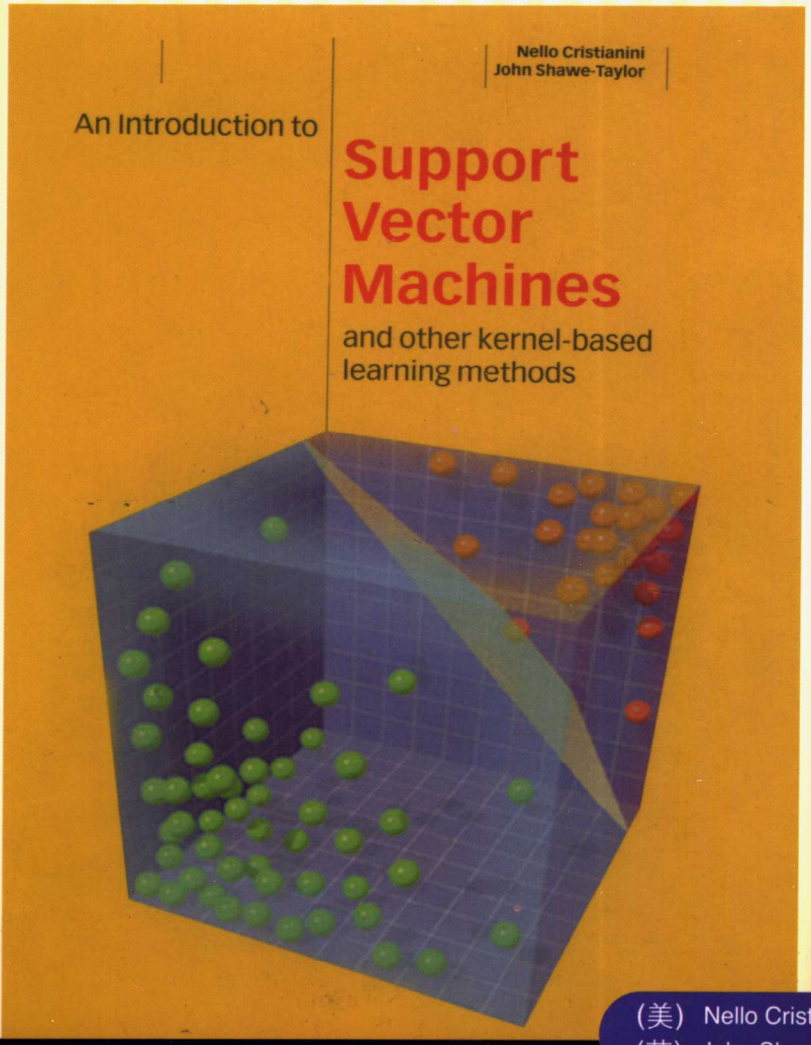
HZ BOOKS

经 典 原 版 书 库

支持向量机导论

（英文版）



(美) Nello Cristianini
(英) John Shawe-Taylor 著

机械工业出版社
China Machine Press

支持向量机导论 (英文版)

An Introduction to Support Vector Machines and Other Kernel-based Learning Methods

“本书对机器学习和数据挖掘的最重要发展领域进行了全面的介绍。书中清晰的条理、富于逻辑性的推演以及优美的文字备受初学者和专家的赞许。在线参考文献与软件是本书独一无二的和适应时代的新颖特点。我竭诚地向读者推荐此书。”

——Olvi Mangasarian，威斯康星大学

“本书介绍支持向量机的理论与实践，是一本优秀的和引人入胜的著作。书中阐明了专业人员所面临的种种关键问题及其解决方案，书后包含一个非常有用的附录，是用伪代码编写的优化算法。”

——David Haussler，加利福尼亚大学

“本书展现了一个主要由Vladimir Vapnik创立的统计学习理论的全新领域，它以正则化技术的研究成果为基础，在数学方法和应用技术两个方面都可能成为一座真正的科学金矿。在这一领域中形成并在日益增多的重大应用中得到使用的主要算法是与正则化类似的支持向量机技术。这本简明扼要的著作对支持向量机及其理论基础进行了严密而自含的介绍。这是值得数学家和工程师一读的读物。”

