selection of Linearly Independent Data 187
4.3.5 Performance Evaluation 189
4.4 Performance Comparison of Different Classifiers 192
4.5 Robust Support Vector Machines 196
4.6 Bayesian Support Vector Machines 197
4.6.1 One-Dimensional Bayesian Decision Functions 199
4.6.2 Parallel Displacement of a Hyperplane 200
4.6.3 Normal Test 201
4.7 Incremental Training 201
4.7.1 Overview 201
4.7.2 Incremental Training Using Hyperspheres 204
4.8 Learning Using Privileged Information 213
4.9 Semi-Supervised Learning 216
4.10 Multiple Classifier Systems 217
4.11 Multiple Kernel Learning 218
4.12 Confidence Level 219
4.13 Visualization 220
References 220

5 Training Methods 227
5.1 Preselecting Support Vector Candidates 227
5.1.1 Approximation of Boundary Data 228
5.1.2 Performance Evaluation 230
5.2 Decomposition Techniques 231
5.3 KKT Conditions Revisited 234
5.4 Overview of Training Methods 239
5.5 Primal–Dual Interior-Point Methods 242

5.5.1 Primal–Dual Interior-Point Methods for Linear Programming . . . 242
5.5.2 Primal–Dual Interior-Point Methods for Quadratic Programming . . . 246
5.5.3 Performance Evaluation . . . 248
5.6 Steepest Ascent Methods and Newton's Methods . . . 252
5.6.1 Solving Quadratic Programming Problems Without Constraints . . . 252
5.6.2 Training of L1 Soft-Margin Support Vector Machines . 254
5.6.3 Sequential Minimal Optimization . . . 259
5.6.4 Training of L2 Soft-Margin Support Vector Machines . 260
5.6.5 Performance Evaluation . . . 261
5.7 Batch Training by Exact Incremental Training . . . 262
5.7.1 KKT Conditions . . . 263
5.7.2 Training by Solving a Set of Linear Equations . . . 264
5.7.3 Performance Evaluation . . . 272
5.8 Active Set Training in Primal and Dual . . . 273
5.8.1 Training Support Vector Machines in the Primal . . . 273
5.8.2 Comparison of Training Support Vector Machines in the Primal and the Dual . . . 276
5.8.3 Performance Evaluation . . . 279
5.9 Training of Linear Programming Support Vector Machines . . . 281
5.9.1 Decomposition Techniques . . . 282
5.9.2 Decomposition Techniques for Linear Programming Support Vector Machines . . . 289
5.9.3 Computer Experiments . . . 297
References . . . 299

6 Kernel-Based Methods . . . 305
6.1 Kernel Least Squares . . . 305
6.1.1 Algorithm . . . 305
6.1.2 Performance Evaluation . . . 308
6.2 Kernel Principal Component Analysis . . . 311
6.3 Kernel Mahalanobis Distance . . . 314
6.3.1 SVD-Based Kernel Mahalanobis Distance . . . 315
6.3.2 KPCA-Based Mahalanobis Distance . . . 318
6.4 Principal Component Analysis in the Empirical Feature Space . . . 319
6.5 Kernel Discriminant Analysis . . . 320
6.5.1 Kernel Discriminant Analysis for Two-Class Problems . 321
6.5.2 Linear Discriminant Analysis for Two-Class Problems in the Empirical Feature Space . . . 324
6.5.3 Kernel Discriminant Analysis for Multiclass Problems . 325
References . . . 327

7 Feature Selection and Extraction 331
7.1 Selecting an Initial Set of Features 331
7.2 Procedure for Feature Selection 332
7.3 Feature Selection Using Support Vector Machines 333
7.3.1 Backward or Forward Feature Selection 333
7.3.2 Support Vector Machine-Based Feature Selection 336
7.3.3 Feature Selection by Cross-Validation 337
7.4 Feature Extraction 339
References 340

