

Advances in Pattern Recognition

Advances in Pattern Recognition is a series of books which brings together current developments in all areas of this multi-disciplinary topic. It covers both theoretical and applied aspects of pattern recognition, and provides texts for students and senior researchers.

Springer also publishes a related journal, **Pattern Analysis and Applications**. For more details see: <http://link.springer.de>

The book series and journal are both edited by Professor Sameer Singh of Exeter University, UK.

Also in this series:

Principles of Visual Information Retrieval

Michael S. Lew (Ed.)

1-85233-381-2

Statistical and Neural Classifiers: An Integrated Approach to Design

Šarūnas Raudys

1-85233-297-2

Advanced Algorithmic Approaches to Medical Image Segmentation

Jasjit Suri, Kamaledin Setarehdan and Sameer Singh (Eds)

1-85233-389-8

NETLAB: Algorithms for Pattern Recognition

Ian T. Nabney

1-85233-440-1

Object Recognition: Fundamentals and Case Studies

M. Bennamoun and G.J. Mamic

1-85233-398-7

Computer Vision Beyond the Visible Spectrum

Bir Bhanu and Ioannis Pavlidis (Eds)

1-85233-604-8

Hexagonal Image Processing: A Practical Approach

Lee Middleton and Jayanthi Sivaswamy

1-85233-914-4

Shigeo Abe

Support Vector Machines for Pattern Classification

With 110 Figures

 Springer

Professor Dr Shigeo Abe
Kobe University, Kobe, Japan

Series editor

Professor Sameer Singh, PhD

Department of Computer Science, University of Exeter, Exeter, EX4 4PT, UK

British Library Cataloguing in Publication Data

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

Library of Congress Cataloging-in-Publication Data

Abe, Shigeo, 1947–

Support vector machines for pattern classification / Shigeo Abe.

p. cm.

Includes bibliographical references and index.

ISBN 1-85233-929-9 (alk. paper)

1. Text processing (Computer science) 2. Pattern recognition systems. 3. Machine learning. I. Title.

QA76.9.T48A23 2005

005.52—dc22

2005040265

Advances in Pattern Recognition ISSN 1617-7916

ISBN-10: 1-85233-929-2

Printed on acid-free paper

ISBN-13: 978-1-85233-929-6

© Springer-Verlag London Limited 2005

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 licences 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 in the United States of America (SB)

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springeronline.com

Preface

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 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 [109, 215, 256, 257].

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 [3, 279]. 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 [256, 257], 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 i th 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 an unknown datum into a definite class, it is classified into that class. In this formulation, if more than one decision function classify a datum into definite classes, or if no decision functions classify the datum into a definite class, the datum 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 [3, pp. 90–3, 119–39].

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 charac-

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

teristics 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 problems, 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 6, 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 7, 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 8, 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

We are grateful to those who are involved in the research project, conducted at the Graduate School of Science and Technology, 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 then 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, and K. Morikawa; and Ph.D. 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 H. Nakayama, S. Miyamoto, J. A. K. Suykens, F. Anouar, G. C. Cawley, H. Motoda, A. Inoue, F. Schwenker, N. Kasabov, and B.-L. Lu for their valuable discussions and useful comments.

The Internet was a valuable source of information in writing the book. Most of the papers listed in the References were obtained from the Internet, from either authors' home pages or free downloadable sites such as:

ESANN: www.dice.ucl.ac.be/esann/proceedings/electronicproceedings.htm

JMLR: www.jmlr.org/papers/

NEC Research Institute CiteSeer: citeseer.nj.nec.com/cs

NIPS: books.nips.cc/

Contents

Preface	V
Nomenclature	1
1 Introduction	3
1.1 Decision Functions	3
1.1.1 Decision Functions for Two-Class Problems	3
1.1.2 Decision Functions for Multiclass Problems	5
1.2 Determination of Decision Functions	10
1.3 Data Sets Used in the Book	11
2 Two-Class Support Vector Machines	15
2.1 Hard-Margin Support Vector Machines	15
2.2 L1 Soft-Margin Support Vector Machines	22
2.3 Mapping to a High-Dimensional Space	25
2.3.1 Kernel Tricks	25
2.3.2 Kernels	27
2.3.3 Normalizing Kernels	30
2.3.4 Properties of Mapping Functions Associated with Kernels	31
2.3.5 Implicit Bias Terms	33
2.4 L2 Soft-Margin Support Vector Machines	37
2.5 Advantages and Disadvantages	39
2.5.1 Advantages	39
2.5.2 Disadvantages	40
2.6 Characteristics of Solutions	40
2.6.1 Hessian Matrix	41
2.6.2 Dependence of Solutions on C	42
2.6.3 Equivalence of L1 and L2 Support Vector Machines	47
2.6.4 Nonunique Solutions	50
2.6.5 Reducing the Number of Support Vectors	58
2.6.6 Degenerate Solutions	61