——Tomaso Poggio，麻省理工学院

支持向量机 (Support Vector Machine, SVM) 是建立在统计学理论最新进展基础上的新一代学习系统。本书是第一本全面介绍支持向量机的著作。支持向量机是在20世纪90年代初提出的，随之引发了对这种技术的广泛应用和深入理论分析。至今在若干实际应用 (如文本编目、手写字符识别、图像分类和生物进化链分析等) 中，支持向量机足以提供最佳的学习性能，而且在机器学习与数据挖掘中已被确立为一种标准工具。学生将会发现本书不仅对他们具有激励作用，同时也很容易理解；对于专业人员而言，本书可以引导他们轻松自如地获得为掌握理论及其应用所需的材料。本书以循序渐进的、自含的、易于接受的方式引入各种概念，而且论述严谨透彻。本书所提供的参考文献和可以下载软件的网站将会成为读者进一步学习的起点。同样，本书及相关网站将引导专业人员了解最新的文献、新应用和在线软件。

作者简介

Nello Cristianini 先后在意大利的里雅斯特大学、英国伦敦大学皇家豪勒威学院、英国布里斯投大学、美国加州大学圣克鲁兹分校学习。他是支持向量机与其他学习系统的理论与应用方面卓有成就的年青研究人员，在各种杂志和国际学术会议上发表了许多有关这一领域的论文。

John Shawe-Taylor 先后在英国剑桥大学、位于斯洛文尼亚的卢布尔雅那大学、加拿大西蒙·弗雷泽大学、英国伦敦大学帝国学院、英国伦敦大学皇家豪勒威学院学习。他发表了许多有关学习系统以及离散数学和计算机科学等领域的论文。他是英国伦敦大学皇家豪勒威学院计算科学系教授，同时还是由16所大学共同成立的欧洲合作基金的协调者，该基金是为了研究神经学习和计算学习。

上架指导：计算机/人工智能

ISBN 7-111-16789-9

9 787111 167891

封面设计：吴 刚

HZ BOOKS

华章图书

华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

投稿热线：(010) 88379604

购书热线：(010) 68995259, 68995264

读者信箱：hzjsj@hzbook.com

ISBN 7-111-16789-9/TP · 4336

定价：29.00元

经 典 原 版 书 库

支持向量机导论

（英文版）

An Introduction to Support Vector Machines and Other Kernel-based Learning Methods

(美) Nello Cristianini
(英) John Shawe-Taylor 著

机械工业出版社
China Machine Press

Nello Cristianini and John Shawe-Taylor: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods (ISBN 0-521-78019-5).

Originally published by Cambridge University Press in 2000.

This reprint edition is published with the permission of the Syndicate of the Press of the University of Cambridge, Cambridge, England.

本书版权登记号：图字：01-2005-2764

图书在版编目（CIP）数据

支持向量机导论（英文版）/（美）克里斯蒂亚尼尼（Cristianini, N.）等著. -北京：机械工业出版社，2005.7

（经典原版书库）

书名原文：An Introduction to Support Vector Machines and Other Kernel-based Learning Methods

ISBN 7-111-16789-9

I. 支 … II. 克… III. 向量计算机－算法理论－英文 IV. TP301.6

中国版本图书馆CIP数据核字（2005）第067782号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：迟振春

北京中兴印刷有限公司印刷 · 新华书店北京发行所发行

2005年7月第1版第1次印刷

787mm × 1092mm 1/16 · 12.75印张

印数：0 001-3 000册

定价：29.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzjsj@hzbook.com
联系电话：(010) 68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

Preface

In the last few years there have been very significant developments in the theoretical understanding of Support Vector Machines (SVMs) as well as algorithmic strategies for implementing them, and applications of the approach to practical problems. We believe that the topic has reached the point at which it should perhaps be viewed as its own subfield of machine learning, a subfield which promises much in both theoretical insights and practical usefulness. Despite reaching this stage of development, we were aware that no organic integrated introduction to the subject had yet been attempted. Presenting a comprehensive introduction to SVMs requires the synthesis of a surprisingly wide range of material, including dual representations, feature spaces, learning theory, optimisation theory, and algorithmics. Though active research is still being pursued in all of these areas, there are stable foundations in each that together form the basis for the SVM concept. By building from those stable foundations, this book attempts a measured and accessible introduction to the subject of Support Vector Machines.

