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 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 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-

[1] To 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/

Kobe, October 2004 *Shigeo Abe*