Learning with kernel machine architectures

by

Theodoros Evgeniou

B.S. Massachusetts Institute of Technology (1995)
M.Eng. Massachusetts Institute of Technology (1996)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2000

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
April 27, 2000

Certified by . . . .
Tomaso Poggio
Uncas and Helen Whitaker Professor of Brain and Cognitive Sciences
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Learning with kernel machine architectures

by

Theodoros Evgeniou

Submitted to the Department of Electrical Engineering and Computer Science
on April 27, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis studies the problem of supervised learning using a family of machines, namely kernel learning machines. A number of standard learning methods belong to this family, such as Regularization Networks (RN) and Support Vector Machines (SVM). The thesis presents a theoretical justification of these machines within a unified framework based on the statistical learning theory of Vapnik. The generalization performance of RN and SVM is studied within this framework, and bounds on the generalization error of these machines are proved.

In the second part, the thesis goes beyond standard one-layer learning machines, and probes into the problem of learning using hierarchical learning schemes. In particular it investigates the question: what happens when instead of training one machine using the available examples we train many of them, each in a different way, and then combine the machines? Two types of ensembles are defined: voting combinations and adaptive combinations. The statistical properties of these hierarchical learning schemes are investigated both theoretically and experimentally: bounds on their generalization performance are proved, and experiments characterizing their behavior are shown.

Finally, the last part of the thesis discusses the problem of choosing data representations for learning. It is an experimental part that uses the particular problem of object detection in images as a framework to discuss a number of issues that arise when kernel machines are used in practice.

Thesis Supervisor: Tomaso Poggio
Title: Uncas and Helen Whitaker Professor of Brain and Cognitive Sciences

Acknowledgments

First of all I would like to express my gratitude to my advisor Professor Tommy Poggio for guiding me throughout my graduate studies. The discussions with Tommy were to a large extent the source of the ideas found in this thesis, and were always important lessons about science and life. I would also like to thank Prof. Tommi Jaakkola for the many useful discussions we had, and Prof. Bob Berwick for co-supervising this thesis.

There is one more unofficial supervisor for this thesis: Dr. Massimiliano Pontil. I thank him both for the close collaboration on this thesis, as well as for the pleasant atmosphere he created all this time. There are many people to thank for the academic as well as friendly support. I am grateful to Dr. Vladimir Vapnik for the very interesting discussions and useful insights he offered for this work. To Tony Ezzat, who in fact introduced me to CBCL in 1995, for being a very supportive officemate. To Federico Girosi for the initial guiding in the field of machine learning, Vinay Kumar, the thoughtful visitor of our office, Sayan Mukherjee, a tough support vector, Constantine Papageorgiou, the bulky provider of ideas and fun, Luis Perez-Breva for his very enthusiastic work, and Prof. Alessandro Verri for the many useful discussions and directions. I would like to thank all other CBCLites for providing the environment for this research: Nicholas Chan, Gadi Geiger, Bernd Heisele, Martin Giese, Chikahito Nakajima, Max Reisenhuber, Ryan Rifkin, Christian Shelton, and particularly Marypat Fitzgerald for her support for everything needed in CBCL.

The thesis, like any other achievement, would have been terribly boring if it had not been written among friends. This list of people is large, and mentioning them entails the risk of forgetting some who I hope do not feel left out - they are not. The thesis was written during the breaks between parties and fun with my roommates Stas Jarecki and Martin Szummer, my friends Serafim Batzoglou, Mark Choudhari, Naved Khan and his family, Angelita Mireles, Agha Mirza, Dave Pelly, Andreas Argiriou and Yiannis Voyatzis with whom I first came to MIT in 1991, and many others, and it wouldn't have finished on time without the life support of Kimi Yagi.

This work wouldn't have started without the life-long love and encouragement of my family: my mother, Haido Hartaba, my father, Konstantinos Evgeniou, and my only sister, Konstantina Evgeniou. The thesis is dedicated to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Learning from examples is at the cornerstone of analyzing complex data sets and of developing intelligent systems. Finding patterns in sequences of numbers, developing "theories" from few examples, designing models for forecasting and decision making, datamining, are some of the areas for which what is called learning from examples is at the core to such a degree that it is often defined as some of them.

With the recent explosion of the internet, more and more complex data need to be analyzed: from web traffic statistics and e-commerce data, to digital collections of images, video, music, and of course text. Intelligent navigation through this data space and extraction of interesting information from this large collection of bits is increasingly needed. At the same time, the development of more and more complex systems such as intelligent agents and robots with advanced language, vision, and decision making modules, is typically approached from the point of view that such systems need to have the ability to learn by themselves through "experience" to perform their various difficult tasks. An ultimate dream in this direction is to achieve a general purpose learning engine to allow machines to learn from experience (and prior knowledge) – somewhat like people do – without the need of extensive programming for each specific task.

1.1.1 Complex data analysis

To better understand the significance of learning for analyzing data, let's consider two simple examples:

- Given the following two sequences of numbers:
  A: `1 3 5 7 11 13 17 19 23 ...`
  B: `2 4 6 8 10 12 14 16 18 ...`
  decide whether number 53 belongs to sequence **A** or **B**.

- Given the following sequence of numbers:
  `200 250 300 190 240 290 180 230 280 170 ...`
  predict what are the next three elements of the sequence.

Analyzing the available data in order to solve problems of this type seems easy in these examples (at first glance only): the analysis can be done by using our (hardly understood) abilities to find patterns, develop theories, and reject possible explanations from experience. In the framework of this thesis, all these abilities can be summarized as the ability to learn from examples: learning, in this framework, is a type of data – the examples – analysis process.

Of course very often the data to be analyzed are not as simple (looking) as the ones in the examples above. In a large-scale scientific experiment or in a collection of historical data of a corporation, many more numbers – and often noisy – are available, and questions similar to the ones above still need to be answered. It is a challenge to develop systems that can analyze such large datasets and develop theories from them that have predictive power: this is the goal of learning theory in the context of complex data analysis.

1.1.2 Building intelligent systems

It is a common belief that learning is at the core of intelligence. Consequently, our attempts to develop intelligent systems are highly constrained by our ability to build systems that can learn. An example that will also be further studied in this thesis can clarify this idea.

Consider the problem of developing a system that from any collection of images can automatically select only those of a particular type (i.e. images containing a face, as considered in chapter 5). One possible strategy can be for the system to memorize all images containing faces, and then given a new collection of images simply select the ones that exactly match with one of the images it has memorized. Unfortunately such an approach would require that all possible images containing faces are stored in memory: clearly this is impractical, and furthermore such a brute force memory-based system would hardly be considered by anyone as "intelligent" (granted the distinction between a "super-memory" system and an intelligent one is a vague one). Alternatively, if such a system were to be considered intelligent, it should be able to "figure out" how a face looks only from a few example images without having to rely on seeing and memorizing all possible images. We would say that a system learned how a face looks from the few available examples, if given a new image (that it has not seen before) it can successfully (with high probability) decide whether the image contains a face or not. In this example it is implied that the smarter the system is the fewer the examples it needs to learn the task. It is also implied that successful learning means low probability of making an error when new images are given.

Clearly such a process (learning) is fundamental for many other similar problems: detecting verbs in sentences, reading/recognizing characters, detecting sounds and music patterns, understanding when a particular internet surfer would actually read a news page etc. Of course very often the learning process can be a trivial one, for example in the case that the aforementioned memory-based approach is practical (i.e. if only few possible images of faces existed). It is important to notice that this can be the case even for complicated-looking tasks. For example, even for the case of face detection discussed above, a system may be intelligent enough to automatically decide upon only a few features that describe a face, and then instead of memorizing all possible images of faces it could memorize only all possible configurations of these features (which can be much less than the possible images of faces). This probes into the issues of finding appropriate representations for learning: intelligence may lie behind the choice of features. The issue of feature selection and learning is a difficult one and very likely not to be ever fully understood.

Finally notice that building intelligent systems and analyzing "complex" data are tightly related: an intelligent face detection system learns through analyzing the (seemingly) complex image data. It is often more of a conceptual distinction the one of "complex data analysis" and "intelligent system development": learning can be seen as a common underlying process.

1.2 Learning as a mathematical problem

We consider learning problems of the type discussed above: the goal is to use available examples of a particular input/output relation (i.e. input is an image, output is a yes/no answer to the question whether there is a face in the image) in order to develop a system that can learn this relation and decide outputs corresponding to future inputs. This is so called *supervised learning* or *learning from examples*, the type of learning considered in this thesis.

We view the problem of learning from examples as that of generating a hypothesis ("theory") from the available examples - the *training data* - that can be subsequently used for tasks such as prediction, pattern recognition, "event" detection, decision making etc. More formally, the data can be seen as points in a vector space, and the "theory" as a function in that space (figure 1-1). Learning from examples can then be regarded as the problem of approximating the desired multivariate function (finding a hypothesis - function F in figure 1-1) from the sparse available data.

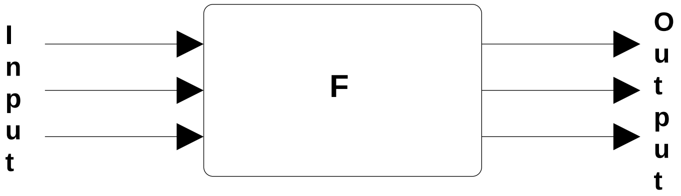

Figure 1-1: A mathematical formulation of learning an input/output relation from examples.

There are many ways one could proceed in searching for a theory-function. For example a simple solution, as discussed above, could be to always consider the function that takes everywhere a fixed value, say zero, apart from the training points where it takes the values given at those points. This would be a simple memory based approach. *A general strategy can be to always consider a fixed set of candidate functions, and then choose the best one according to the data and a desired criterion.* The choices of the set of candidate theories and of the criterion give plenty of room for a number of various learning methods. This thesis studies learning methods of this second type.

Whatever the approach one takes, the main goal is the following: how to best use the available training examples to generate a theory that can best work when given new data – has the best predictive power. This was also the key issue in the examples discussed above. This thesis studies the problem of learning within a well formulated mathematical framework, namely statistical learning theory, within which questions of this type can be answered. Theoretical foundations as well as practical issues are discussed.

1.2.1 Regression and Classification

Depending on the type of the output values of the relation learned (see Figure 1-1), we distinguish between two types of supervised learning:

- **Regression:** The outputs are real-valued, and therefore the problem is that of approximating a real-valued function from the examples.

- **Classification:** (or *pattern recognition*) The outputs take only a few possible values (finite number). In this case the problem is to discriminate between a number of types/categories of inputs.

We discuss both types of learning machines simultaneously, and discriminate between them only when it is necessary.

1.3 Outline of the thesis

The Statistical Learning Theory framework

The first part, chapter 2, reviews the basic theoretical framework of the thesis, particularly of chapters 3 and 4, namely Statistical Learning Theory (SLT) [Vapnik, 1982, Vapnik, 1995, Vapnik, 1998]. The theory has been developed to analyze learning methods of the following form:

Learning in Statistical Learning Theory

Given set of example pairs of an input-output relation, the problem of learning is approached in two basic steps:

1. A set of candidate functions (theories) that can be used to characterize the input-output relation is defined: we call this a *hypothesis space.*

2. Within the chosen hypothesis space, the function that best describes the example data according to a given criterion is found. The criterion used is *the minimization of errors over the example data made by the function.*

More formally, in the second step the theory uses the principle of *empirical risk minimization* which, as the name suggests, is the following:

- Define a loss function V that measures the error made by a function f on an input-output pair. If $\mathbf{x}$ is the input and y the actual output, then V is of the form $V(y, f(\mathbf{x}))$ and measures the error made when $f(\mathbf{x})$ is predicted while y is the output.

- Find the function in the hypothesis space that minimizes the empirical error on the example input-output data. If $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ is the set of ℓ available examples ($\mathbf{x}_i$ is the input and y_i the corresponding output), and $\mathcal{F}$ is the hypothesis space considered, then according to the empirical risk minimization principle the solution to the learning problem is the function $f \in \mathcal{F}$ that minimizes the *empirical error*:
$$\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i))$$

In the general form, the theory suggests that empirical risk minimization is performed iteratively in a number of hypothesis spaces, and that eventually the "best" solution among the ones found in each hypothesis space is chosen. The theory gives conditions under which this approach leads to solutions that can be reliably used to find the output of the relation corresponding to a new (i.e. future) input. Appropriate measure of "reliability" is also defined: it is the *expected error* the solution makes on a new input. Therefore the theory answers questions of the following type:

- How large is the expected error of the solution found using the approach above?

- How different is the empirical error of the solution from the expected one?

- How fast does the expected error decrease as the number of examples increases?

In the first part of the thesis, chapter 2, this framework is formally presented. Moreover, a technical extension of the standard SLT of Vapnik, based on work in the PAC learning community [Kearns and Shapire, 1994, Alon *et al.*, 1993, Valiant, 1984], is presented. This extended SLT will be used in chapter 3 to theoretically justify and analyze a class of learning machines, namely kernel learning machines.

Learning using Kernel Machines

A number of machines can be developed in the aforementioned framework. Among others, two are the key choices to be made when designing a learning machine:

1. Choose the loss function V.

2. Choose the set of possible functions, hypothesis space $\mathcal{F}$.

This thesis concentrates on so called *kernel machines*. These are learning machines for which the hypothesis space is a subspace of a Reproducing Kernel Hilbert Space $\mathcal{F}$ with kernel K (hence the name kernel machines) [Wahba, 1990, Aronszajn, 1950, Wahba, 1980, Wahba, 1985] - see chapter 3. Learning using these machines can be seen as a variational problem of finding the function f that minimizes the functional

$$\min_{f \in \mathcal{F}} H[f] = \frac{1}{l} \sum_{i=1}^{l} V(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_K^2 \tag{1.1}$$

where $\|f\|_K^2$ is a norm in the Reproducing Kernel Hilbert Space $\mathcal{F}$ defined by the positive definite function K, λ is a parameter often called the regularization parameter [Wahba, 1990, Powell, 1992, Poggio and Girosi, 1990, Girosi *et al.*, 1995, Ivanov, 1976], $V(y, f(\mathbf{x}))$ is, as mentioned above, the loss function measuring the error made when function f outputs $f(\mathbf{x})$ given input $\mathbf{x}$ while the actual output is y, and $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ represent the ℓ given example data - the training data.

Within this family of machines the thesis focuses on two particular kernel machines: standard regularization networks (RN) [Tikhonov and Arsenin, 1977, Ivanov, 1976, Girosi *et al.*, 1995, Wahba, 1990] and Support Vector Machines [Vapnik, 1998, Cortes and Vapnik, 1995] for both regression (SVMR) and classification (SVMC). These are kernel machines for the following choices of the loss function V:

- Standard (L_2) Regularization Networks (RN)

$$V(y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2 \tag{1.2}$$

- Support Vector Machines Regression (SVMR)

$$V(y_i, f(\mathbf{x}_i)) = |y_i - f(\mathbf{x}_i)|_\epsilon \tag{1.3}$$

- Support Vector Machines Classification (SVMC)

$$V(y_i, f(\mathbf{x}_i)) = |1 - y_i f(\mathbf{x}_i)|_+ \tag{1.4}$$

where $| \cdot |_\epsilon$ is Vapnik's epsilon-insensitive norm (see later), and y_i is a real number in RN and SVMR, whereas it takes values $-1, 1$ in SVMC. Loss function (1.4) is also called the *soft margin* loss function. For SVMC, two other loss functions will also be considered:

- The *hard margin* loss function:

$$V(y_i, f(\mathbf{x})) = \theta(1 - y_i f(\mathbf{x}_i)) \tag{1.5}$$

- The *misclassification* loss function:

$$V(y_i, f(\mathbf{x})) = \theta(-y_i f(\mathbf{x}_i)) \tag{1.6}$$

where $\theta(\cdot)$ is the Heaviside function. For classification one should minimize (1.6) (or (1.5)), but in practice other loss functions, such as the soft margin one (1.4) [Cortes and Vapnik, 1995, Vapnik, 1995], are used.