The book is intended for machine learning students and practitioners who want a gentle but rigorous introduction to this new class of learning systems. It is organised as a textbook that can be used either as a central text for a course on SVMs, or as an additional text in a neural networks, machine learning, or pattern recognition class. Despite its organisation as a textbook, we have kept the presentation self-contained to ensure that it is suitable for the interested scientific reader not necessarily working directly in machine learning or computer science. In this way the book should give readers from other scientific disciplines a practical introduction to Support Vector Machines enabling them to apply the approach to problems from their own domain. We have attempted to provide the reader with a route map through the rigorous derivation of the material. For this reason we have only included proofs or proof sketches where they are accessible and where we feel that they enhance the understanding of the main ideas. Readers who are interested in the detailed proofs of the quoted results are referred to the original articles.

Exercises are provided at the end of the chapters, as well as pointers to relevant literature and on-line software and articles. Given the potential instability of on-line material, in some cases the book points to a dedicated website, where the relevant links will be kept updated, hence ensuring that readers can continue to

access on-line software and articles. We have always endeavoured to make clear who is responsible for the material even if the pointer to it is an indirect one. We hope that authors will not be offended by these occasional indirect pointers to their work. Each chapter finishes with a section entitled Further Reading and Advanced Topics, which fulfils two functions. First by moving all the references into this section we have kept the main text as uncluttered as possible. Again we ask for the indulgence of those who have contributed to this field when we quote their work but delay giving a reference until this section. Secondly, the section is intended to provide a starting point for readers who wish to delve further into the topics covered in that chapter. The references will also be held and kept up to date on the website. A further motivation for moving the references out of the main body of text is the fact that the field has now reached a stage of maturity which justifies our unified presentation. The two exceptions we have made to this rule are firstly for theorems which are generally known by the name of the original author such as Mercer's theorem, and secondly in Chapter 8 which describes specific experiments reported in the research literature.

The fundamental principle that guided the writing of the book is that it should be accessible to students and practitioners who would prefer to avoid complicated proofs and definitions on their way to using SVMs. We believe that by developing the material in intuitively appealing but rigorous stages, in fact SVMs appear as simple and natural systems. Where possible we first introduce concepts in a simple example, only then showing how they are used in more complex cases. The book is self-contained, with an appendix providing any necessary mathematical tools beyond basic linear algebra and probability. This makes it suitable for a very interdisciplinary audience.

Much of the material was presented in five hours of tutorials on SVMs and large margin generalisation held at the University of California at Santa Cruz during 1999, and most of the feedback received from these was incorporated into the book. Part of this book was written while Nello was visiting the University of California at Santa Cruz, a wonderful place to work thanks to both his hosts and the environment of the campus. During the writing of the book, Nello made frequent and long visits to Royal Holloway, University of London. Nello would like to thank Lynda and her family for hosting him during these visits. Together with John he would also like to thank Alex Gammerman, the technical and administrative staff, and academic colleagues of the Department of Computer Science at Royal Holloway for providing a supportive and relaxed working environment, allowing them the opportunity to concentrate on the writing.

Many people have contributed to the shape and substance of the book, both indirectly through discussions and directly through comments on early versions of the manuscript. We would like to thank Kristin Bennett, Colin Campbell, Nicolò Cesa-Bianchi, David Haussler, Ralf Herbrich, Ulrich Kockelkorn, John Platt, Tomaso Poggio, Bernhard Schölkopf, Alex Smola, Chris Watkins, Manfred Warmuth, Chris Williams, and Bob Williamson.

