

## Preface

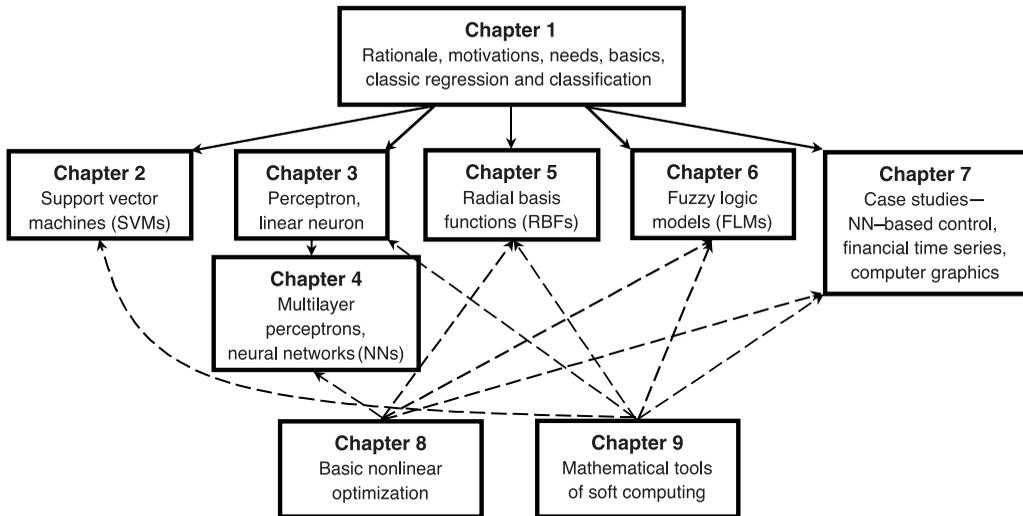
This is a book about learning from experimental data and about transferring human knowledge into analytical models. Performing such tasks belongs to soft computing. Neural networks (NNs) and support vector machines (SVMs) are the mathematical structures (models) that stand behind the idea of learning, and fuzzy logic (FL) systems are aimed at embedding structured human knowledge into workable algorithms. However, there is no clear boundary between these two modeling approaches. The notions, basic ideas, fundamental approaches, and concepts common to these two fields, as well as the differences between them, are discussed in some detail. The sources of this book are course material presented by the author in undergraduate and graduate lectures and seminars, and the research of the author and his graduate students. The text is therefore both class- and practice-tested.

The primary idea of the book is that not only is it useful to treat support vector machines, neural networks, and fuzzy logic systems as parts of a connected whole but it is in fact necessary. Thus, a systematic and unified presentation is given of these seemingly different fields—learning from experimental data and transferring human knowledge into mathematical models.

Each chapter is arranged so that the basic theory and algorithms are illustrated by practical examples and followed by a set of problems and simulation experiments. In the author's experience, this approach is the most accessible, pleasant, and useful way to master this material, which contains many new (and potentially difficult) concepts. To some extent, the problems are intended to help the reader acquire technique, but most of them serve to illustrate and develop further the basic subject matter of the chapter. The author feels that this structure is suitable both for a textbook used in a formal course and for self-study.

How should one read this book? A kind of newspaper reading, starting with the back pages, is potentially viable but not a good idea. However, there are useful sections at the back. There is an armory of mathematical weaponry and tools containing a lot of useful and necessary concepts, equations, and methods. More or less frequent trips to the back pages (chapters 8 and 9) are probably unavoidable. But in the usual way of books, one should most likely begin with this preface and continue reading to the end of chapter 1. This first chapter provides a pathway to the learning and soft computing field, and after that, readers may continue with any chapters they feel will be useful. Note, however, that chapters 3 and 4 are connected and should be read in that order. (See the figure, which represents the connections between the chapters.)

In senior undergraduate classes, the order followed was chapters 1, 3, 4, 5, and 6, and chapters 8 and 9 when needed. For graduate classes, chapter 2 on support vector machines is not omitted, and the order is regular, working directly through chapters 1–6.



There is some redundancy in this book for several reasons. The whole subject of this book is a blend of different areas. The various fields bound together here used to be separate, and today they are amalgamated in the broad area of learning and soft computing. Therefore, in order to present each particular segment of the learning and soft computing field, one must follow the approaches, tools, and terminology in each specific area. Each area was developed separately by researchers, scientists, and enthusiasts with different backgrounds, so many things were repeated. Thus, in this presentation there are some echoes but, the author believes, not too many. He agrees with the old Latin saying, *Repetio est mater studiorum*—Repetition is the mother of learning. This provides the second explanation of “redundancy” in this volume.