A unified justification for kernel machines

Formulating RN and SVM as kernel machines is the first step towards the development of a unified theory that can justify both of them. The second and most important part is the study of the statistical properties of kernel machines within the statistical learning theory framework outlined above. That is, the study of questions like the ones discussed above. Chapter 3 presents a framework based on the technical extension of standard SLT presented in chapter 2 within which the kernel machines listed above are theoretically justified and analyzed. In particular the following technical issues are discussed (see chapter 2 for definitions):

1. Study of the VC-dimension of the kernel machines outlined above (SVM and RN).

2. Study of the V_γ dimension of the kernel machines outlined above (SVM and RN).

3. Based on the results of (1) and (2), uniform convergence in probability both for RN and for SVM regression is shown.

4. Bounds on the expected risk of the solution of RN, of SVMR, and of SMVC are given. In the case of SVMC new bounds both on the misclassification expected error of the solution, as well as the hard and soft margin errors are shown.

This will provide a unified theoretical justification and statistical analysis of RN and SVM. Other loss functions, and therefore other kernel machines, will also be justified within this unified framework. This will be the end of the first part of the thesis.

Learning using ensembles of Kernel Machines

The second part of the thesis, chapter 4, probes into the following question: what happens when instead of training one machine using the available examples we train many of them, each in a different way, and then combine the machines found? There are a number of reasons for considering such a scenario. One intuition is that if a number of machines are trained so that each machine uses different information - that is, for example, different training data or features – then a combination of such machines may be more robust to noise. Another justification is based on the following problem. When training a learning machine, often the problem of choosing parameters (i.e. features or kernels) arises. One approach is to find a way to estimate the right parameters. Another is to train many machines each with different parameters, and then find a way to combine the machines. Finally, it is generally not clear whether an ensemble of many machines would perform better than a single machine.

Chapter 4 studies ensembles of general kernel machines (1.1) for the problem of classification. It considers the general case where each of the machines in the ensemble uses a different kernel. Let T be the number of machines, and let $K^{(t)}$ be the kernel used by machine t. Notice that, as a special case, appropriate choices of $K^{(t)}$ lead to machines that may have different features. Let $f^{(t)}(\mathbf{x})$ be the optimal solution of machine t. Chapter 4 considers ensembles that are linear combinations of the individual machines, that is, the "overall" machine $F(\mathbf{x})$ is of the form:

$$F(\mathbf{x}) = \sum_{t=1}^{T} \beta_t f^{(t)}(\mathbf{x}) \tag{1.7}$$

Two types of ensembles are considered:

1. *Voting Combination of Classifiers* (VCC): this is the case where the coefficients β_t in 1.7 are not learned (i.e. $\beta_t = \frac{1}{T}$).

2. *Adaptive Combinations of Classifiers* (ACC): these are ensembles of the form (1.7) with the coefficients β_t also learned (adapted) from the training data.

The chapter theoretically studies the statistical properties of VCC, and experimentally characterizes both VCC and ACC. In particular, chapter 4:

- Shows new theoretical bounds on the expected error of VCC for kernel machines - voting combinations of SVMs are considered as a special case.

- Presents experiments validating the theoretical findings.

- Experimentally characterizes both VCC and ACC, and compares them with single machines in the case of SVM.

Representations for learning: an application to object detection

An important issue that arises when kernel machines are used is that of the choice of the kernel and the data representation. In fact the two issues are closely related, since a kernel effectively defines a feature space where the data are mapped to [Wahba, 1990, Vapnik, 1998]. Finding appropriate kernels and data representations is very much problem specific. Chapter 5 studies this issue in the particular case of object detection in images.

The trainable system for object detection used in chapter 5 is based on [Papageorgiou *et al.*, 1998b] and can be used to learn any class of objects. The overall framework has been motivated and successfully applied in the past [Papageorgiou *et al.*, 1998b]. The system consists of three parts:

- A set of (positive) example images of the object class considered (i.e. images of frontal faces) and a set of negative examples (i.e. any non-face image) are collected.

- The images are transformed into vectors in a chosen representation (i.e. in a simple case this can be a vector of the size of the image with the values at each pixel location).

- The vectors (examples) are used to train a SVM classifier to learn the classification task of separating positive from negative examples.

Two choices need to be made: the representation in the second stage, and the kernel of the SVM in the third stage. These are the main issues addressed experimentally in chapter 5. The object detection system is trained with different representations: in the simplest case the pixel values of the images are used, while in a different case features and kernels extracted from probabilistic models describing the class of images considered (i.e. images of faces) are used. The chapter also addresses the following questions: can feature selection improve performance of the SVM classifier? can SVM perform well even when many (possibly irrelevant) features are used? Based on the experimental findings and the theoretical results of the thesis, chapter 5 discusses a number of topics and suggest conjectures regarding representations and learning.

1.3.1 Contributions of the thesis

To summarize, the thesis will consist of three main parts. First (chapter 2) the basic theoretical tools are reviewed: standard Statistical Learning Theory (SLT) and a technical extension of it. Within the extended SLT a theoretical justification and statistical analysis of kernel learning machines, including Support Vector Machines (SVM) and Regularization Networks (RN), is provided (chapter 3). In the second part other learning architectures, namely ensembles of learning machines, are investigated (chapter 4). Finally an application to object detection provides a testbed to discuss important practical issues involved in using learning machines, in particular the problem of finding appropriate data representations (chapter 5). The main contributions of the thesis can be summarized as follows:

1. The thesis reviews standard Statistical Learning Theory and develops an extension within which a *new* (unified) theoretical justification of a number of kernel machines, including RN and SVM, is provided.

2. Within the extended SLT framework, *new* bounds on the expected error (performance) of a large class of kernel machines and particularly SVM, the main learning machines considered in the thesis, are proven.

3. In the second part ensembles of machines are studied. Two types of ensembles are defined: voting combinations, and adaptive combinations. *New* theoretical results on the statistical properties of voting ensembles of kernel machines for classification are shown.

4. The new theoretical findings on voting ensembles of machines are experimentally validated. Both voting and adaptive combinations of machines are further characterized experimentally.

5. The third part discusses some important practical issues, particularly the problem of finding appropriate data representations for learning. A trainable system for object detection in images provides the main experimental setup where ideas are tested and discussed.

Chapter 2

Statistical Learning Theory

2.1 A Mathematical formulation of learning

We consider the case of learning from examples as defined in the statistical learning theory framework [Vapnik, 1982, Vapnik, 1995, Vapnik, 1998]. We have two sets of variables $\mathbf{x} \in X \subseteq R^d$ and $y \in Y \subseteq R$ that are related by a probabilistic relationship. We say that the relationship is probabilistic because generally an element of X does not determine uniquely an element of Y, but rather a probability distribution on Y. This can be formalized assuming that a probability distribution $P(\mathbf{x}, y)$ is defined over the set $X \times Y$. The probability distribution $P(\mathbf{x}, y)$ is unknown, and under very general conditions can be written as $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$ where $P(y|\mathbf{x})$ is the conditional probability of y given $\mathbf{x}$, and $P(\mathbf{x})$ is the marginal probability of $\mathbf{x}$. We are provided with *examples* of this probabilistic relationship, that is with a data set $D_\ell \equiv \{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^{\ell}$ called the *training data*, obtained by sampling ℓ times the set $X \times Y$ according to $P(\mathbf{x}, y)$. The problem of learning consists in, given the data set D_ℓ, providing an *estimator*, that is a function $f : X \to Y$, that can be used, given any value of $\mathbf{x} \in X$, to predict a value y.

In statistical learning theory, the standard way to solve the learning problem consists in defining a *risk functional*, which measures the average amount of error associated with an estimator, and then to look for the estimator, among the allowed ones, with the lowest risk. If $V(y, f(\mathbf{x}))$ is the loss function measuring the error we make when we predict y by $f(\mathbf{x})$, then the average error is the so called *expected risk*:

$$I[f] \equiv \int_{X,Y} V(y, f(\mathbf{x}))P(\mathbf{x}, y)\; d\mathbf{x}dy \tag{2.1}$$

We assume that the expected risk is defined on a “large” class of functions $\mathcal{F}$ and we will denote by f_0 the function which minimizes the expected risk in $\mathcal{F}$:

$$f_0(\mathbf{x}) = \arg\min_{\mathcal{F}} I[f] \tag{2.2}$$

The function f_0 is our ideal estimator, and it is often called the *target* function.

2.1.1 The Empirical Risk Minimization learning principle

Unfortunately the target function cannot be found in practice, because the probability distribution $P(\mathbf{x}, y)$ that defines the expected risk is unknown, and only a sample of it, the data set D_ℓ, is available. To overcome this shortcoming we need an *induction principle* that we can use to "learn" from the limited number of training data we have. Statistical learning theory as developed by Vapnik builds on the so-called *empirical risk minimization (ERM)* induction principle. The ERM method consists in using the data set D_ℓ to build a stochastic approximation of the expected risk, which is usually called the *empirical risk*, and is defined as:

$$I_{\mathrm{emp}}[f;\ell] = \frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)). \tag{2.3}$$

The central question is whether the expected risk of the minimizer of the empirical risk in $\mathcal{F}$ is close to the expected risk of f_0. Notice that the question is not necessarily whether we can find f_0 but whether we can *"imitate"* f_0 in the sense that the expected risk of our solution is close to that of f_0. Formally the question is finding under which conditions the method of ERM satisfies:

$$\lim_{\ell\to\infty} I_{\mathrm{emp}}[\hat{f}_\ell;\ell] = \lim_{\ell\to\infty} I[\hat{f}_\ell] = I[f_0] \tag{2.4}$$

in probability (all statements are probabilistic since we start with $P(\mathbf{x}, y)$ on the data), where we note with $\hat{f}_\ell$ the minimizer of the empirical risk (2.3) in $\mathcal{F}$.

It can been shown (see for example [Vapnik, 1998]) that in order for the limits in eq. (2.4) to hold true in probability, or more precisely, for the empirical risk minimization principle to be *non-trivially consistent* (see [Vapnik, 1998] for a discussion about consistency versus non-trivial consistency), the following *uniform law of large numbers* (which "translates" to *one-sided uniform convergence in probability* of empirical risk to expected risk in $\mathcal{F}$) is a *necessary and sufficient* condition:

$$\lim_{\ell\to\infty} P\left\{\sup_{f\in\mathcal{F}} (I[f] - I_{\mathrm{emp}}[f;\ell]) > \epsilon\right\} = 0 \quad \forall\epsilon > 0 \tag{2.5}$$

Intuitively, if $\mathcal{F}$ is very "large" then we can always find $\hat{f}_\ell \in \mathcal{F}$ with 0 empirical error. This however does not guarantee that the expected risk of $\hat{f}_\ell$ is also close to 0, or close to $I[f_0]$.

Typically in the literature the *two-sided uniform convergence in probability*:

$$\lim_{\ell\to\infty} P\left\{\sup_{f\in\mathcal{F}} |I[f] - I_{\mathrm{emp}}[f;\ell]| > \epsilon\right\} = 0 \quad \forall\epsilon > 0 \tag{2.6}$$

is considered, which clearly implies (2.5). We focus on the stronger two-sided case and note that one can get one-sided uniform convergence with some minor technical changes to the theory. We will not discuss the technical issues involved in the relations between consistency, non-trivial consistency, two-sided and one-sided uniform convergence (a discussion can be found in [Vapnik, 1998]), and from now on we concentrate on the two-sided uniform convergence in probability, which we simply refer to as *uniform convergence*.

The theory of uniform convergence of ERM has been developed in [Vapnik and Chervonenkis, 1971, Vapnik and Chervonenkis, 1981, Vapnik and Chervonenkis, 1991, Vapnik, 1982, Vapnik, 1998]. It has also been studied in the context of *empirical processes* [Dudley, 1984, Pollard, 1984, Dudley *et al.*, 1991]. Here we summarize the main results of the theory. Before doing so, we present some early results on uniform convergence for a particular case, that of density estimation.

2.1.2 Uniform convergence for density estimation: Glivenko-Cantelli and Kolmogorov

Uniform convergence has been studied by Glivenko, Cantelli, and Kolmogorov [Glivenko, 1933, Cantelli, 1933, Kolmogorov, 1933] in a particular case of hypothesis spaces $\mathcal{F}$ considered for the problem of density estimation.

Let X be a 1-dimensional random variable (the results can be generalized to many dimensions), and consider the problem of estimating the probability distribution function of X:

$$F(x) = P\ (X < x)$$

from a set of random independent samples

$$x_1, x_2, ..., x_\ell.$$

obtained in accordance with $F(x)$. Notice that the distribution function $F(x)$ can be written as

$$F(x) = \int_{-\infty}^{\infty} \theta(x - \xi)\ p(\xi)\ d\xi$$

where $p(\xi)$ is the probability density of r.v. X according to which the examples $x_1, x_2, ..., x_\ell$ are drawn.

Following the general learning method outlined above, we consider the empirical distribution function

$$F_{\text{emp}}(x; \ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(x - x_i)$$

with θ being the Heaviside function. Figure 2-1 shows the true and empirical probability distribution functions. As in the general case outlined above, we pose the following question:

How different is the empirical distribution from the true one?

This question can be answered in the framework outlined above by redefining the density estimation problem as follows. Consider the set of functions (hypothesis space):

$$\mathcal{F} = \{f_\alpha(x) = \theta(\alpha - x)\ ;\ \alpha \in (-\infty, \infty)\}$$

and define the loss function V to be simply $V(y, f_\alpha(x)) = f_\alpha(x) = \theta(\alpha - x)$ (notice that there is no y in this special case). With this definition, the empirical error $I_{\text{emp}}[f_\alpha; \ell]$ and the expected error $I[f_\alpha]$ considered above become the empirical and

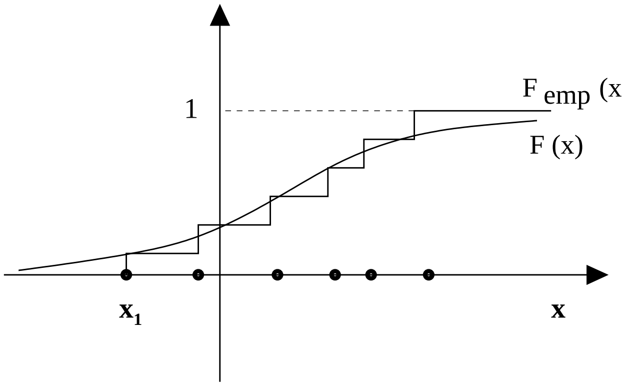

Figure 2-1: The empirical distribution and the true distribution function.

true distribution functions $F_{\text{emp}}(\alpha;\ell)$ and $F(\alpha)$, respectively. Uniform convergence (2.5) can now be rewritten as:

$$\lim_{\ell\to\infty} P\left\{\sup_{f\in\mathcal{F}} |I[f]-I_{\text{emp}}[f;\ell]| > \epsilon\right\} = \lim_{\ell\to\infty} P\left\{\sup_{\alpha\in(-\infty,\infty)} |F(\alpha)-F_{\text{emp}}(\alpha;\ell)| > \epsilon\right\} \tag{2.7}$$

It turns out that for the particular hypothesis space and loss function constructed here, the limit of (2.7) is 0 for every $\epsilon > 0$, namely uniform convergence takes place. In particular the following theorems, shown within a few months difference in 1933, hold:

Theorem 2.1.1 *(Glivenko-Cantelli, 1933) The convergence*

$$\lim_{\ell\to\infty} P\left(\sup_{\alpha\in(-\infty,\infty)} \mid F(\alpha)-F_{\text{emp}}(\alpha;\ell)\mid\right) = 0$$

takes place.

Theorem 2.1.2 *(Kolmogorov-Smirnov Distribution) The rate of convergence of the empirical to the true distribution function follows the following law:*

$$\lim_{\ell\to\infty} P\left(\sqrt{\ell}\sup_{\alpha\in(-\infty,\infty)} \mid F(\alpha)-F_{\text{emp}}(\alpha;\ell)\mid < \epsilon\right) = 1-2\sum_{k=1}^{\infty}(-1)^{k-1}e^{-2\epsilon^2k^2}\ .$$

We do not show the proof of these theorems here, but we only show part of the proof of theorem 2.1.2. Theorem 2.1.2 is based on the following simple observation:

Lemma 2.1.1 *(Kolmogorov, 1933) The probability function*

$$P\{\sqrt{\ell}\sup_{\alpha\in(-\infty,\infty)} |F(\alpha)-F_{\text{emp}}(\alpha;\ell)| < \epsilon\}$$

is independent of the distribution function $F(x)$ on condition the latter is continuous.