2.6.7	Duplicate Copies of Data	63
2.6.8	Imbalanced Data	65
2.6.9	Classification for the Blood Cell Data	65
2.7	Class Boundaries for Different Kernels	70
2.8	Developing Classifiers	72
2.8.1	Model Selection	73
2.8.2	Estimating Generalization Errors	73
2.8.3	Sophistication of Model Selection	77
2.9	Invariance for Linear Transformation	77
3	Multiclass Support Vector Machines	83
3.1	One-against-All Support Vector Machines	84
3.1.1	Conventional Support Vector Machines	84
3.1.2	Fuzzy Support Vector Machines	85
3.1.3	Equivalence of Fuzzy Support Vector Machines and Support Vector Machines with Continuous Decision Functions	89
3.1.4	Decision-Tree-Based Support Vector Machines	91
3.2	Pairwise Support Vector Machines	96
3.2.1	Conventional Support Vector Machines	96
3.2.2	Fuzzy Support Vector Machines	97
3.2.3	Performance Comparison of Fuzzy Support Vector Machines	98
3.2.4	Cluster-Based Support Vector Machines	101
3.2.5	Decision-Tree-Based Support Vector Machines	102
3.2.6	Pairwise Classification with Correcting Classifiers	112
3.3	Error-Correcting Output Codes	113
3.3.1	Output Coding by Error-Correcting Codes	114
3.3.2	Unified Scheme for Output Coding	114
3.3.3	Equivalence of ECOC with Membership Functions	115
3.3.4	Performance Evaluation	116
3.4	All-at-Once Support Vector Machines	118
3.4.1	Basic Architecture	118
3.4.2	Sophisticated Architecture	120
3.5	Comparisons of Architectures	122
3.5.1	One-against-All Support Vector Machines	122
3.5.2	Pairwise Support Vector Machines	123
3.5.3	ECOC Support Vector Machines	123
3.5.4	All-at-Once Support Vector Machines	124
3.5.5	Training Difficulty	124
3.5.6	Training Time Comparison	127

4	Variants of Support Vector Machines	129
4.1	Least Squares Support Vector Machines	129
4.1.1	Two-Class Least Squares Support Vector Machines . . .	129
4.1.2	One-against-All Least Squares Support Vector Machines	132
4.1.3	Pairwise Least Squares Support Vector Machines	133
4.1.4	All-at-Once Least Squares Support Vector Machines . . .	134
4.1.5	Performance Comparison	136
4.2	Linear Programming Support Vector Machines	140
4.2.1	Architecture	140
4.2.2	Performance Evaluation	143
4.3	Incremental Training	146
4.4	Robust Support Vector Machines	149
4.5	Bayesian Support Vector Machines	149
4.5.1	One-Dimensional Bayesian Decision Functions	150
4.5.2	Parallel Displacement of a Hyperplane	151
4.5.3	Normal Test	152
4.6	Committee Machines	153
4.7	Confidence Level	153
4.8	Visualization	154
5	Training Methods	155
5.1	Preselecting Support Vector Candidates	155
5.1.1	Approximation of Boundary Data	156
5.1.2	Performance Evaluation	158
5.2	Decomposition Techniques	159
5.3	KKT Conditions Revisited	162
5.4	Overview of Training Methods	165
5.5	Primal-Dual Interior-Point Methods	167
5.5.1	Primal-Dual Interior-Point Methods for Linear Programming	167
5.5.2	Primal-Dual Interior-Point Methods for Quadratic Programming	171
5.5.3	Performance Evaluation	173
5.6	Steepest Ascent Methods	178
5.6.1	Training Algorithms	178
5.6.2	Sequential Minimal Optimization	182
5.6.3	Training of L2 Soft-Margin Support Vector Machines . .	184
5.6.4	Performance Evaluation	185
5.7	Training of Linear Programming Support Vector Machines . . .	186
5.7.1	Primal-Dual Problems	186
5.7.2	Training by Decomposition	188