We would also like to thank David Tranah and Cambridge University Press for being so supportive and helpful in the processing of the book. Alessio Cristianini assisted in the establishment of the website. Kostantinos Veropoulos

helped to create the pictures for Chapter 6 which were generated using his software package at the University of Bristol. We would like to thank John Platt for providing the SMO pseudocode included in Appendix A.

Nello would like to thank the EPSRC for supporting his research and Colin Campbell for being a very understanding and helpful supervisor. John would like to thank the European Commission for support through the NeuroCOLT2 Working Group, EP27150.

Since the first edition appeared a small number of errors have been brought to our attention, and we have endeavoured to ensure that they were all corrected before reprinting. We would be grateful if anyone discovering further problems contact us through the feedback facility on the book's web page **www.support-vector.net**.

Nello Cristianini and John Shawe-Taylor
June, 2000

Notation

N	dimension of feature space
$y \in Y$	output and output space
$\mathbf{x} \in X$	input and input space
F	feature space
$\mathscr{F}$	general class of real-valued functions
$\mathscr{L}$	class of linear functions
$\langle \mathbf{x} \cdot \mathbf{z} \rangle$	inner product between $\mathbf{x}$ and $\mathbf{z}$
$\phi : X \to F$	mapping to feature space
$K(\mathbf{x}, \mathbf{z})$	kernel $\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$
$f(\mathbf{x})$	real- valued function before thresholding
n	dimension of input space
R	radius of the ball containing the data
ε-insensitive	loss function insensitive to errors less than ε
$\mathbf{w}$	weight vector
b	bias
α	dual variables or Lagrange multipliers
L	primal Lagrangian
W	dual Lagrangian
$\|\cdot\|_p$	p-norm
ln	natural logarithm
e	base of the natural logarithm
log	logarithm to the base 2
$\mathbf{x}'$, $\mathbf{X}'$	transpose of vector, matrix
$\mathbb{N}$, $\mathbb{R}$	natural, real numbers
S	training sample
ℓ	training set size
η	learning rate
ε	error probability
δ	confidence
γ	margin
ξ	slack variables
d	VC dimension

Contents

1

The Learning Methodology

The construction of machines capable of learning from experience has for a long time been the object of both philosophical and technical debate. The technical aspect of the debate has received an enormous impetus from the advent of electronic computers. They have demonstrated that machines can display a significant level of learning ability, though the boundaries of this ability are far from being clearly defined.

The availability of reliable learning systems is of strategic importance, as there are many tasks that cannot be solved by classical programming techniques, since no mathematical model of the problem is available. So for example it is not known how to write a computer program to perform hand-written character recognition, though there are plenty of examples available. It is therefore natural to ask if a computer could be trained to recognise the letter 'A' from examples – after all this is the way humans learn to read. We will refer to this approach to problem solving as the learning methodology

The same reasoning applies to the problem of finding genes in a DNA sequence, filtering email, detecting or recognising objects in machine vision, and so on. Solving each of these problems has the potential to revolutionise some aspect of our life, and for each of them machine learning algorithms could provide the key to its solution.

In this chapter we will introduce the important components of the learning methodology, give an overview of the different kinds of learning and discuss why this approach has such a strategic importance. After the framework of the learning methodology has been introduced, the chapter ends with a roadmap for the rest of the book, anticipating the key themes, and indicating why Support Vector Machines meet many of the challenges confronting machine learning systems. As this roadmap will descibe the role of the different chapters, we urge our readers to refer to it before delving further into the book.

1.1 Supervised Learning

When computers are applied to solve a practical problem it is usually the case that the method of deriving the required output from a set of inputs can be described explicitly. The task of the system designer and eventually the programmer implementing the specifications will be to translate that method

into a sequence of instructions which the computer will follow to achieve the desired effect.

As computers are applied to solve more complex problems, however, situations can arise in which there is no known method for computing the desired output from a set of inputs, or where that computation may be very expensive. Examples of this type of situation might be modelling a complex chemical reaction, where the precise interactions of the different reactants are not known, or classification of protein types based on the DNA sequence from which they are generated, or the classification of credit applications into those who will default and those who will repay the loan.