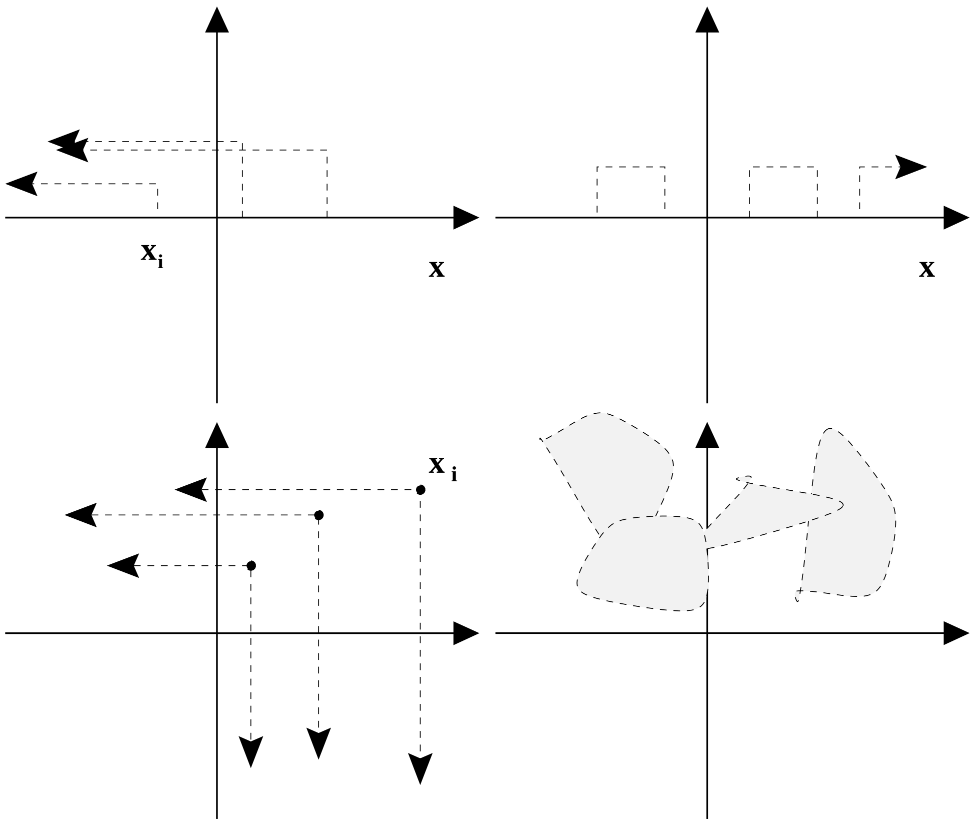

Figure 2-2: The indicator functions considered for the problem of density estimation have the simple "shapes" of orthants shown on the left, while in the general case considered in statistical learning theory the shape and number of the indicator functions can be arbitrary as schematically indicated on the right. Top is a 1-d example, and bottom a 2-d one (bottom left is a 2-d density estimation problem).

Proof. Let X be a random variable with continuous distribution function $F(x)$. For the random variable $Y = F(x)$ corresponds the distribution function $F^0(y)$ such that:

$$F^0(y) = 0 \qquad y \leq 0; \tag{2.8}$$

$$F^0(y) = y \quad 0 \leq y \leq 1; \tag{2.9}$$

$$F^0(y) = 0 \qquad y \geq 1; \tag{2.10}$$

Given empirical distribution functions $F_{\text{emp}}(x;\ell)$ and $F^0_{\text{emp}}(y;\ell)$ for X and Y after ℓ observations, the following hold

$$F_{\text{emp}}(x;\ell) - F(x) = F^0_{\text{emp}}[F(x);\ell] - F^0[F(x)] = F^0_{\text{emp}}(y;\ell) - F(y) \Rightarrow$$

$$\Rightarrow \sup_x |F_{\text{emp}}(x;\ell) - F(x)| = \sup_y |F^0_{\text{emp}}(y;\ell) - F^0(y)|.$$

So the probability function $P\{\sqrt{\ell}\ \sup_\alpha\ |F_{\text{emp}}(\alpha;\ell) - F(\alpha)| < \epsilon\}$ for any continuous distribution function is identical to that when the true distribution is the uniform one

$F^0(x)$. □

Using this trick of transforming all distributions to uniform ones, Kolmogorov proved theorem 2.1.2 in a short paper in 1933 ([Kolmogorov, 1933] - see translation [Kolmogorov, 1992]).

Before discussing uniform convergence for general sets of functions (hypothesis spaces), it is worth providing some intuition about the relation between the density estimation case considered in this section and the general case of the previous and next sections. In the case of density estimation the set of functions considered are indicator function (take values 0 or 1) that have particular shapes: the functions take the value 1 inside an orthant (for example in the 1-d case this is the space $x < \alpha$ – see figure 2-2 also for a 2-d example), and the value 0 outside. Uniform convergence in this case means that as the number of examples increases, the empirical number of points that "fall within an orthant" approach the expected one *for all orthants simultaneously*. In the general case considered in statistical learning theory, the indicator functions do not necessarily have the simple shape of an orthant (see figure 2-2). In this case uniform convergence implies that as the number of examples increases, the empirical number of points that "fall within a function (shape)" approaches the expected one *simultaneously for all indicator functions in the hypothesis space*.

Although, as shown by Glivenko, Cantelli, and Kolmogorov, uniform convergence takes places for indicator functions corresponding to orthants, it does not necessarily take place for any space of indicator functions. Statistical learning theory provides the conditions under which uniform convergence takes place for a set (hypothesis space) of indicator functions: the conditions are given in terms of quantities that characterize hypothesis spaces, namely VC-entropy, growth function, and VC-dimension of a hypothesis space (see below). In the more general case, real-valued functions (instead of indicator ones) are also considered. From this point of view, statistical learning theory, as we outline below, is a generalization of the Glivenko-Cantelli and Kolmogorov theorems that hold only for the particular case of indicator functions corresponding to orthants. We now turn to discuss the general case of uniform convergence in the framework of statistical learning theory.

2.1.3 Uniform convergence for supervised learning

Vapnik and Chervonenkis [Vapnik and Chervonenkis, 1971, Vapnik and Chervonenkis, 1981] studied under what conditions uniform convergence of the empirical risk to expected risk takes place. The results are formulated in terms of three important quantities that measure the complexity of a set of functions: the *VC entropy*, the *annealed VC entropy*, and the *growth function*. We begin with the definitions of these quantities. First we define the *minimal ϵ-net* of a set, which intuitively measures the "cardinality" of a set at "resolution" ϵ:

Definition 2.1.1 *Let A be a set in a metric space $\mathcal{A}$ with distance metric d. For a fixed $\epsilon > 0$, the set $B \subseteq \mathcal{A}$ is called an ϵ-net of A in $\mathcal{A}$, if for any point $a \in A$ there is a point $b \in B$ such that $d(a, b) < \epsilon$. We say that the set B is a minimal ϵ-net of A*

in $\mathcal{A}$, if it is finite and contains the minimal number of elements.

Given a training set $D_\ell = \{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^{\ell}$, consider the set of ℓ-dimensional vectors:

$$q(f) = (V(y_1, f(\mathbf{x}_1)), ..., V(y_\ell, f(\mathbf{x}_\ell))) \tag{2.11}$$

with $f \in \mathcal{F}$, and define the number of elements of the minimal ϵ-net of this set under the metric:

$$d(q(f), q(f')) = \max_{1 \leq i \leq \ell} |V(y_i, f(\mathbf{x}_i)) - V(y_i, f'(\mathbf{x}_i))|$$

to be $\mathcal{N}^{\mathcal{F}}(\epsilon; D_\ell)$ (which clearly depends both on $\mathcal{F}$ and on the loss function V). Intuitively this quantity measures how many different functions effectively we have at "resolution" ϵ, when we only care about the values of the functions at points in D_ℓ. Using this quantity we now give the following definitions:

Definition 2.1.2 *Given a set $X \times Y$ and a probability $P(\mathbf{x}, y)$ defined over it, the VC entropy of a set of functions $V(y, f(\mathbf{x}))$, $f \in \mathcal{F}$, on a data set of size ℓ is defined as:*

$$H^{\mathcal{F}}(\epsilon; \ell) \equiv \int_{X,Y} \ln \mathcal{N}^{\mathcal{F}}(\epsilon; D_\ell) \prod_{i=1}^{\ell} P(\mathbf{x}_i, y_i) d\mathbf{x}_i dy_i$$

Definition 2.1.3 *Given a set $X \times Y$ and a probability $P(\mathbf{x}, y)$ defined over it, the annealed VC entropy of a set of functions $V(y, f(\mathbf{x}))$, $f \in \mathcal{F}$, on a data set of size ℓ is defined as:*

$$H^{\mathcal{F}}_{\text{ann}}(\epsilon; \ell) \equiv \ln \int_{X,Y} \mathcal{N}^{\mathcal{F}}(\epsilon; D_\ell) \prod_{i=1}^{\ell} P(\mathbf{x}_i, y_i) d\mathbf{x}_i dy_i$$

Definition 2.1.4 *Given a set $X \times Y$, the growth function of a set of functions $V(y, f(\mathbf{x}))$, $f \in \mathcal{F}$, on a data set of size ℓ is defined as:*

$$G^{\mathcal{F}}(\epsilon; \ell) \equiv \ln \left(\sup_{D_\ell \in (X \times Y)^\ell} \mathcal{N}^{\mathcal{F}}(\epsilon; D_\ell) \right)$$

Notice that all three quantities are functions of the number of data ℓ and of ϵ, and that clearly:

$$H^{\mathcal{F}}(\epsilon; \ell) \leq H^{\mathcal{F}}_{\text{ann}}(\epsilon; \ell) \leq G^{\mathcal{F}}(\epsilon; \ell) \ .$$

These definitions can easily be extended in the case of *indicator functions*, i.e. functions taking binary values[1] such as $\{-1, 1\}$, in which case the three quantities do not depend on ϵ for $\epsilon < 1$, since the vectors (2.11) are all at the vertices of the hypercube $\{0, 1\}^\ell$.

Using these definitions we can now state three important results of statistical learning theory [Vapnik, 1998]:

[1] In the case of indicator functions, y is binary, and V is 0 for $f(x) = y$, 1 otherwise.

- For a given probability distribution $P(\mathbf{x}, y)$:

  1. The necessary and sufficient condition for uniform convergence is that

  $$\lim_{\ell \to \infty} \frac{H^{\mathcal{F}}(\epsilon; \ell)}{\ell} = 0 \ \ \forall \epsilon > 0$$

  2. A sufficient condition for *fast asymptotic rate of convergence*[2] is that

  $$\lim_{\ell \to \infty} \frac{H^{\mathcal{F}}_{\text{ann}}(\epsilon; \ell)}{\ell} = 0 \ \ \forall \epsilon > 0$$

  It is an open question whether this is also a necessary condition.

- A sufficient condition for distribution *independent* (that is, for any $P(\mathbf{x}, y)$) fast rate of convergence is that

  $$\lim_{\ell \to \infty} \frac{G^{\mathcal{F}}(\epsilon; \ell)}{\ell} = 0 \ \ \forall \epsilon > 0$$

  For indicator functions this is also a necessary condition.

According to statistical learning theory, these three quantities are what one should consider when designing and analyzing learning machines: the VC-entropy and the annealed VC-entropy for an analysis which depends on the probability distribution $P(\mathbf{x}, y)$ of the data, and the growth function for a distribution *independent* analysis. We consider only distribution *independent* results, although the reader should keep in mind that distribution dependent results are likely to be important in the future.

Unfortunately the growth function of a set of functions is difficult to compute in practice. So the standard approach in statistical learning theory is to use an upper bound on the growth function which is given using another important quantity, the *VC-dimension*, which is another (*looser*) measure of the complexity, *capacity*, of a set of functions. We concentrate on this quantity, but it is important that the reader keeps in mind that the VC-dimension is in a sense a "weak" measure of complexity of a set of functions, so it typically leads to loose upper bounds on the growth function: in general one is better off, theoretically, using directly the growth function. We now discuss the VC-dimension and its implications for learning.

The VC-dimension was first defined for the case of indicator functions and then was extended to real valued functions.

Definition 2.1.5 *The VC-dimension of a set $\{\theta(f(\mathbf{x})), f \in \mathcal{F}\}$, of indicator functions is the maximum number h of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_h$ that can be separated into two classes in all 2^h possible ways using functions of the set.*
If, for any number N, it is possible to find N points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ that can be separated in all the 2^N possible ways, we will say that the VC-dimension of the set is infinite.

[2] This means that for any $\ell > \ell_0$ we have that $P\{\sup_{f \in \mathcal{F}} |I[f] - I_{\text{emp}}[f; \ell]| > \epsilon\} < e^{-c\epsilon^2 \ell}$ for some constant $c > 0$. Intuitively, fast rate is typically needed in practice.

The remarkable property of this quantity is that, although as we mentioned the VC-dimension only provides an upper bound to the growth function, in the case of indicator functions, *finiteness of the VC-dimension is a* **necessary and sufficient** *condition for uniform convergence (eq. (2.6))* **independent** *of the underlying distribution $P(\mathbf{x}, y)$.*

Definition 2.1.6 *Let $A \leq V(y, f(\mathbf{x})) \leq B$, $f \in \mathcal{F}$, with A and $B < \infty$. The VC-dimension of the set $\{V(y, f(\mathbf{x})),\ f \in \mathcal{F}\}$ is defined as the VC-dimension of the set of indicator functions $\{\theta\left(V(y, f(\mathbf{x})) - \alpha\right),\quad \alpha \in (A, B)\}$.*

Sometimes we refer to the VC-dimension of $\{V(y, f(\mathbf{x})),\ f \in \mathcal{F}\}$ as the VC dimension of V in $\mathcal{F}$. It can be easily shown that for $y \in \{-1, +1\}$ and for $V(y, f(\mathbf{x})) = \theta(-yf(\mathbf{x}))$ as the loss function, the VC dimension of V in $\mathcal{F}$ computed using definition 2.1.6 is equal to the VC dimension of the set of indicator functions $\{\theta(f(\mathbf{x})),\ f \in \mathcal{F}\}$ computed using definition 2.1.5. In the case of real valued functions, finiteness of the VC-dimension is *only sufficient* for uniform convergence. Later in this chapter we will discuss a measure of capacity that provides also necessary conditions.

An important outcome of the work of Vapnik and Chervonenkis is that the uniform deviation between empirical risk and expected risk in a hypothesis space can be bounded in terms of the VC-dimension, as shown in the following theorem:

Theorem 2.1.3 *(Vapnik and Chervonenkis 1971) Let $A \leq V(y, f(\mathbf{x})) \leq B$, $f \in \mathcal{F}$, $\mathcal{F}$ be a set of bounded functions and h the VC-dimension of V in $\mathcal{F}$. Then, with probability at least $1 - \eta$, the following inequality holds simultaneously for all the elements f of $\mathcal{F}$:*

$$I_{\mathrm{emp}}[f;\ell] - (B-A)\sqrt{\frac{h\ln\frac{2e\ell}{h} - \ln(\frac{\eta}{4})}{\ell}} \leq I[f] \leq I_{\mathrm{emp}}[f;\ell] + (B-A)\sqrt{\frac{h\ln\frac{2e\ell}{h} - \ln(\frac{\eta}{4})}{\ell}} \tag{2.12}$$

The quantity $|I[f] - I_{\mathrm{emp}}[f;\ell]|$ is often called *estimation error*, and bounds of the type above are usually called *VC bounds*[3]. From eq. (2.12) it is easy to see that with probability at least $1 - \eta$:

$$I[\hat{f}_\ell] - 2(B-A)\sqrt{\frac{h\ln\frac{2e\ell}{h} - \ln(\frac{\eta}{4})}{\ell}} \leq I[f_0] \leq I[\hat{f}_\ell] + 2(B-A)\sqrt{\frac{h\ln\frac{2e\ell}{h} - \ln(\frac{\eta}{4})}{\ell}} \tag{2.13}$$

where $\hat{f}_\ell$ is, as in (2.4), the minimizer of the empirical risk in $\mathcal{F}$.