This book is divided into nine chapters. Chapter 1 gives examples of applications, presents the basic tools of soft computing (neural networks, support vector machines, and fuzzy logic models), reviews the classical problems of approximation of multivariate functions, and introduces the standard statistical approaches to regression and classification that are based on the knowledge of probability-density functions.

Chapter 2 presents the basics of statistical learning theory when there is no information about the probability distribution but only experimental data. The VC dimension and structural risk minimization are introduced. A description is given of the SVM learning algorithm based on quadratic programming that leads to parsimonious SVMs, that is, NNs or SVMs having a small number of hidden layer neurons. This parsimony results from sophisticated learning that matches model capacity

to data complexity. In this way, good generalization, meaning the performance of the SVM on previously unseen data, is assured.

Chapter 3 deals with two early learning units—the perceptron and the linear neuron (adaline)—as well as with single-layer networks. Five different learning algorithms for the linear activation function are presented. Despite the fact that the linear neuron appears to be very simple, it is the constitutive part of almost all models treated here and therefore is a very important processing unit. The linear neuron can be looked upon as a graphical (network) representation of classical linear regression and linear classification (discriminant analysis) schemes.

A genuine neural network (a multilayer perceptron)—one that comprises at least one hidden layer having neurons with nonlinear activation functions—is introduced in chapter 4. The error-correction type of learning, introduced for single-layer networks in chapter 3, is generalized, and the gradient-based learning method known as error backpropagation is discussed in detail here. Also shown are some of the generally accepted heuristics while training multilayer perceptrons.

Chapter 5 is concerned with regularization networks, which are better known as radial basis function (RBF) networks. The notion of ill-posed problems is discussed as well as how regularization leads to networks whose activation functions are radially symmetric. Details are provided on how to find a parsimonious radial basis network by applying the orthogonal least squares approach. Also explored is a linear programming approach to subset (basis function or support vector) selection that, similar to the QP based algorithm for SVMs training, leads to parsimonious NNs and SVMs.

Fuzzy logic modeling is the subject of chapter 6. Basic notions of fuzzy modeling are introduced—fuzzy sets, relations, compositions of fuzzy relations, fuzzy inference, and defuzzification. The union, intersection, and Cartesian product of a family of sets are described, and various properties are established. The similarity between, and sometimes even the equivalence of, RBF networks and fuzzy models is noted in detail. Finally, fuzzy additive models (FAMs) are presented as a simple yet powerful fuzzy modeling technique. FAMs are the most popular type of fuzzy models in applications today.

Chapter 7 presents three case studies that show the beauty and strength of these modeling tools. Neural networks–based control systems, financial time series prediction, and computer graphics by applying neural networks or fuzzy models are discussed at length.

Chapter 8 focuses on the most popular classical approaches to nonlinear optimization, which is the crucial part of learning from data. It also describes the novel massive search algorithms known as genetic algorithms or evolutionary computing.

Chapter 9 contains specific mathematical topics and tools that might be helpful for understanding the theoretical aspects of soft models, although these concepts and tools are not covered in great detail. It is supposed that the reader has some knowledge of probability theory, linear algebra, and vector calculus. Chapter 9 is designed only for easy reference of properties and notation.

A few words about the accompanying software are in order. All the software is based on MATLAB. All programs run in versions 5 and 6. The author designed and created the complete `approx` directory, the entire SVM toolbox for classification and regression, the multilayer perceptron routine that includes the error backpropagation learning, all first versions of core programs for RBF models for  $n$ -dimensional inputs, and some of the core fuzzy logic models. Some programs date back as far as 1992, so they may be not very elegant. However, all are effective and perform their allotted tasks as well as needed.

The author's students took an important part in creating user-friendly programs with attractive pop-up menus and boxes. At the same time, those students were from different parts of the world, and the software was developed in different countries—Yugoslavia, the United States, Germany, and New Zealand. Most of the software was developed in New Zealand. These facts are mentioned to explain why readers may find program notes and comments in English, Serbian,<sup>1</sup> and German. (However, all the basic comments are written in English.) We deliberately left these lines in various languages as nice traces of the small modern world. Without the work of these multilingual, ingenious, diligent students and colleagues, many of the programs would be less user-friendly and, consequently, less adequate for learning purposes.