6	Feature Selection and Extraction	189
6.1	Procedure for Feature Selection	189
6.2	Feature Selection Using Support Vector Machines	190
6.2.1	Backward or Forward Feature Selection	190
6.2.2	Support Vector Machine-Based Feature Selection.....	193
6.2.3	Feature Selection by Cross-Validation	194
6.3	Feature Extraction	195
7	Clustering	201
7.1	Domain Description	201
7.2	Extension to Clustering	207
8	Kernel-Based Methods	209
8.1	Kernel Least Squares	209
8.1.1	Algorithm	209
8.1.2	Performance Evaluation	212
8.2	Kernel Principal Component Analysis	215
8.3	Kernel Mahalanobis Distance	218
8.3.1	SVD-Based Kernel Mahalanobis Distance	218
8.3.2	KPCA-Based Mahalanobis Distance	221
9	Maximum-Margin Multilayer Neural Networks	223
9.1	Approach.....	223
9.2	Three-Layer Neural Networks	224
9.3	CARVE Algorithm	227
9.4	Determination of Hidden-Layer Hyperplanes	227
9.4.1	Rotation of Hyperplanes	229
9.4.2	Training Algorithm	231
9.5	Determination of Output-Layer Hyperplanes	232
9.6	Determination of Parameter Values	233
9.7	Performance Evaluation	233
9.8	Summary.....	234
10	Maximum-Margin Fuzzy Classifiers	237
10.1	Kernel Fuzzy Classifiers with Ellipsoidal Regions	238
10.1.1	Conventional Fuzzy Classifiers with Ellipsoidal Regions	238
10.1.2	Extension to a Feature Space	239
10.1.3	Transductive Training.....	240
10.1.4	Maximizing Margins	244
10.1.5	Performance Evaluation	247
10.1.6	Summary.....	252
10.2	Fuzzy Classifiers with Polyhedral Regions	253
10.2.1	Training Methods	253
10.2.2	Performance Evaluation	261

11	Function Approximation	265
11.1	Optimal Hyperplanes	265
11.2	L1 Soft-Margin Support Vector Regressors	269
11.3	L2 Soft-Margin Support Vector Regressors	271
11.4	Training Speedup	273
11.5	Steepest Ascent Methods	274
11.5.1	Subproblem Optimization	275
11.5.2	Convergence Check	277
11.6	Candidate Set Selection	278
11.6.1	Inexact KKT Conditions	278
11.6.2	Exact KKT Conditions	278
11.6.3	Selection of Violating Variables	280
11.7	Variants of Support Vector Regressors	280
11.7.1	Linear Programming Support Vector Regressors	281
11.7.2	ν -Support Vector Regressors	281
11.7.3	Least Squares Support Vector Regressors	283
11.8	Performance Evaluation	285
11.8.1	Evaluation Conditions	285
11.8.2	Effect of Working Set Size on Speedup	286
11.8.3	Comparison of L1 and L2 Support Vector Regressors	286
11.8.4	Comparison of Exact and Inexact KKT Conditions	288
11.8.5	Comparison with Other Training Methods	290
11.8.6	Performance Comparison with Other Approximation Methods	291
11.8.7	Robustness for Outliers	294
11.8.8	Summary	295
A	Conventional Classifiers	297
A.1	Bayesian Classifiers	297
A.2	Nearest Neighbor Classifiers	298
B	Matrices	301
B.1	Matrix Properties	301
B.2	Least Squares Methods and Singular Value Decomposition	303
B.3	Covariance Matrices	305
C	Quadratic Programming	309
C.1	Optimality Conditions	309
C.2	Properties of Solutions	310
D	Positive Semidefinite Kernels and Reproducing Kernel Hilbert Space	313
D.1	Positive Semidefinite Kernels	313
D.2	Reproducing Kernel Hilbert Space	317

References	319
Index	339

Nomenclature

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^T : transpose of matrix A
- B : set of bounded support vector indices
- b_i : bias term of the i th hyperplane
- C : margin parameter
- d : degree of a polynomial kernel
- $\mathbf{g}(\mathbf{x})$: mapping function from \mathbf{x} to the feature space
- γ : parameter for a radial basis function kernel
- $H(\mathbf{x}, \mathbf{x}')$: 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 i th hyperplane
- X_i : set for class i training data
- $|X_i|$: number of data in the set X_i
- \mathbf{x}_i : i th 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