A very interesting feature of inequalities (2.12) and (2.13) is that they are non-asymptotic, meaning that they hold for any finite number of data points ℓ, and that the error bounds do not necessarily depend on the dimensionality of the variable $\mathbf{x}$.

Observe that theorem (2.1.3) and inequality (2.13) are meaningful in practice only if the VC-dimension of the loss function V in $\mathcal{F}$ is finite and less than ℓ. Since the

[3] It is important to note that bounds on the expected risk using the annealed VC-entropy also exist. These are tighter than the VC-dimension ones.

space $\mathcal{F}$ where the loss function V is defined is usually very large (i.e. all functions in L_2), one typically considers smaller hypothesis spaces $\mathcal{H}$. The cost associated with restricting the space is called the *approximation error* (see below). In the literature, space $\mathcal{F}$ where V is defined is called the *target space*, while $\mathcal{H}$ is what is called the *hypothesis space*. Of course, all the definitions and analysis above still hold for $\mathcal{H}$, where we replace f_0 with the minimizer of the expected risk in $\mathcal{H}$, $\hat{f}_\ell$ is now the minimizer of the empirical risk in $\mathcal{H}$, and h the VC-dimension of the loss function V in $\mathcal{H}$. Inequalities (2.12) and (2.13) suggest a method for achieving good generalization: not only minimize the empirical risk, but instead minimize a combination of the empirical risk and the complexity of the hypothesis space. This observation leads us to the method of *Structural Risk Minimization* that we describe below. Before doing so, we first present a technical extension of the standard SLT of Vapnik that will provide the basis for developing a unified justification of the kernel machines in chapter 3.

ϵ-uniform convergence and the V_γ dimension

As mentioned above finiteness of the VC-dimension is *not* a necessary condition for uniform convergence in the case of real valued functions. To get a necessary condition we need a slight extension of the VC-dimension that has been developed (among others) in [Kearns and Shapire, 1994, Alon *et al.*, 1993], known as the V_γ–dimension[4]. Here we summarize the main results of that theory that we will also use later on to design regression machines for which we will have distribution independent uniform convergence.

Definition 2.1.7 *Let $A \leq V(y, f(\mathbf{x})) \leq B$, $f \in \mathcal{F}$, with A and $B < \infty$. The V_γ-dimension of V in $\mathcal{F}$ (of the set $\{V(y, f(\mathbf{x})),\ f \in \mathcal{F}\}$) is defined as the the maximum number h of vectors $(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_h, y_h)$ that can be separated into two classes in all 2^h possible ways using rules:*

$$\begin{array}{l} \textit{class 1 if: } V(y_i, f(\mathbf{x}_i)) \geq s + \gamma \\ \textit{class 0 if: } V(y_i, f(\mathbf{x}_i)) \leq s - \gamma \end{array}$$

for $f \in \mathcal{F}$ and some $s \geq 0$. If, for any number N, it is possible to find N points $(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_N, y_N)$ that can be separated in all the 2^N possible ways, we will say that the V_γ-dimension of V in $\mathcal{F}$ is infinite.

Notice that for $\gamma = 0$ this definition becomes the same as definition 2.1.6 for VC-dimension. Intuitively, for $\gamma > 0$ the "rule" for separating points is more restrictive than the rule in the case $\gamma = 0$. It requires that there is a "margin" between the points: points for which $V(y, f(\mathbf{x}))$ is between $s + \gamma$ and $s - \gamma$ are not classified. As a consequence, the V_γ dimension is a decreasing function of γ and in particular is smaller than the VC-dimension.

[4] In the literature, other quantities, such as the *fat-shattering* dimension and the P_γ dimension, are also defined. They are closely related to each other, and are essentially equivalent to the V_γ dimension for our purpose. The reader can refer to [Alon *et al.*, 1993, Bartlett *et al.*, 1996] for an in-depth discussion on this topic.

If V is an indicator function, say $\theta(-yf(\mathbf{x}))$, then for any γ definition 2.1.7 reduces to that of the VC-dimension of a set of indicator functions.

Generalizing slightly the definition of eq. (2.6) we will say that for a given $\epsilon > 0$ the ERM method converges ϵ-uniformly in $\mathcal{F}$ in probability, (or that there is ϵ-uniform convergence) if:

$$\lim_{\ell\to\infty} P\left\{\sup_{f\in\mathcal{F}} |I_{\mathrm{emp}}[f;\ell] - I[f]| > \epsilon\right\} = 0. \tag{2.14}$$

Notice that if eq. (2.14) holds for every $\epsilon > 0$ we have uniform convergence (eq. (2.6)). It can be shown (variation of [Vapnik, 1998]) that ϵ-uniform convergence in probability implies that:

$$I[\hat{f}_\ell] \leq I[f_0] + 2\epsilon \tag{2.15}$$

in probability, where, as before, $\hat{f}_\ell$ is the minimizer of the empirical risk and f_0 is the minimizer of the expected expected risk in $\mathcal{F}$[5].

The basic theorems for the V_γ-dimension are the following:

Theorem 2.1.4 *(Alon et al. , 1993) Let $A \leq V(y, f(\mathbf{x}))) \leq B$, $f \in \mathcal{F}$, $\mathcal{F}$ be a set of bounded functions. For any $\epsilon > 0$, if the V_γ dimension of V in $\mathcal{F}$ is finite for $\gamma = \alpha\epsilon$ for some constant $\alpha \geq \frac{1}{48}$, then the ERM method ϵ-converges in probability.*

Theorem 2.1.5 *(Alon et al. , 1993) Let $A \leq V(y, f(\mathbf{x}))) \leq B$, $f \in \mathcal{F}$, $\mathcal{F}$ be a set of bounded functions. The ERM method uniformly converges (in probability) if and only if the V_γ dimension of V in $\mathcal{F}$ is finite for every $\gamma > 0$. So finiteness of the V_γ dimension for every $\gamma > 0$ is a necessary* **and** *sufficient condition for distribution independent uniform convergence of the ERM method for real-valued functions.*

Theorem 2.1.6 *(Alon et al. , 1993) Let $A \leq V(y, f(\mathbf{x})) \leq B$, $f \in \mathcal{F}$, $\mathcal{F}$ be a set of bounded functions. For any $\epsilon \geq 0$, for all $\ell \geq \frac{2}{\epsilon^2}$ we have that if h_γ is the V_γ dimension of V in $\mathcal{F}$ for $\gamma = \alpha\epsilon$ ($\alpha \geq \frac{1}{48}$), h_γ finite, then:*

$$P\left\{\sup_{f\in\mathcal{F}} |I_{\mathrm{emp}}[f;\ell] - I[f]| > \epsilon\right\} \leq \mathcal{G}(\epsilon, \ell, h_\gamma), \tag{2.16}$$

where $\mathcal{G}$ is an increasing function of h_γ and a decreasing function of ϵ and ℓ, with $\mathcal{G} \to 0$ as $\ell \to \infty$ [6].

From this theorem we can easily see that for any $\epsilon > 0$, for all $\ell \geq \frac{2}{\epsilon^2}$:

$$P\left\{I[\hat{f}_\ell] \leq I[f_0] + 2\epsilon\right\} \geq 1 - 2\mathcal{G}(\epsilon, \ell, h_\gamma), \tag{2.17}$$

where $\hat{f}_\ell$ is, as before, the minimizer of the empirical risk in $\mathcal{F}$. An important observations to keep in mind is that theorem 2.1.6 requires the V_γ dimension of the loss

[5] This is like ϵ-learnability in the PAC model [Valiant, 1984].

[6] Closed forms of $\mathcal{G}$ can be derived (see for example [Alon *et al.*, 1993]) but we do not present them here for simplicity of notation.

function V in $\mathcal{F}$. In the case of classification, this implies that if we want to derive bounds on the expected misclassification we have to use the V_γ dimension of the loss function $\theta(-yf(\mathbf{x}))$ (which is the $VC-\textit{dimension}$ of the set of indicator functions $\{\text{sgn }(f(\mathbf{x})), f \in \mathcal{F}\}$), and *not* the V_γ dimension of the set $\mathcal{F}$. It will be important to keep this observation in mind when studying classification machines in chapter 3.

2.2 The Structural Risk Minimization learning principle

The idea of SRM is to define a nested sequence of hypothesis spaces $H_1 \subset H_2 \subset \ldots \subset H_{n(\ell)}$ with $n(\ell)$ a non-decreasing integer function of ℓ, where each hypothesis space H_i has VC-dimension finite and larger than that of all previous sets, i.e. if h_i is the VC-dimension of space H_i, then $h_1 \leq h_2 \leq \ldots \leq h_{n(\ell)}$. For example H_i could be the set of polynomials of degree i, or a set of splines with i nodes, or some more complicated nonlinear parameterization. For each element H_i of the structure the solution of the learning problem is:

$$\hat{f}_{i,\ell} = \arg \min_{f \in H_i} I_{\text{emp}}[f;\ell] \tag{2.18}$$

Because of the way we define our structure it should be clear that the larger i is the smaller the empirical error of $\hat{f}_{i,\ell}$ is (since we have greater “flexibility” to fit our training data), but the larger the VC-dimension part (second term) of the right hand side of (2.12) is. Using such a nested sequence of more and more complex hypothesis spaces, the SRM learning technique consists of choosing the space $H_{n^*(\ell)}$ for which the right hand side of inequality (2.12) is minimized. It can be shown [Vapnik, 1982] that for the chosen solution $\hat{f}_{n^*(\ell),\ell}$ inequalities (2.12) and (2.13) hold with probability at least $(1-\eta)^{n(\ell)} \approx 1-n(\ell)\eta$ [7], where we replace h with $h_{n^*(\ell)}$, f_0 with the minimizer of the expected risk in $H_{n^*(\ell)}$, namely $f_{n^*(\ell)}$, and $\hat{f}_\ell$ with $\hat{f}_{n^*(\ell),\ell}$.

With an appropriate choice of $n(\ell)$[8] it can be shown that as $\ell \to \infty$ and $n(\ell) \to \infty$, the expected risk of the solution of the method approaches in probability the minimum of the expected risk in $\mathcal{H} = \bigcup_{i=1}^{\infty} H_i$, namely $I[f_\mathcal{H}]$. Moreover, if the target function f_0 belongs to the closure of $\mathcal{H}$, then eq. (2.4) holds in probability (see for example [Vapnik, 1998]).

However, in practice ℓ is finite (“small”), so $n(\ell)$ is small which means that $\mathcal{H} = \bigcup_{i=1}^{n(\ell)} H_i$ is a small space. Therefore $I[f_\mathcal{H}]$ may be much larger than the expected risk of our target function f_0, since f_0 may not be in $\mathcal{H}$. The distance between $I[f_\mathcal{H}]$ and $I[f_0]$ is called the approximation error and can be bounded using results from approximation theory. We do not discuss these results here and refer the reader to [Lorentz, 1986, DeVore, 1998].

[7] We want (2.12) to hold simultaneously for all spaces H_i, since we choose the best $\hat{f}_{i,\ell}$.

[8] Various cases are discussed in [Devroye *et al.*, 1996], i.e. $n(\ell) = \ell$.

2.2.1 Structural Risk Minimization using the V_γ dimension

The theory of the V_γ dimension justifies the "extended" SRM method we describe below. It is important to keep in mind that the method we describe is only of theoretical interest and will only be used later as a theoretical motivation for RN and SVM. It should be clear that all the definitions and analysis above still hold for any hypothesis space $\mathcal{H}$, where we replace f_0 with the minimizer of the expected risk in $\mathcal{H}$, $\hat{f}_\ell$ is now the minimizer of the empirical risk in $\mathcal{H}$, and h the VC-dimension of the loss function V in $\mathcal{H}$.

Let ℓ be the number of training data. For a fixed $\epsilon > 0$ such that $\ell \geq \frac{2}{\epsilon^2}$, let $\gamma = \frac{1}{48}\epsilon$, and consider, as before, a nested sequence of hypothesis spaces $H_1 \subset H_2 \subset \ldots \subset H_{n(\ell,\epsilon)}$, where each hypothesis space H_i has V_γ-dimension finite and larger than that of all previous sets, i.e. if h_i is the V_γ-dimension of space H_i, then $h_1 \leq h_2 \leq \ldots \leq h_{n(\ell,\epsilon)}$. For each element H_i of the structure consider the solution of the learning problem to be:

$$\hat{f}_{i,\ell} = \arg\min_{f \in H_i} I_{\text{emp}}[f;\ell]. \tag{2.19}$$

Because of the way we define our structure the larger i is the smaller the empirical error of $\hat{f}_{i,\ell}$ is (since we have more "flexibility" to fit our training data), but the larger the right hand side of inequality (2.16) is. Using such a nested sequence of more and more complex hypothesis spaces, this *extended SRM* learning technique consists of finding the structure element $H_{n^*(\ell,\epsilon)}$ for which the trade off between empirical error and the right hand side of (2.16) is optimal. One practical idea is to find numerically for each H_i the "effective" ϵ_i so that the bound (2.16) is the same for all H_i, and then choose $\hat{f}_{i,\ell}$ for which the sum of the empirical risk and ϵ_i is minimized.

We *conjecture* that as $\ell \to \infty$, for appropriate choice of $n(\ell,\epsilon)$ with $n(\ell,\epsilon) \to \infty$ as $\ell \to \infty$, the expected risk of the solution of the method converges in probability to a value less than 2ϵ away from the minimum expected risk in $\mathcal{H} = \bigcup_{i=1}^{\infty} H_i$. Notice that we described an SRM method for a fixed ϵ. If the V_γ dimension of H_i is finite for every $\gamma > 0$, we can further modify the extended SRM method so that $\epsilon \to 0$ as $\ell \to \infty$. We *conjecture* that if the target function f_0 belongs to the closure of $\mathcal{H}$, then as $\ell \to \infty$, with appropriate choices of ϵ, $n(\ell,\epsilon)$ and $n^*(\ell,\epsilon)$ the solution of this SRM method can be proven (as before) to satisfy eq. (2.4) in probability. Finding appropriate forms of ϵ, $n(\ell,\epsilon)$ and $n^*(\ell,\epsilon)$ is an open theoretical problem (which is mostly a technical matter). Again, as in the case of "standard" SRM, in practice ℓ is finite so $\mathcal{H} = \bigcup_{i=1}^{n(\ell,\epsilon)} H_i$ is a small space and the solution of this method may have expected risk much larger that the expected risk of the target function. Approximation theory can be used to bound this difference [Niyogi and Girosi, 1996].

The proposed method is difficult to implement in practice since it is difficult to decide the optimal trade off between empirical error and the bound (2.16). If we had constructive bounds on the deviation between the empirical and the expected risk like that of theorem 2.1.3 then we could have a practical way of choosing the optimal element of the structure. Unfortunately existing bounds of that type [Alon *et al.*, 1993, Bartlett *et al.*, 1996] are not tight. So the final choice of the element of the structure may be done in practice using other techniques such as cross-validation [Wahba, 1990].

This "extended" structural risk minimization method with the theorems outlined above will provide the basic tools for theoretically justifying and analyzing a number of kernel machines. This is the topic of the next chapter.

Chapter 3

Learning with Kernel Machines

This chapter studies a particular type of learning machines, namely *kernel machines*, within the SRM framework presented in chapter 2. These are learning machines for which the hypothesis space is a subspace of a Reproducing Kernel Hilbert Space (hence the name kernel machines). Two particular cases in this family of machines are Regularization Networks (RN) and Support Vector Machines (SVM). The chapter discusses general kernel machines, and, based on the theory presented in chapter 2, presents a theoretical justification and statistical analysis particularly of RN and SVM. A large part of this chapter can be found in [Evgeniou *et al.*, 1999].