These tasks cannot be solved by a traditional programming approach since the system designer cannot precisely specify the method by which the correct output can be computed from the input data. An alternative strategy for solving this type of problem is for the computer to attempt to learn the input/output functionality from examples, in the same way that children learn which are sports cars simply by being told which of a large number of cars are sporty rather than by being given a precise specification of sportiness. The approach of using examples to synthesise programs is known as the *learning methodology*, and in the particular case when the examples are input/output pairs it is called *supervised learning*. The examples of input/output functionality are referred to as the *training data*.

The input/output pairings typically reflect a functional relationship mapping inputs to outputs, though this is not always the case as for example when the outputs are corrupted by noise. When an underlying function from inputs to outputs exists it is referred to as the *target function*. The estimate of the target function which is learnt or output by the learning algorithm is known as the *solution* of the learning problem. In the case of classification this function is sometimes referred to as the *decision function*. The solution is chosen from a set of candidate functions which map from the input space to the output domain. Usually we will choose a particular set or class of candidate functions known as *hypotheses* before we begin trying to learn the correct function. For example, so-called *decision trees* are hypotheses created by constructing a binary tree with simple decision functions at the internal nodes and output values at the leaves. Hence, we can view the choice of the set of hypotheses (or *hypothesis space*) as one of the key ingredients of the learning strategy. The algorithm which takes the training data as input and selects a hypothesis from the hypothesis space is the second important ingredient. It is referred to as the *learning algorithm*.

In the case of learning to distinguish sports cars the output is a simple yes/no tag which we can think of as a binary output value. For the problem of recognising protein types, the output value will be one of a finite number of categories, while the output values when modelling a chemical reaction might be the concentrations of the reactants given as real values. A learning problem with binary outputs is referred to as a *binary classification* problem, one with a finite number of categories as *multi-class classification*, while for real-valued outputs the problem becomes known as *regression*. This book will consider all of these

types of learning, though binary classification is always considered first as it is often the simplest case.

There are other types of learning that will not be considered in this book. For example *unsupervised learning* considers the case where there are no output values and the learning task is to gain some understanding of the process that generated the data. This type of learning includes density estimation, learning the support of a distribution, clustering, and so on. There are also models of learning which consider more complex interactions between a learner and their environment. Perhaps the simplest case is when the learner is allowed to query the environment about the output associated with a particular input. The study of how this affects the learner's ability to learn different tasks is known as *query learning*. Further complexities of interaction are considered in *reinforcement learning*, where the learner has a range of actions at their disposal which they can take to attempt to move towards states where they can expect high rewards. The learning methodology can play a part in reinforcement learning if we treat the optimal action as the output of a function of the current state of the learner. There are, however, significant complications since the quality of the output can only be assessed indirectly as the consequences of an action become clear.

Another type of variation in learning models is the way in which the training data are generated and how they are presented to the learner. For example, there is a distinction made between *batch* learning in which all the data are given to the learner at the start of learning, and *on-line* learning in which the learner receives one example at a time, and gives their estimate of the output, before receiving the correct value. In on-line learning they update their current hypothesis in response to each new example and the quality of learning is assessed by the total number of mistakes made during learning.

The subject of this book is a family of techniques for learning to perform input/output mappings from labelled examples for the most part in the batch setting, that is for applying the supervised learning methodology from batch training data.

1.2 Learning and Generalisation

We discussed how the quality of an on-line learning algorithm can be assessed in terms of the number of mistakes it makes during the training phase. It is not immediately clear, however, how we can assess the quality of a hypothesis generated during batch learning. Early machine learning algorithms aimed to learn representations of simple symbolic functions that could be understood and verified by experts. Hence, the goal of learning in this paradigm was to output a hypothesis that performed the correct classification of the training data and early learning algorithms were designed to find such an accurate fit to the data. Such a hypothesis is said to be *consistent*. There are two problems with the goal of generating a verifiable consistent hypothesis.

The first is that the function we are trying to learn may not have a simple representation and hence may not be easily verified in this way. An example