8 Clustering 343
8.1 Domain Description 343
8.2 Extension to Clustering 349
References 351

9 Maximum-Margin Multilayer Neural Networks 353
9.1 Approach 353
9.2 Three-Layer Neural Networks 354
9.3 CARVE Algorithm 357
9.4 Determination of Hidden-Layer Hyperplanes 358
9.4.1 Rotation of Hyperplanes 359
9.4.2 Training Algorithm 362
9.5 Determination of Output-Layer Hyperplanes 363
9.6 Determination of Parameter Values 363
9.7 Performance Evaluation 364
References 365

10 Maximum-Margin Fuzzy Classifiers 367
10.1 Kernel Fuzzy Classifiers with Ellipsoidal Regions 368
10.1.1 Conventional Fuzzy Classifiers with Ellipsoidal Regions 368
10.1.2 Extension to a Feature Space 369
10.1.3 Transductive Training 370
10.1.4 Maximizing Margins 375
10.1.5 Performance Evaluation 378
10.2 Fuzzy Classifiers with Polyhedral Regions 382
10.2.1 Training Methods 383
10.2.2 Performance Evaluation 391
References 393

11 Function Approximation 395
11.1 Optimal Hyperplanes 395
11.2 L1 Soft-Margin Support Vector Regressors 399
11.3 L2 Soft-Margin Support Vector Regressors 401
11.4 Model Selection 403
11.5 Training Methods 403

11.5.1 Overview . . . 403
11.5.2 Newton's Methods . . . 405
11.5.3 Active Set Training . . . 422
11.6 Variants of Support Vector Regressors . . . 429
11.6.1 Linear Programming Support Vector Regressors . . . 430
11.6.2 ν-Support Vector Regressors . . . 431
11.6.3 Least-Squares Support Vector Regressors . . . 432
11.7 Variable Selection . . . 435
11.7.1 Overview . . . 435
11.7.2 Variable Selection by Block Deletion . . . 436
11.7.3 Performance Evaluation . . . 437
References . . . 438

A Conventional Classifiers . . . 443
A.1 Bayesian Classifiers . . . 443
A.2 Nearest-Neighbor Classifiers . . . 444
References . . . 445

B Matrices . . . 447
B.1 Matrix Properties . . . 447
B.2 Least-Squares Methods and Singular Value Decomposition . . . 449
B.3 Covariance Matrices . . . 452
References . . . 454

C Quadratic Programming . . . 455
C.1 Optimality Conditions . . . 455
C.2 Properties of Solutions . . . 456

D Positive Semidefinite Kernels and Reproducing Kernel Hilbert Space . . . 459
D.1 Positive Semidefinite Kernels . . . 459
D.2 Reproducing Kernel Hilbert Space . . . 463
References . . . 465

Index . . . 467

Symbols

We use lowercase bold letters to denote vectors and uppercase italic letters to denote matrices. The following list shows the symbols used in the book:

α_i	Lagrange multiplier for $\mathbf{x}_i$
ξ_i	slack variable associated with $\mathbf{x}_i$
A^{-1}	inverse of matrix A
$A^\top$	transpose of matrix A
B	set of bounded support vector indices
b_i	bias term of the ith hyperplane
C	margin parameter
d	degree of a polynomial kernel
$\boldsymbol{\phi}(\mathbf{x})$	mapping function from $\mathbf{x}$ to the feature space
γ	parameter for a radial basis function kernel
$K(\mathbf{x}, \mathbf{x}')$	kernel
l	dimension of the feature space
M	number of training data
m	number of input variables
n	number of classes
S	set of support vector indices
U	set of unbounded support vector indices
$\|\mathbf{x}\|$	Euclidean norm of vector $\mathbf{x}$
$\mathbf{w}_i$	coefficient vector of the ith hyperplane
X_i	set for class i training data
$\|X_i\|$	number of data in the set X_i
$\mathbf{x}_i$	ith m-dimensional training data
y_i	class label 1 or -1 for input $\mathbf{x}_i$ for pattern classification and a scalar output for function approximation