3.1 Setup of kernel machines

Following the mathematical formulation of learning considered in Statistical Learning Theory, two are the key choices to be made when designing a learning machine:

1. Choose the loss function V.

2. Choose the set of possible functions – hypothesis space.

This chapter considers hypothesis spaces that are subsets of a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ defined by a kernel K (see below for an overview of RKHS). Within the RKHS $\mathcal{H}$, we perform structural risk minimization by first defining a structure of hypothesis spaces.

The basic idea is to define a structure in terms of a nested sequence of hypothesis spaces $H_1 \subset H_2 \subset \ldots \subset H_{n(\ell)}$ with H_m being the set of functions f in the RKHS $\mathcal{H}$ with:

$$\|f\|_K^2 \leq A_m^2, \tag{3.1}$$

where $\|f\|_K^2$ is a norm in $\mathcal{H}$ defined by the positive definite function K (see below), and A_m is a monotonically increasing sequence of positive constants. Following the SRM method outlined in chapter 2, for each m we solve the following constrained minimization problem:

$$\begin{aligned} \min_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) & \\ \|f\|_K^2 \leq A_m^2 & \end{aligned} \tag{3.2}$$

Machines of the form (3.2) are called *kernel machines*. Learning using kernel machines of the form (3.2) leads to using the Lagrange multiplier λ_m and to minimizing

$$\frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda_m(\|f\|_K^2 - A_m^2),$$

with respect to $f \in \mathcal{H}$ and maximizing with respect to $\lambda_m \geq 0$ for each element of the structure. As discussed in chapter 2, we can then choose the optimal $n^*(\ell)$ and the associated $\lambda^*(\ell)$, and get the optimal solution $\hat{f}_{n^*(\ell)}$.

The solution we get using this method is clearly the same as the solution of:

$$\min_{f\in\mathcal{H}} \frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda^*(\ell)\|f\|_K^2 \tag{3.3}$$

where $\lambda^*(\ell)$ is the optimal Lagrange multiplier corresponding to the optimal element of the structure $A_{n^*(\ell)}$. We also define *kernel machines* to be machines of the general form:

$$\min_{f\in\mathcal{H}} \frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda\|f\|_K^2 \tag{3.4}$$

We discuss more formally the relations between machines (3.2) and (3.4) at the end of the chapter, and for the moment we call kernel machines to be either of the two formulations. Furthermore, it turns out that the solution of the kernel machines (3.4) for any differentiable loss function V (this is the case for the loss functions considered in this thesis) has always the same form, that is:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i), \tag{3.5}$$

with the coefficients c_i found by solving the minimization problem (3.4) [Girosi, 1998]. Often the term kernel machines will refer also to machines of the form (3.5).

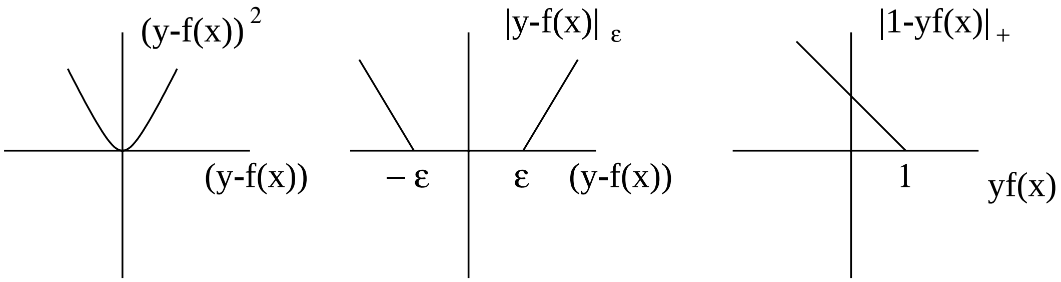

Figure 3-1: The three loss functions considered: L_2 for RN (left), L_ϵ for SVMR (middle), and soft margin for SVMC (right).

Notice that the choice of the loss function V leads to a family of learning machines. In particular it leads to classical L_2 Regularization Networks, to SVM regression, and to SVM classification, for the following specific choices of the loss function V:

- $V(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ for Regularization Networks.
- $V(y, f(\mathbf{x})) = |y - f(\mathbf{x})|_\epsilon$ for SVM regression.
- $V(y, f(\mathbf{x})) = |1 - yf(\mathbf{x})|_+$ for SVM classification.

where $|\cdot|_\epsilon$ is Vapnik's ϵ-insensitive loss function L_ϵ [Vapnik, 1998] with $|x|_\epsilon = |x| - \epsilon$ for $|x| \geq \epsilon$ and 0 otherwise, while $|\cdot|_+$ is the soft margin loss function [Vapnik, 1998, Cortes and Vapnik, 1995] with $|x|_+ = x$ for $x \geq 0$ and 0 otherwise. These loss functions are shown in figure 3-1. For SVM classification the loss functions:

- $V(y, f(\mathbf{x})) = \theta(1 - yf(\mathbf{x}))$ (hard margin loss function), and
- $V(y, f(\mathbf{x})) = \theta(-yf(\mathbf{x}))$ (misclassification loss function)

will also be discussed. These particular kernel machines are reviewed in this chapter. First an overview of RKHS, which are the hypothesis spaces considered in the thesis, is presented.

3.1.1 Reproducing Kernel Hilbert Spaces: a brief overview

A Reproducing Kernel Hilbert Space (RKHS) [Aronszajn, 1950] is a Hilbert space $\mathcal{H}$ of functions defined over some bounded domain $X \subset R^d$ with the property that, for each $\mathbf{x} \in X$, the evaluation functionals $\mathcal{F}_\mathbf{x}$ defined as

$$\mathcal{F}_\mathbf{x}[f] = f(\mathbf{x}) \quad \forall f \in \mathcal{H}$$

are linear, bounded functionals. The boundedness means that there exists a $U = U_x \in R^+$ such that:

$$|\mathcal{F}_\mathbf{x}[f]| = |f(\mathbf{x})| \leq U||f||$$

for all f in the RKHS.

It can be proved [Wahba, 1990] that to every RKHS $\mathcal{H}$ there corresponds a *unique positive definite* function $K(\mathbf{x}, \mathbf{y})$ of two variables in X, called the *reproducing kernel* of $\mathcal{H}$ (hence the terminology RKHS), that has the following *reproducing property*:

$$f(\mathbf{x}) =< f(\mathbf{y}), K(\mathbf{y}, \mathbf{x}) >_\mathcal{H} \quad \forall f \in \mathcal{H}, \tag{3.6}$$

where $< \cdot, \cdot >_\mathcal{H}$ denotes the scalar product in $\mathcal{H}$. The function K behaves in $\mathcal{H}$ as the delta function does in L_2, although L_2 is not a RKHS (the functionals $\mathcal{F}_\mathbf{x}$ are clearly not bounded).

To make things clearer we sketch a way to construct a RKHS, which is relevant to this thesis. The mathematical details (such as the convergence or not of certain series) can be found in the theory of integral equations [Hochstadt, 1973, Cochran, 1972, Courant and Hilbert, 1962].

Let us assume that we have a sequence of positive numbers λ_n and linearly independent functions $\phi_n(\mathbf{x})$ such that they define a function $K(\mathbf{x}, \mathbf{y})$ in the following way [1]:

[1] When working with complex functions $\phi_n(\mathbf{x})$ this formula should be replaced with $K(\mathbf{x}, \mathbf{y}) \equiv \sum_{n=0}^\infty \lambda_n \phi_n(\mathbf{x}) \phi_n^*(\mathbf{y})$

$$K(\mathbf{x}, \mathbf{y}) \equiv \sum_{n=0}^{\infty} \lambda_n \phi_n(\mathbf{x}) \phi_n(\mathbf{y}), \tag{3.7}$$

where the series is well defined (for example it converges uniformly). A simple calculation shows that the function K defined in eq. (3.7) is positive definite. Let us now take as our Hilbert space to be the set of functions of the form:

$$f(\mathbf{x}) = \sum_{n=0}^{\infty} a_n \phi_n(\mathbf{x}) \tag{3.8}$$

for any $a_n \in R$, and define the scalar product in our space to be:

$$< \sum_{n=0}^{\infty} a_n \phi_n(\mathbf{x}), \sum_{n=0}^{\infty} d_n \phi_n(\mathbf{x}) >_{\mathcal{H}} \equiv \sum_{n=0}^{\infty} \frac{a_n d_n}{\lambda_n}. \tag{3.9}$$

Assuming that all the evaluation functionals are bounded, it is now easy to check that such an Hilbert space is a RKHS with reproducing kernel given by $K(\mathbf{x}, \mathbf{y})$. In fact we have:

$$< f(\mathbf{y}), K(\mathbf{y}, \mathbf{x}) >_{\mathcal{H}} = \sum_{n=0}^{\infty} \frac{a_n \lambda_n \phi_n(\mathbf{x})}{\lambda_n} = \sum_{n=0}^{\infty} a_n \phi_n(\mathbf{x}) = f(\mathbf{x}), \tag{3.10}$$

hence equation (3.6) is satisfied.

Notice that when we have a finite number of ϕ_n, the λ_n can be arbitrary (finite) numbers, since convergence is ensured. In particular they can all be equal to one.

Generally, it is easy to show [Wahba, 1990] that whenever a function K of the form (3.7) is available, it is possible to construct a RKHS as shown above. Vice versa, for any RKHS there is a unique kernel K and corresponding λ_n, ϕ_n, that satisfy equation (3.7) and for which equations (3.8), (3.9) and (3.10) hold for all functions in the RKHS. Moreover, equation (3.9) shows that the norm of the RKHS has the form:

$$\|f\|_K^2 = \sum_{n=0}^{\infty} \frac{a_n^2}{\lambda_n} \tag{3.11}$$

The ϕ_n consist a basis for the RKHS (not necessarily orthonormal), and the kernel K is the "correlation" matrix associated with these basis functions. It is in fact well known that there is a close relation between Gaussian processes and RKHS [Marroquin *et al.*, 1987, Girosi *et al.*, 1991, Poggio and Girosi, 1998]. Wahba [Wahba, 1990] discusses in depth the relation between regularization, RKHS and correlation functions of Gaussian processes. The choice of the ϕ_n defines a space of functions – the functions that are spanned by the ϕ_n.

We also call the space $\{(\phi_n(\mathbf{x}))_{n=1}^{\infty}, \ \mathbf{x} \in X\}$ the *feature space* induced by the kernel K. The choice of the ϕ_n defines the feature space where the data $\mathbf{x}$ are "mapped". We refer to the dimensionality of the feature space as the *dimensionality of the RKHS*. This is clearly equal to the number of basis elements ϕ_n, which does not necessarily have to be infinite. For example, with K a Gaussian, the dimensionality of the RKHS is infinite ($\phi_n(\mathbf{x})$ are the Fourier components $e^{i\mathbf{n}\cdot\mathbf{x}}$), while when K is a polynomial of degree k ($K(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}\cdot\mathbf{y})^k$), the dimensionality of the RKHS is finite, and all the infinite sums above are replaced with finite sums.

It is well known that expressions of the form (3.7) actually abound. In fact, it follows from Mercer's theorem [Hochstadt, 1973] that any function $K(\mathbf{x},\mathbf{y})$ which is the kernel of a positive operator in $L_2(\Omega)$ has an expansion of the form (3.7), in which the ϕ_i and the λ_i are respectively the orthogonal eigenfunctions and the positive eigenvalues of the operator corresponding to K. In [Stewart, 1976] it is reported that the positivity of the operator associated to K is equivalent to the statement that the kernel K is positive definite, that is the matrix $K_{ij} = K(\mathbf{x}_i,\mathbf{x}_j)$ is positive definite for all choices of distinct points $\mathbf{x}_i \in X$. Notice that a kernel K could have an expansion of the form (3.7) in which the ϕ_n are not necessarily its eigenfunctions. The only requirement is that the ϕ_n are linearly independent but not necessarily orthogonal.

In the case that the space X has finite cardinality, the "functions" f are evaluated only at a finite number of points $\mathbf{x}$. If M is the cardinality of X, then the RKHS becomes an M-dimensional space where the functions f are basically M-dimensional vectors, the kernel K becomes an $M \times M$ matrix, and the condition that makes it a valid kernel is that it is a symmetric positive definite matrix (semi-definite if M is larger than the dimensionality of the RKHS). Positive definite matrices are known to be the ones which define dot products, i.e. $fKf^T \geq 0$ for every f in the RKHS. The space consists of all M-dimensional vectors f with finite norm fKf^T.

Summarizing, RKHS are Hilbert spaces where the dot product is defined using a function $K(\mathbf{x},\mathbf{y})$ which needs to be positive definite just like in the case that X has finite cardinality. The elements of the RKHS are all functions f that have a finite norm given by equation (3.11). Notice the equivalence of a) choosing a specific RKHS $\mathcal{H}$ b) choosing a set of ϕ_n and λ_n c) choosing a reproducing kernel K. The last one is the most natural for most applications.

Finally, it is useful to notice that the solutions of the methods discussed in this chapter can be written both in the form (3.5), and in the form (3.8). Often in the literature formulation (3.5) is called the *dual* form of f, while (3.8) is called the *primal* form of f.

3.2 Regularization Networks

In this section we consider the approximation scheme that arises from the minimization of the quadratic functional

$$\min_{f\in\mathcal{H}} H[f] = \frac{1}{\ell}\sum_{i=1}^{\ell}(y_i - f(\mathbf{x}_i))^2 + \lambda\|f\|_K^2 \tag{3.12}$$

for a fixed λ. Formulations like equation (3.12) are a special form of regularization theory developed by Tikhonov, Ivanov [Tikhonov and Arsenin, 1977, Ivanov, 1976] and others to solve ill-posed problems and in particular to solve the problem of approximating the functional relation between $\mathbf{x}$ and y given a finite number of examples $D_\ell = \{\mathbf{x}_i, y_i\}_{i=1}^{\ell}$.

In classical regularization the data term is an L_2 loss function for the empirical risk, whereas the second term – called *stabilizer* – is usually written as a functional $\Omega(f)$ with certain properties [Tikhonov and Arsenin, 1977, Poggio and Girosi, 1989, Girosi *et al.*, 1995]. Here we consider a special class of stabilizers, that is the norm $\|f\|_K^2$ in a RKHS induced by a symmetric, positive definite function $K(\mathbf{x},\mathbf{y})$. This choice allows us to develop a framework of regularization which includes most of the usual regularization schemes. The only significant omission in this treatment – that we make here for simplicity – is the restriction on K to be symmetric positive definite so that the stabilizer is a norm. However, the theory can be extended without problems to the case in which K is positive semidefinite, in which case the stabilizer is a semi-norm [Wahba, 1990, Madych and Nelson, 1990a, Dyn, 1991, Dyn *et al.*, 1986]. This approach was also sketched in [Smola and Schölkopf, 1998].

The stabilizer in equation (3.12) effectively constrains f to be in the RKHS defined by K. It is possible to show (see for example [Poggio and Girosi, 1989, Girosi *et al.*, 1995]) that the function that minimizes the functional (3.12) has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i), \tag{3.13}$$

where the coefficients c_i depend on the data and satisfy the following linear system of equations:

$$(K + \lambda I)\mathbf{c} = \mathbf{y} \tag{3.14}$$

where I is the identity matrix, and we have defined

$$(\mathbf{y})_i = y_i \ , \quad (\mathbf{c})_i = c_i \ , \quad (K)_{ij} = K(\mathbf{x}_i, \mathbf{x}_j).$$

It is remarkable that the solution of the more general case of

$$\min_{f \in \mathcal{H}} H[f] = \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i - f(\mathbf{x}_i)) + \lambda \|f\|_K^2, \tag{3.15}$$

where the function V is any differentiable function, is quite similar: the solution has exactly the same general form of (3.13), though the coefficients cannot be found anymore by solving a linear system of equations as in equation (3.14) [Girosi, 1991, Girosi *et al.*, 1991, Smola and Schölkopf, 1998]. For a proof see [Girosi, 1998].