As mentioned earlier, most of the core programs were developed by the author. Around them, many pieces of user-friendly software were developed as follows. In writing several versions of a program based on  $n$ -dimensional radial basis Gaussian functions Srbojub Jovanović and Lothar Niemetz took part. Oliver Wohlfarth and Kirčo Popčanovski wrote the first appealing lines of the genetic algorithm for learning in neural networks. This program was further developed by Volker Müller for static problems and by Joachim Löchner for dynamic ones. Löchner based some parts of his pop-up menus on Matthias Schanzenbach's and Herbert Vollert's program. These programs are not supplied at present. We wait to see the response to the basic routines first. Paulo Jorge Furtado Correia developed the first versions of the neural networks in C, but these had to be omitted from the MATLAB-based software. Weihong Wang wrote parts of modular networks. However, modular networks are not covered in this book, so a few software pieces on this topic are being held for a future edition. Thorsten Rommel modified and created original programs for neural networks-based control. He wrote software for recursive least squares for on-line

learning of the output layer weights. Many results in section 7.1 are obtained by applying his programs. Dieter Reusing developed a few user-friendly routines for the application of five methods on the linear neuron in section 3.2.2. Chang Bing Wang was took a crucial part in developing routines for computer graphics. The graphs and animations in section 7.3 are results of his curiosity. Faimeen Shah developed appealing pieces of software for financial time series analysis. He based parts of his program on routines from Löchner but made large steps in designing user-friendly software aimed specifically at financial time series analysis. All graphs in section 7.2 are obtained by using his routines. David Simunic and Geoffrey Taylor developed a user-friendly fuzzy logic environment as a part of their final-year project. The reader will enjoy taking the first steps in fuzzy modeling using this software with good-looking frames and windows. The author took part in mathematical solutions during the design of relational matrices. Routines for fuzzy logic control of mobile robots were developed by Wei Ming Chen and Gary Chua. Zoran Vojinović is developing applications of neural networks in company resources management, and Jonathan Robinson is using SVM for image compression. Finally, Tim Wu and Ivana Hadžić just became members of the learning and soft modeling group in the Department of Mechanical Engineering, University of Auckland. Wu's part is on chunking algorithms in SVM learning, and Hadžić is investigating the linear programming approach in designing sparse NNs or SVMs. All the software that corresponds to this book is for fair use only and free for all educational purposes. It is not for use in any kind of commercial activity.

The *Solutions Manual*, which contains the solutions to the problems in this book, has been prepared for instructors who wish to refer to the author's methods of solution. It is available from the publisher (The MIT Press, Computer Science, 5 Cambridge Center, Cambridge, MA 02142-1493, U.S.A.). The MATLAB programs needed for the simulation experiments can be retrieved at <ftp://mitpress.mit.edu/kecman/software>. This files can also be retrieved from the book's site, [www.support-vector.ws](http://www.support-vector.ws). The password is `learnscvk`.

The author is very grateful to his students, colleagues, and friends for their unceasing enthusiasm and support in the pursuit of knowledge in this tough and challenging field of learning and soft computing. A preliminary draft of this book was used in the author's senior undergraduate and graduate courses at various universities in Germany and New Zealand. The valuable feedback from the curious students who took these courses made many parts of this book easier to read. He thanks them for that. The author also warmly acknowledges the suggestions of all six unknown reviewers. He hopes that some parts of the book are more comprehensible because of their contributions.

The author thanks the University of Auckland's Research Committee for its support. As is always the case, he could have used much more money than was allotted to him, but he warmly acknowledges the tender support. The friendly atmosphere at the Department of Mechanical Engineering made the writing of this book easier than is typically the case with such an endeavor. The credit for the author's sentences being more understandable to English-speaking readers belongs partly to Emil Melnichenko, and the author thanks him. The author also thanks Douglas Sery and Deborah Cantor-Adams of The MIT Press for making the whole publishing process as smooth as possible. Their support and care in developing the manuscript, and in reviewing and editing it, are highly appreciated.

In one way or another, many people have been supportive during the author's work in the fascinating and challenging field of learning and soft computing. It is impossible to acknowledge by name all these friends, and he gives sincere thanks to them all. He is, however, particularly indebted to James H. Williams, Petar V. Kokotović, Zoran Gajić, Rolf Isermann, Peter Blessing, Heinz-Werner Röder, Stanoje Bingulac, and Dobrivoje Popović.

And, Ana was always around.