The approximation scheme of equation (3.13) has a simple interpretation in terms of a network with one layer of hidden units [Poggio and Girosi, 1992, Girosi *et al.*, 1995]. Using different kernels we get various RN's. A short list of examples is given in Table 1.

When the kernel K is positive semidefinite, there is a subspace of functions f which have norm $\|f\|_K^2$ equal to zero. They form the null space of the functional $\|f\|_K^2$ and in this case the minimizer of (3.12) has the form [Wahba, 1990]:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{\alpha=1}^{k} b_\alpha \psi_\alpha(\mathbf{x}), \tag{3.16}$$

Kernel Function	Regularization Network
$K(\mathbf{x}-\mathbf{y}) = \exp(-\Vert\mathbf{x}-\mathbf{y}\Vert^2)$	Gaussian RBF
$K(\mathbf{x}-\mathbf{y}) = (\Vert\mathbf{x}-\mathbf{y}\Vert^2 + c^2)^{-\frac{1}{2}}$	Inverse Multiquadric
$K(\mathbf{x}-\mathbf{y}) = (\Vert\mathbf{x}-\mathbf{y}\Vert^2 + c^2)^{\frac{1}{2}}$	Multiquadric
$K(\mathbf{x}-\mathbf{y}) = \Vert\mathbf{x}-\mathbf{y}\Vert^{2n+1}$ $K(\mathbf{x}-\mathbf{y}) = \Vert\mathbf{x}-\mathbf{y}\Vert^{2n}\ln(\Vert\mathbf{x}-\mathbf{y}\Vert)$	Thin plate splines
$K(\mathbf{x},\mathbf{y}) = \tanh(\mathbf{x}\cdot\mathbf{y}-\theta)$	(only for some values of θ) Multi Layer Perceptron
$K(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}\cdot\mathbf{y})^d$	Polynomial of degree d
$K(x,y) = B_{2n+1}(x-y)$	B-splines
$K(x,y) = \frac{\sin(d+1/2)(x-y)}{\sin\frac{(x-y)}{2}}$	Trigonometric polynomial of degree d

Table 3.1: Some possible kernel functions. The first four are radial kernels. The multiquadric and thin plate splines are positive semidefinite and thus require an extension of the simple RKHS theory presented here. The last three kernels were proposed by Vapnik (Vapnik,1998), originally for SVM. The last two kernels are one-dimensional: multidimensional kernels can be built by tensor products of one-dimensional ones. The functions B_n are piecewise polynomials of degree n, whose exact definition can be found in (Schumaker,1981).

where $\{\psi_\alpha\}_{\alpha=1}^k$ is a basis in the null space of the stabilizer, which in most cases is a set of polynomials, and therefore will be referred to as the "polynomial term" in equation (3.16). The coefficients b_α and c_i depend on the data. For the classical regularization case of equation (3.12), the coefficients of equation (3.16) satisfy the following linear system:

$$(K+\lambda I)\mathbf{c} + \Psi^T\mathbf{b} = \mathbf{y}, \tag{3.17}$$

$$\Psi\mathbf{c} = 0, \tag{3.18}$$

where I is the identity matrix, and we have defined

$$(\mathbf{y})_i = y_i\ , \quad (\mathbf{c})_i = c_i\ , \quad (\mathbf{b})_i = b_i\ ,$$
$$(K)_{ij} = K(\mathbf{x}_i,\mathbf{x}_j)\ , \quad (\Psi)_{\alpha i} = \psi_\alpha(\mathbf{x}_i).$$

When the kernel is positive definite, as in the case of the Gaussian, the null space of the stabilizer is empty. However, it is often convenient to redefine the kernel and the norm induced by it so that the induced RKHS contains only zero-mean functions, that is functions $f_1(\mathbf{x})$ s.t. $\int_X f_1(\mathbf{x})dx = 0$. In the case of a radial kernel K, for instance, this amounts to considering a new kernel

$$K'(\mathbf{x},\mathbf{y}) = K(\mathbf{x},\mathbf{y}) - \lambda_0$$

without the zeroth order Fourier component, and a norm

$$\|f\|_{K'}^2 = \sum_{n=1}^{\infty} \frac{a_n^2}{\lambda_n}. \tag{3.19}$$

The null space induced by the new K' is the space of constant functions. Then the minimizer of the corresponding functional (3.12) has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K'(\mathbf{x}, \mathbf{x}_i) + b, \tag{3.20}$$

with the coefficients satisfying equations (3.17) and (3.18), that respectively become:

$$(K' + \lambda I)\mathbf{c} + \mathbf{1}b = (K - \lambda_0 I + \lambda I)\mathbf{c} + \mathbf{1}b = (K + (\lambda - \lambda_0)I)\mathbf{c} + \mathbf{1}b = \mathbf{y}, \tag{3.21}$$

$$\sum_{i=1}^{\ell} c_i = 0. \tag{3.22}$$

Equations (3.20) and (3.22) imply that the the minimizer of (3.12) is of the form:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K'(\mathbf{x}, \mathbf{x}_i) + b = \sum_{i=1}^{\ell} c_i (K(\mathbf{x}, \mathbf{x}_i) - \lambda_0) + b = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i) + b. \tag{3.23}$$

Thus we can effectively use a positive definite K *and* the constant b, since the only change in equation (3.21) just amounts to the use of a different λ. Choosing to use a non-zero b effectively means choosing a different feature space and a different stabilizer from the usual case of equation (3.12): the constant feature is not considered in the RKHS norm and therefore is not "penalized". This choice is often quite reasonable, since in many regression and, especially, classification problems, shifts by a constant in f should not be penalized.

In summary, the argument of this section shows that using a RN of the form (3.23) (for a certain class of kernels K) is equivalent to minimizing functionals such as (3.12) or (3.15). The choice of K is equivalent to the choice of a corresponding RKHS and leads to various classical learning techniques such as RBF networks. We discuss connections between regularization and other techniques later in this section.

Notice that in the framework we use here the kernels K are not required to be radial or even shift-invariant. Regularization techniques used to solve supervised learning problems [Poggio and Girosi, 1989, Girosi *et al.*, 1995] were typically used with shift invariant stabilizers (tensor product and additive stabilizers are exceptions, see [Girosi *et al.*, 1995]). We now turn to such kernels.

3.2.1 Examples of Regularization Networks

Radial Basis Functions

Let us consider a special case of the kernel K of the RKHS, which is the standard case in several papers and books on regularization [Wahba, 1990, Poggio and Girosi, 1990, Girosi *et al.*, 1995]: the case in which K is shift invariant, that is $K(\mathbf{x},\mathbf{y}) = K(\mathbf{x}-\mathbf{y})$ and the even more special case of a *radial* kernel $K(\mathbf{x},\mathbf{y}) = K(||\mathbf{x}-\mathbf{y}||)$. A radial positive definite K defines a RKHS in which the "features" ϕ_n are Fourier components that is

$$K(\mathbf{x},\mathbf{y}) \equiv \sum_{n=0}^{\infty} \lambda_n \phi_n(\mathbf{x})\phi_n(\mathbf{y}) \equiv \sum_{n=0}^{\infty} \lambda_n e^{i2\pi\mathbf{n}\cdot\mathbf{x}} e^{-i2\pi\mathbf{n}\cdot\mathbf{y}}. \tag{3.24}$$

Thus any positive definite radial kernel defines a RKHS over $[0,1]$ with a scalar product of the form:

$$< f, g >_{\mathcal{H}} \equiv \sum_{n=0}^{\infty} \frac{\tilde{f}(\mathbf{n})\tilde{g}^*(\mathbf{n})}{\lambda_n}, \tag{3.25}$$

where $\tilde{f}$ is the Fourier transform of f. The RKHS becomes simply the subspace of $L_2([0,1]^d)$ of the functions such that

$$\|f\|_K^2 = \sum_{n=1}^{\infty} \frac{|\tilde{f}(\mathbf{n})|^2}{\lambda_n} < +\infty. \tag{3.26}$$

Functionals of the form (3.26) are known to be *smoothness* functionals. In fact, the rate of decrease to zero of the Fourier transform of the kernel will control the smoothness property of the function in the RKHS. For radial kernels the minimizer of equation (3.12) becomes:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(||\mathbf{x}-\mathbf{x}_i||) + b \tag{3.27}$$

and the corresponding RN is a *Radial Basis Function Network*. Thus Radial Basis Function networks are a special case of RN [Poggio and Girosi, 1989, Girosi *et al.*, 1995].

In fact *all* translation-invariant stabilizers $K(\mathbf{x},\mathbf{x}_i) = K(\mathbf{x}-\mathbf{x}_i)$ correspond to RKHS's where the basis functions ϕ_n are Fourier eigenfunctions and only differ in the spectrum of the eigenvalues (for a Gaussian stabilizer the spectrum is Gaussian, that is $\lambda_n = Ae^{(-n^2/2)}$ (for $\sigma = 1$)). For example, if $\lambda_n = 0$ for all $n > n_0$, the corresponding RKHS consists of all bandlimited functions, that is functions with zero Fourier components at frequencies higher than n_0[2]. Generally λ_n are such that they decrease as n increases, therefore restricting the class of functions to be functions with decreasing high frequency Fourier components.

In classical regularization with translation invariant stabilizers and associated kernels, the common experience, often reported in the literature, is that the form of the kernel does not matter much. It is a conjecture that this may be because all translation invariant K induce the same type of ϕ_n features - the Fourier basis functions.

[2] The simplest K is then $K(x,y) = sinc(x-y)$, or kernels that are convolution with it.

Regularization, generalized splines and kernel smoothers

A number of approximation and learning techniques can be studied in the framework of regularization theory and RKHS. For instance, starting from a reproducing kernel it is easy [Aronszajn, 1950] to construct kernels that correspond to tensor products of the original RKHS; it is also easy to construct the additive sum of several RKHS in terms of a reproducing kernel.

- **Tensor Product Splines:** In the particular case that the kernel is of the form:

  $$K(\mathbf{x}, \mathbf{y}) = \Pi_{j=1}^{d} k(x^j, y^j)$$

  where x^j is the jth coordinate of vector $\mathbf{x}$ and k is a positive definite function with one-dimensional input vectors, the solution of the regularization problem becomes:

  $$f(\mathbf{x}) = \sum_i c_i \Pi_{j=1}^{d} k(x_i^j, x^j)$$

  Therefore we can get tensor product splines by choosing kernels of the form above [Aronszajn, 1950].

- **Additive Splines:** In the particular case that the kernel is of the form:

  $$K(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} k(x^j, y^j)$$

  where x^j is the jth coordinate of vector $\mathbf{x}$ and k is a positive definite function with one-dimensional input vectors, the solution of the regularization problem becomes:

  $$f(\mathbf{x}) = \sum_i c_i (\sum_{j=1}^{d} k(x_i^j, x^j)) = \sum_{j=1}^{d} (\sum_i c_i k(x_i^j, x^j)) = \sum_{j=1}^{d} f_j(x^j)$$

  So in this particular case we get the class of *additive approximation* schemes of the form:

  $$f(\mathbf{x}) = \sum_{j=1}^{d} f_j(x^j)$$

A more extensive discussion on relations between known approximation methods and regularization can be found in [Girosi *et al.*, 1995].

Dual representation of Regularization Networks

Every RN can be written as

$$f(\mathbf{x}) = \mathbf{c} \cdot \mathbf{K}(\mathbf{x}) \tag{3.28}$$

where $\mathbf{K}(\mathbf{x})$ is the vector of functions such that $(\mathbf{K}(\mathbf{x}))_i = K(\mathbf{x}, \mathbf{x}_i)$. Since the coefficients $\mathbf{c}$ satisfy the equation (3.14), equation (3.28) becomes

$$f(\mathbf{x}) = (K + \lambda I)^{-1}\mathbf{y} \cdot \mathbf{K}(\mathbf{x}) \ .$$

We can rewrite this expression as

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i b_i(\mathbf{x}) = \mathbf{y} \cdot \mathbf{b}(\mathbf{x}) \tag{3.29}$$

in which the vector $\mathbf{b}(\mathbf{x})$ of basis functions is defined as:

$$\mathbf{b}(\mathbf{x}) = (K + \lambda I)^{-1}\mathbf{K}(\mathbf{x}) \tag{3.30}$$

and now depends on all the data points and on the regularization parameter λ. The representation (3.29) of the solution of the approximation problem is known as the *dual*[3] of equation (3.28), and the basis functions $b_i(\mathbf{x})$ are called the *equivalent kernels*, because of the similarity with the kernel smoothing technique [Silverman, 1984, Härdle, 1990, Hastie and Tibshirani, 1990]. Notice that, while in equation (3.28) the difficult part is the computation of coefficients c_i, the kernel function $K(\mathbf{x}, \mathbf{x}_i)$ being predefined, in the dual representation (3.29) the difficult part is the computation of the basis function $b_i(\mathbf{x})$, the coefficients of the expansion being explicitly given by the y_i.

As observed in [Girosi *et al.*, 1995], the dual representation of a RN shows clearly how careful one should be in distinguishing between local vs. global approximation techniques. In fact, we expect (see [Silverman, 1984] for the 1-D case) that in most cases the kernels $b_i(\mathbf{x})$ decrease with the distance of the data points $\mathbf{x}_i$ from the evaluation point, so that only the neighboring data affect the estimate of the function at $\mathbf{x}$, providing therefore a "local" approximation scheme. Even if the original kernel K is not "local", like the absolute value $|x|$ in the one-dimensional case or the multiquadric $K(\mathbf{x}) = \sqrt{1 + \|\mathbf{x}\|^2}$, the basis functions $b_i(\mathbf{x})$ are bell shaped, local functions, whose locality will depend on the choice of the kernel K, on the density of data points, and on the regularization parameter λ. This shows that apparently "global" approximation schemes can be regarded as local, *memory-based* techniques (see equation 3.29) [Mhaskar, 1993a].

3.2.2 From regression to classification

In the particular case that the unknown function takes only two values, i.e. -1 and 1, we have the problem of binary pattern classification, i.e. the case where we are

[3] Notice that this "duality" is different from the one mentioned at the end of section 3.1.1.

given data that belong to one of two classes (classes -1 and 1) and we want to find a function that separates these classes. It can be shown [Duda and Hart, 1973] that, if V in equation (3.15) is $(y-f(\mathbf{x}))^2$, and if K defines a finite dimensional RKHS, then the minimizer of the equation

$$H[f] = \frac{1}{\ell}\sum_{i=1}^{\ell}(f(\mathbf{x}_i) - y_i)^2 + \lambda\|f\|_K^2, \tag{3.31}$$

for $\lambda \to 0$ approaches *asymptotically* the function in the RKHS that is closest in the L_2 norm to the regression function:

$$f_0(\mathbf{x}) = Pr(y = 1|\mathbf{x}) - Pr(y = -1|\mathbf{x}) \tag{3.32}$$

The *optimal Bayes rule classifier* is given by thresholding the regression function, i.e. by $\text{sign}(f_0(\mathbf{x}))$. Notice that in the case of infinite dimensional RKHS asymptotic results ensuring consistency are available (see [Devroye *et al.*, 1996], theorem 29.8) but depend on several conditions that are not automatically satisfied in the case we are considering. The Bayes classifier is the best classifier, given the correct probability distribution P. However, approximating function (3.32) in the RKHS in L_2 does not necessarily imply that we find the best approximation to the Bayes classifier. For classification, only the sign of the regression function matters and not the exact value of it. Notice that an approximation of the regression function using a mean square error criterion places more emphasis on the most probable data points and not on the most "important" ones which are the ones near the separating boundary.

In the next section we will study Vapnik's more natural approach to the problem of classification that is based on choosing a loss function V different from the square error. This approach leads to solutions that emphasize data points near the separating surface.

3.3 Support Vector Machines

In this section we first discuss the technique of Support Vector Machines (SVM) for Regression (SVMR) [Vapnik, 1995, Vapnik, 1998] in terms of the SVM functional. We will characterize the form of the solution, and then discuss SVM for Classification (SVMC). We also show that SVM for binary pattern classification can be derived as a special case of the regression formulation.

3.3.1 SVM in RKHS

Once again the problem is to learn a functional relation between $\mathbf{x}$ and y given a finite number of examples $D_\ell = \{\mathbf{x}_i, y_i\}_{i=1}^{\ell}$.

The method of SVMR [Vapnik, 1998] corresponds to the following functional

$$H[f] = \frac{1}{\ell}\sum_{i=1}^{\ell}|y_i - f(\mathbf{x}_i)|_\epsilon + \lambda\|f\|_K^2 \tag{3.33}$$

which is a special case of equation (3.15) and where

$$V(x) = |x|_\epsilon \equiv \begin{cases} 0 & \text{if } |x| < \epsilon \\ |x| - \epsilon & \text{otherwise,} \end{cases} \tag{3.34}$$

is the ϵ−Insensitive Loss Function (ILF) (also noted with L_ϵ). Note that the ILF assigns zero cost to errors smaller then ϵ. In other words, for the cost function $| \cdot |_\epsilon$ any function closer than ϵ to the data points is a perfect interpolant. We can think of the parameter ϵ as the resolution at which we want to look the data. For this reason we expect that the larger ϵ is, the simpler the representation will be [Girosi, 1997].

The minimizer of H in the RKHS $\mathcal{H}$ defined by the kernel K has the general form given by equation (3.23), that is

$$f(x) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}_i, \mathbf{x}) + b, \tag{3.35}$$

where we can include the constant b for the same reasons discussed in the previous section.

In order to find the solution of SVM we have to minimize functional (3.33) (with V given by equation (3.34)) with respect to f. Since it is difficult to deal with the function $V(x) = |x|_\epsilon$, the above problem is replaced by the following equivalent problem (by *equivalent* we mean that the same function minimizes both functionals), in which an additional set of variables is introduced:

Problem 3.3.1

$$\min_{f,\xi,xi*} \Phi(f, \xi, \xi^*) = \frac{C}{\ell} \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) + \frac{1}{2} \|f\|_K^2 \tag{3.36}$$

subject to the constraints:

$$\begin{array}{lll} f(\mathbf{x}_i) - y_i & \leq \epsilon + \xi_i & i = 1, \dots, \ell \\ y_i - f(\mathbf{x}_i) & \leq \epsilon + \xi_i^* & i = 1, \dots, \ell \\ \xi_i, \xi_i^* & \geq 0 & i = 1, \dots, \ell. \end{array} \tag{3.37}$$

The parameter C in (3.36) has been introduced in order to be consistent with the standard SVM notations [Vapnik, 1998]. Note that λ in eq. (3.33) corresponds to $\frac{1}{2C}$. The equivalence is established just noticing that in problem (3.3.1) a (linear) penalty is paid only when the absolute value of the error exceeds ϵ (because of the L_ϵ loss function). Notice that if either of the two top constraints is satisfied with some non-zero ξ_i (or ξ_i^*), the other is automatically satisfied with a zero value for ξ_i^* (or ξ_i).

Problem (3.3.1) can be solved through the technique of Lagrange multipliers. For details see [Vapnik, 1998]. The result is that the function which solves problem (3.3.1) can be written as:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b,$$

where α_i^* and α_i are the solution of the following QP-problem:

Problem 3.3.2

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \mathcal{W}(\boldsymbol{\alpha},\boldsymbol{\alpha}^*) = \epsilon\sum_{i=1}^{\ell}(\alpha_i^* + \alpha_i) - \sum_{i=1}^{\ell} y_i(\alpha_i^* - \alpha_i) + \frac{1}{2}\sum_{i,j=1}^{\ell}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(\mathbf{x}_i,\mathbf{x}_j),$$

subject to the constraints:

$$\sum_{i=1}^{\ell}(\alpha_i^* - \alpha_i) = 0,$$
$$0 \leq \alpha_i^*, \alpha_i \leq \frac{C}{\ell}, \qquad i = 1,\dots,\ell.$$

The solutions of problems (3.3.1) and (3.3.2) are related by the Kuhn-Tucker conditions:

$$\alpha_i(f(\mathbf{x}_i) - y_i - \epsilon - \xi_i) = 0 \qquad i = 1,\dots,\ell \tag{3.38}$$
$$\alpha_i^*(y_i - f(\mathbf{x}_i) - \epsilon - \xi_i^*) = 0 \qquad i = 1,\dots,\ell \tag{3.39}$$
$$(\frac{C}{\ell} - \alpha_i)\xi_i = 0 \qquad i = 1,\dots,\ell \tag{3.40}$$
$$(\frac{C}{\ell} - \alpha_i^*)\xi_i^* = 0 \qquad i = 1,\dots,\ell. \tag{3.41}$$

The input data points $\mathbf{x}_i$ for which α_i or α_i^* are different from zero are called *support vectors* (SVs). Observe that α_i and α_i^* cannot be simultaneously different from zero, so that the constraint $\alpha_i\alpha_i^* = 0$ holds true. Any of the SVs for which $0 < \alpha_j < \frac{C}{\ell}$ (and therefore $\xi_j = 0$) can be used to compute the parameter b. In fact, in this case it follows from the Kuhn-Tucker conditions that:

$$f(\mathbf{x}_j) = \sum_{i=1}^{\ell}(\alpha_i^* - \alpha_i)K(\mathbf{x}_i,\mathbf{x}_j) + b = y_j + \epsilon.$$

from which b can be computed. The SVs are those data points $\mathbf{x}_i$ at which the error is either greater or equal to ϵ[4]. Points at which the error is smaller than ϵ are never support vectors, and do not enter in the determination of the solution. A consequence of this fact is that if the SVM were run again on the new data set consisting of only the SVs the same solution would be found.

3.3.2 From regression to classification

In the previous section we discussed the connection between regression and classification in the framework of regularization. In this section, after stating the formulation of SVM for binary pattern classification (SVMC) as developed by Cortes and Vapnik [Cortes and Vapnik, 1995], we discuss a connection between SVMC and SVMR.

[4] In degenerate cases however, it can happen that points whose error is equal to ϵ are not SVs.

We will not discuss the theory of SVMC here; we refer the reader to [Vapnik, 1998]. We point out that the SVM technique has first been proposed for binary pattern classification problems and then extended to the general regression problem [Vapnik, 1995].

SVMC can be formulated as the problem of minimizing:

$$H(f) = \frac{1}{\ell}\sum_{i}^{\ell} |1 - y_i f(\mathbf{x}_i)|_+ + \frac{1}{2C}\|f\|_K^2, \tag{3.42}$$

which is again of the form (3.4). Using the fact that $y_i \in \{-1, +1\}$ it is easy to see that formulation (3.42) is equivalent to the following quadratic programming problem, originally proposed by Cortes and Vapnik [Cortes and Vapnik, 1995]:

Problem 3.3.3

$$\min_{f \in \mathcal{H}, \boldsymbol{\xi}} \Phi(f, \boldsymbol{\xi}) = \frac{C}{\ell}\sum_{i=1}^{\ell} \xi_i + \frac{1}{2}\|f\|_K^2$$

subject to the constraints:

$$\begin{array}{lll} y_i f(\mathbf{x}_i) & \geq 1 - \xi_i, & i = 1, \ldots, \ell \\ \xi_i & \geq 0, & i = 1, \ldots, \ell. \end{array} \tag{3.43}$$

The solution of this problem is again of the form:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}_i, \mathbf{x}) + b, \tag{3.44}$$

where it turns out that $0 \leq c_i \leq \frac{C}{\ell}$. The input data points $\mathbf{x}_i$ for which c_i is different from zero are called, as in the case of regression, *support vectors* (SVs). It is often possible to write the solution $f(\mathbf{x})$ as a linear combination of SVs in a number of different ways (for example in case that the feature space induced by the kernel K has dimensionality lower than the number of SVs). The SVs that appear in *all* these linear combinations are called *essential support vectors.*

Roughly speaking the motivation for problem (3.3.3) is to minimize the empirical error measured by $\sum_{i=1}^{\ell} \xi_i$[5] while controlling capacity measured in terms of the norm of f in the RKHS. In fact, the norm of f is related to the notion of *margin*, an important idea for SVMC for which we refer the reader to [Vapnik, 1998, Burges, 1998].

We now address the following question: what happens if we apply the SVMR formulation given by problem (3.3.1) to the binary pattern classification case, i.e. the case where y_i take values $\{-1, 1\}$, treating classification as a regression on binary data?

[5] For binary pattern classification the empirical error is defined as a sum of binary numbers which in problem (3.3.3) would correspond to $\sum_{i=1}^{\ell} \theta(\xi_i)$. However in such a case the minimization problem becomes computationally intractable. This is why in practice in the cost functional $\Phi(f, \boldsymbol{\xi})$ we approximate $\theta(\xi_i)$ with ξ_i. We discuss this further at the end of this chapter.

Notice that in problem (3.3.1) each example has to satisfy two inequalities (which come out of using the L_ϵ loss function), while in problem (3.3.3) each example has to satisfy one inequality. It is possible to show that for a given constant C in problem (3.3.3), there exist C and ϵ in problem (3.3.1) such that the solutions of the two problems are the same, up to a constant factor. This is summarized in the following theorem:

Theorem 3.3.1 *Suppose the classification problem (3.3.3) is solved with parameter C, and the optimal solution is found to be f. Then, there exists a value $a \in (0, 1)$ such that for $\forall \epsilon \in [a, 1)$, if the regression problem (3.3.1) is solved with parameter $(1 - \epsilon)C$, the optimal solution will be $(1 - \epsilon)f$.*

We refer to [Pontil *et al.*, 1998] for the proof. A direct implication of this result is that one can solve any SVMC problem through the SVMR formulation. It is an open question what theoretical implications theorem 3.3.1 may have about SVMC and SVMR. In particular, chapter 4 presents some recent theoretical results on SVMC that have not yet been extended to SVMR. It is possible that theorem 3.3.1 may help to extend them to SVMR.

3.4 SRM for RNs and SVMs

At the beginning of this chapter we outlined how one should implement both RN and SVM according to SRM. The idea is to solve a series of constrained minimization problems of the form (3.2), namely:

$$\min_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i))$$
$$\|f\|_K^2 \leq A_m^2 \tag{3.45}$$

for a sequence of constants $A_1 < A_2 < \ldots A_{n(\ell)}$, and then pick among the solutions found the optimal one according to the SRM principle presented in chapter 2. The solution found is the same as the one found by solving directly the minimization problem (3.4), namely:

$$\min_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_K^2 \ , \tag{3.46}$$

where λ is the regularization parameter that, according to the SRM principle, should be equal to the optimal Lagrange multiplier found for the optimal $A_{n^*(\ell)}$ (see the beginning of the chapter). We come back to this issue at the end of this chapter, and for the moment we consider only kernel machines of the form (3.45).

For the SRM principle to be used it is required that the hypothesis spaces considered do indeed define a structure. It is required, in other words, that the "complexity" of the set of functions:

$$\left\{ V(y, f(\mathbf{x})) \ ; \ f \in \mathcal{H} \ , \ \|f\|_K^2 \leq A^2 \right\} \tag{3.47}$$

is an increasing function of A. Thus, we need to show that either the VC-dimension or the V_γ dimension of the set of functions (3.47) is an increasing function of A for the loss functions considered. This would theoretically justify the chosen loss functions (for RN and SVM), so we now turn to this issue. We first consider the problem of regression with RN and SVMR. Classification with SVMC is considered later.

3.4.1 Why not use the VC-dimension

Unfortunately it can be shown that when the loss function V is $(y - f(\mathbf{x}))^2$ (L_2) and also when it is $|y_i - f(\mathbf{x}_i)|_\epsilon$ (L_ϵ), the VC-dimension of $V(y, f(\mathbf{x}))$ with $\|f\|_K^2 \leq A^2$ does not depend on A, and is infinite if the RKHS is infinite dimensional. More precisely we have the following theorem (also shown in [Williamson *et al.*, 1998]):

Theorem 3.4.1 *Let D be the dimensionality of a RKHS $\mathcal{H}$. For both the L_2 and the ϵ-insensitive loss function, the VC-dimension of the set of functions $\{V(y, f(\mathbf{x}))\ ;\ f \in \mathcal{H},\ ||f||_K^2 \leq A^2\}$ is $O(D)$, independently of A. Moreover, if D is infinite, the VC-dimension is infinite for any $A \neq 0$.*

Proof

Consider first the case of L_p loss functions. Consider an infinite dimensional RKHS, and the set of functions with norm $\|f\|_K^2 \leq A^2$. If for any N we can find N points that we can shatter using functions of our set according to the rule:

$$\begin{aligned} \text{class } 1 \text{ if :} &\quad |y - f(\mathbf{x})|^p \geq s \\ \text{class } -1 \text{ if :} &\quad |y - f(\mathbf{x})|^p \leq s \end{aligned}$$

then clearly the VC dimension is infinite. Consider N distinct points $(\mathbf{x}_i, y_i)$ with $y_i = 0$ for all i, and let the smallest eigenvalue of matrix G with $G_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ be λ. Since we are in infinite dimensional RKHS, matrix G is always invertible [Wahba, 1990], so $\lambda > 0$ since G is positive definite and finite dimensional (λ may decrease as N increases, but for any finite N it is well defined and $\neq 0$).

For any separation of the points, we consider a function f of the form $f(\mathbf{x}) = \sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x})$, which is a function of the form (3.8). We need to show that we can find coefficients α_i such that the RKHS norm of the function is $\leq A^2$. Notice that the norm of a function of this form is $\boldsymbol{\alpha}^T G \boldsymbol{\alpha}$ where $(\boldsymbol{\alpha})_i = \alpha_i$ (throughout the proofs bold letters are used for noting vectors). Consider the set of linear equations

$$\begin{aligned} \mathbf{x}_j \in \text{class } 1 : &\quad \textstyle\sum_{i=1}^N \alpha_i G_{ij} = s^{\frac{1}{p}} + \eta \quad \eta > 0 \\ \mathbf{x}_j \in \text{class } -1 : &\quad \textstyle\sum_{i=1}^N \alpha_i G_{ij} = s^{\frac{1}{p}} - \eta \quad \eta > 0 \end{aligned}$$

Let $s = 0$. If we can find a solution $\boldsymbol{\alpha}$ to this system of equations such that $\boldsymbol{\alpha}^T G \boldsymbol{\alpha} \leq A^2$ we can perform this separation, and since this is any separation we can shatter the N points. Notice that the solution to the system of equations is $G^{-1}\boldsymbol{\eta}$ where $\boldsymbol{\eta}$ is the vector whose components are $(\boldsymbol{\eta})_i = \eta$ when $\mathbf{x}_i$ is in class 1, and $-\eta$ otherwise. So we need $(G^{-1}\boldsymbol{\eta})^T G (G^{-1}\boldsymbol{\eta}) \leq A^2 \Rightarrow \boldsymbol{\eta}^T G^{-1} \boldsymbol{\eta} \leq A^2$. Since the smallest eigenvalue of G is $\lambda > 0$, we have that $\boldsymbol{\eta}^T G^{-1} \boldsymbol{\eta} \leq \frac{\boldsymbol{\eta}^T \boldsymbol{\eta}}{\lambda}$. Moreover $\boldsymbol{\eta}^T \boldsymbol{\eta} = N\eta^2$. So if we choose η small enough such that $\frac{N\eta^2}{\lambda} \leq A^2 \Rightarrow \eta^2 \leq \frac{A^2\lambda}{N}$, the norm of the solution is less than A^2, which completes the proof.

For the case of the L_ϵ loss function the argument above can be repeated with $y_i = \epsilon$ to prove again that the VC dimension is infinite in an infinite dimensional RKHS.

Finally, notice that the same proof can be repeated for finite dimensional RKHS to show that the VC dimension is never less than the dimensionality D of the RKHS, since it is possible to find D points for which matrix G is invertible and repeat the proof above. As a consequence the VC dimension cannot be controlled by A. □

It is thus impossible to use the standard SRM with this kind of hypothesis spaces: in the case of finite dimensional RKHS, the RKHS norm of f cannot be used to define a structure of spaces with increasing VC-dimensions, and in the (typical) case that the dimensionality of the RKHS is infinite, it is not even possible to use the bounds on the expected error that the theory gives (the bounds in chapter 2). So *the VC-dimension cannot be used directly neither for RN nor for SVMR.*

On the other hand, we can still use the V_γ dimension and the extended SRM method outlined in chapter 2. This is discussed next.

3.4.2 A theoretical justification of RN and SVM regression

It turns out that, under mild "boundedness" conditions, for both the $(y - f(\mathbf{x}))^2$ (L_2) and the $|y_i - f(\mathbf{x}_i)|_\epsilon$ (L_ϵ) loss functions, the V_γ dimension of $V(y, f(\mathbf{x}))$ with $\|f\|_K^2 \leq A^2$ *does* depend on A, and is finite even if the RKHS is infinite dimensional. More precisely we have the following theorem (a tighter computation that holds under some conditions is shown later in this section):

Theorem 3.4.2 *Let D be the dimensionality of a RKHS $\mathcal{H}$ with kernel K. Assume for both the input space X and the output space Y are bounded, let R be the radius of the smallest ball containing the data $\mathbf{x}$ in the feature space induced by kernel K, and assume $y \in [0,1]$. For both the L_2 and the ϵ-insensitive loss function, the V_γ dimension of the set of functions $\{V(y, f(\mathbf{x}))\ ;\ f \in \mathcal{H},\ ||f||_K^2 \leq A^2\}$ is finite for $\forall\ \gamma > 0$, with $h \leq \mathcal{O}(\min\ (D, \frac{(R^2+1)(A^2+1)}{\gamma^2})$.*

Proof

Let's consider first the case of the L_1 loss function. Let B be the upper bound on the loss function (which exists for the loss functions considered since both spaces X and Y are bounded). From definition 2.1.7 we can decompose the rules for separating points as follows:

$$\begin{array}{rl} \text{class } 1 \text{ if} & y_i - f(\mathbf{x}_i) \geq s + \gamma \\ \text{or} & y_i - f(\mathbf{x}_i) \leq -(s + \gamma) \\ \text{class } -1 \text{ if} & y_i - f(\mathbf{x}_i) \leq s - \gamma \\ \text{and} & y_i - f(\mathbf{x}_i) \geq -(s - \gamma) \end{array} \tag{3.48}$$

for some $\gamma \leq s \leq B - \gamma$. For any N points, the number of separations of the points we can get using rules (3.48) is not more than the number of separations we can get using the product of two indicator functions with margin (of hyperplanes with margin):

$$\begin{array}{rrl} \text{function (a)}: & \text{class } 1 \text{ if} & y_i - f_1(\mathbf{x}_i) \geq s_1 + \gamma \\ & \text{class } -1 \text{ if} & y_i - f_1(\mathbf{x}_i) \leq s_1 - \gamma \\ \text{function (b)}: & \text{class } 1 \text{ if} & y_i - f_2(\mathbf{x}_i) \geq -(s_2 - \gamma) \\ & \text{class } -1 \text{ if} & y_i - f_2(\mathbf{x}_i) \leq -(s_2 + \gamma) \end{array} \tag{3.49}$$

where $\|f_1\|_K^2 \leq A^2$ and $\|f_2\|_K^2 \leq A^2$ and $\gamma \leq s_1, s_2 \leq B - \gamma$. This is shown as follows.

Clearly the product of the two indicator functions (3.49) has less "separating power" when we add the constraints $s_1 = s_2 = s$ and $f_1 = f_2 = f$. Furthermore, even with these constraints we still have more "separating power" than we have using rules (3.48): any separation realized using (3.48) can also be realized using the product of the two indicator functions (3.49) under the constraints $s_1 = s_2 = s$ and $f_1 = f_2 = f$. For example, if $y - f(\mathbf{x}) \geq s + \gamma$ then indicator function (a) will give $+1$, indicator function (b) will give also $+1$, so their product will give $+1$ which is what we get if we follow (3.48). Similarly for all other cases.

As mentioned in chapter 2, for any N points the number of ways we can separate them is bounded by the growth function. Moreover, for products of indicator functions it is known [Vapnik, 1998] that the growth function is bounded by the product of the growth functions of the indicator functions. Furthermore, the indicator functions in (3.49) are hyperplanes with margin in the $D + 1$ dimensional space of vectors $\{\phi_n(\mathbf{x}), y\}$ where the radius of the data is $R^2 + 1$, the norm of the hyperplane is bounded by $A^2 + 1$, (where in both cases we add 1 because of y), and the margin is at least $\frac{\gamma^2}{A^2+1}$. The V_γ dimension h_γ of these hyperplanes is known [Vapnik, 1998, Bartlett and Shawe-Taylor, 1998b] to be bounded by $h_\gamma \leq \min((D+1)+1, \frac{(R^2+1)(A^2+1)}{\gamma^2})$. So the growth function of the separating rules (3.48) is bounded by the product of the growth functions $(\frac{eN}{h_\gamma})^{h_\gamma}$, that is $\mathcal{G}(N) \leq \left((\frac{eN}{h_\gamma})^{h_\gamma}\right)^2$ whenever $N \geq h_\gamma$. If h_γ^{reg} is the V_γ dimension, then h_γ^{reg} cannot be larger than the larger number N for which inequality $2^N \leq (\frac{eN}{h_\gamma})^{2h_\gamma}$ holds. From this, after some algebraic manipulations (take the log of both sides) we get that $N \leq 5h_\gamma$, therefore $h_\gamma^{reg} \leq 5 \min\ (D+2, \frac{(R^2+1)(A^2+1)}{\gamma^2})$ which proves the theorem for the case of L_1 loss functions.

For general L_p loss functions we can follow the same proof where (3.48) now needs to be rewritten as:

$$\begin{array}{rl} \text{class } 1 \text{ if} & y_i - f(\mathbf{x}_i) \geq (s+\gamma)^{\frac{1}{p}} \\ \text{or} & f(\mathbf{x}_i) - y_i \geq (s+\gamma)^{\frac{1}{p}} \\ \text{class } -1 \text{ if} & y_i - f(\mathbf{x}_i) \leq (s-\gamma)^{\frac{1}{p}} \\ \text{and} & f(\mathbf{x}_i) - y_i \leq (s-\gamma)^{\frac{1}{p}} \end{array} \tag{3.50}$$

Moreover, for $1 < p < \infty$, $(s+\gamma)^{\frac{1}{p}} \geq s^{\frac{1}{p}} + \frac{\gamma}{pB}$ (since $\gamma = \left((s+\gamma)^{\frac{1}{p}}\right)^p - \left(s^{\frac{1}{p}}\right)^p =$ $= ((s+\gamma)^{\frac{1}{p}} - s^{\frac{1}{p}})(((s+\gamma)^{\frac{1}{p}})^{p-1} + \ldots + (s^{\frac{1}{p}})^{p-1}) \leq ((s+\gamma)^{\frac{1}{p}} - s^{\frac{1}{p}})(B + \ldots B) =$ $= ((s+\gamma)^{\frac{1}{p}} - s^{\frac{1}{p}})(pB)$) and $(s-\gamma)^{\frac{1}{p}} \leq s^{\frac{1}{p}} - \frac{\gamma}{pB}$ (similarly). Repeating the same argument as above, we get that the V_γ dimension is bounded by 5 min $(D + 2, \frac{(pB)^2(R^2+1)(A^2+1)}{\gamma^2})$. Finally, for the L_ϵ loss function (3.48) can be rewritten as:

$$\begin{array}{rl} \text{class 1 if} & y_i - f(\mathbf{x}_i) \geq s + \gamma + \epsilon \\ \text{or} & f(\mathbf{x}_i) - y_i \geq s + \gamma + \epsilon \\ \text{class } -1 \text{ if} & y_i - f(\mathbf{x}_i) \leq s - \gamma + \epsilon \\ \text{and} & f(\mathbf{x}_i) - y_i \leq s - \gamma + \epsilon \end{array} \tag{3.51}$$

where calling $s' = s + \epsilon$ we can simply repeat the proof above and get the same upper bound on the V_γ dimension as in the case of the L_1 loss function. (Notice that the constraint $\gamma \leq s \leq B - \gamma$ is not taken into account. Taking this into account may slightly change the V_γ dimension for L_ϵ. Since it is a constraint, it can only decrease - or not change - the V_γ dimension). □

Notice that for fixed γ and fixed radius of the data the only variable that controls the V_γ dimension is the upper bound on the RKHS norm of the functions, namely A. Moreover, the V_γ dimension is finite for $\forall\ \gamma > 0$; therefore, according to theorem (2.1.5), ERM uniformly converges in $\{f \in \mathcal{H}\ ;\ \|f\|_K^2 \leq A^2\}$ for any $A < \infty$, both for RN and for SVMR.

Theoretically, we can use the extended SRM method with a sequence of hypothesis spaces each defined for different As. To repeat, for a fixed $\gamma > 0$ (we can let γ go to 0 as $\ell \to \infty$) we first define a structure $H_1 \subset H_2 \subset \ldots \subset H_{n(\ell)}$ where H_m is the set of bounded functions f in a RKHS with $||f||_K^2 \leq A_m^2$, $A_m < \infty$, and the numbers A_m form an increasing sequence. Then we minimize the empirical risk in each H_m by solving the problem:

$$\begin{array}{c} \text{minimize } \frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) \\ \text{subject to}: \ ||f||_K^2 \leq A_m^2 \end{array} \tag{3.52}$$

To solve this minimization problem we minimize

$$\frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda_m(||f||_K^2 - A_m^2) \tag{3.53}$$

with respect to f and maximize with respect to the Lagrange multiplier λ_m. If f_m is the solution of this problem, at the end we choose the optimal $f_{n^*(\ell)}$ in $F_{n^*(\ell)}$ with the associated $\lambda_{n^*(\ell)}$, where optimality is decided based on a trade off between empirical error and the bound (2.16) for the fixed γ (which, as we mentioned, can approach zero). In the case of RN, V is the L_2 loss function, whereas in the case of SVMR it is the ϵ-insensitive loss function.

In practice it is difficult to implement the extended SRM for two main reasons. First, as we discussed in chapter 2, SRM using the V_γ dimension is practically difficult because we do not have tight bounds to use in order to pick the optimal $F_{n^*(\ell)}$ (combining theorems 3.4.2 and 2.1.6, bounds on the expected risk of RN and SVMR machines of the form (3.52) can be derived, but these bounds are not practically useful). Second, even if we could make a choice of $F_{n^*(\ell)}$, it is computationally difficult to implement SRM since (3.52) is a constrained minimization problem one with non-linear constraints, and solving such a problem for a number of spaces H_m can be computationally difficult. So implementing SRM using the V_γ dimension of nested subspaces of a RKHS is practically a very difficult problem.

On the other hand, if we had the optimal Lagrange multiplier $\lambda_{n^*(\ell)}$, we could simply solve the unconstrained minimization problem:

$$\frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda_{n^*(\ell)}||f||_K^2 \tag{3.54}$$

both for RN and for SVMR. This is exactly the problem we solve in practice, as described earlier in this chapter. Since the value $\lambda_{n^*(\ell)}$ is not known in practice, we can only "implement" the extended SRM approximately by minimizing (3.54) with various values of λ and then picking the best λ using techniques such as cross-validation [Allen, 1974, Wahba, 1980, Wahba, 1985, Kearns *et al.*, 1995], Generalized Cross Validation, Finite Prediction Error and the MDL criteria (see [Vapnik, 1998] for a review and comparison). An important remark to make is that for machine (3.54), although as mentioned before it is equivalent to machine (3.2) for the "right" choice of $\lambda_{n^*(\ell)}$, because in general we do not know $\lambda_{n^*(\ell)}$ without actually training machine (3.2) we cannot directly use the theorems of chapter 2. We will come back to this issue at the end of this chapter.

Summarizing, both the RN and the SVMR methods discussed can be seen as approximations of the extended SRM method using the V_γ dimension, with nested hypothesis spaces being of the form $\{f \in \mathcal{H} \,:\, \|f\|_K^2 \leq A^2\}$, $\mathcal{H}$ being a RKHS defined by kernel K. For both RN and SVMR the V_γ dimension of the loss function V in these spaces is finite for $\forall\ \gamma > 0$, so the ERM method uniformly converges for any $A < \infty$, and we can use the extended SRM method outlined in chapter 2.

The V_γ dimension in a special case

Before proceeding to the case of classification, we present one more computation of the V_γ dimension for RN and SVMR that can be used to compute an "empirical" V_γ dimension, as discussed below.

We assume that the data $\mathbf{x}$ are restricted so that for any finite dimensional matrix G with entries $G_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ the largest eigenvalue of G is always $\leq M^2$ for a given constant M. We consider only the case that the RKHS is infinite dimensional. We note with B the upper bound of $V(y, f(\mathbf{x}))$. Under these assumptions we can show that:

Theorem 3.4.3 *The V_γ dimension for regression using L_1 loss function and for hypothesis space $\mathcal{H}_A = \{f(\mathbf{x}) = \sum_{n=1}^{\infty} w_n\phi_n(\mathbf{x}) + b \mid \sum_{n=1}^{\infty}\frac{w_n^2}{\lambda_n} \leq A^2\}$ is finite for $\forall\gamma > 0$. In particular:*

1. *If b is constrained to be zero, then $V_\gamma \leq \left[\frac{M^2A^2}{\gamma^2}\right]$*

2. If b is a free parameter, $V_\gamma \leq 4\left[\frac{M^2A^2}{\gamma^2}\right]$

Proof of part 1.

Suppose we can find $N > \left[\frac{M^2A^2}{\gamma^2}\right]$ points $\{(x_1, y_1), ..., (x_N, y_N)\}$ that we can shatter. Let $s \in [\gamma, B - \gamma]$ be the value of the parameter used to shatter the points.

Consider the following "separation"[6]: if $|y_i| < s$, then (x_i, y_i) belongs in class 1. All other points belong in class -1. For this separation we need:

$$\begin{array}{ll} |y_i - f(x_i)| \geq s + \gamma, & \text{if } |y_i| < s \\ |y_i - f(x_i)| \leq s - \gamma, & \text{if } |y_i| \geq s \end{array} \tag{3.55}$$

This means that: for points in class 1 f takes values either $y_i+s+\gamma+\delta_i$ or $y_i-s-\gamma-\delta_i$, for $\delta_i \geq 0$. For points in the second class f takes values either $y_i + s - \gamma - \delta_i$ or $y_i - s + \gamma + \delta_i$, for $\delta_i \in [0, (s - \gamma)]$. So (3.55) can be seen as a system of linear equations:

$$\sum_{n=1}^{\infty} w_n \phi_n(\mathbf{x}_i) = t_i. \tag{3.56}$$

with t_i being $y_i+s+\gamma+\delta_i$, or $y_i-s-\gamma-\delta_i$, or $y_i+s-\gamma-\delta_i$, or $y_i-s+\gamma+\delta_i$, depending on i. We first use lemma 3.4.1 to show that for any solution (so t_i are fixed now) there is another solution with not larger norm that is of the form $\sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x})$.

Lemma 3.4.1 *Among all the solutions of a system of equations (3.56) the solution with the minimum RKHS norm is of the form: $\sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x})$ with $\boldsymbol{\alpha} = G^{-1}\boldsymbol{t}$.*

Proof of lemma

We introduce the $N \times \infty$ matrix $A_{in} = \sqrt{\lambda_n}\phi_n(\mathbf{x}_i)$ and the new variable $z_n = \frac{w_n}{\sqrt{\lambda_n}}$. We can write system (3.56) as follows:

$$A\mathbf{z} = \mathbf{t}. \tag{3.57}$$

Notice that the solution of the system of equation 3.56 with minimum RKHS norm, is equivalent to the Least Square (LS) solution of equation 3.57. Let us denote with $\mathbf{z}^0$ the LS solution of system 3.57. We have:

$$\mathbf{z}^0 = (A^\top A)^+ A^\top \mathbf{t} \tag{3.58}$$

where + denotes pseudoinverse. To see how this solution looks like we use Singular Value Decomposition techniques:

$$\begin{array}{rcl} A & = & U\Sigma V^\top, \\ A^\top & = & V\Sigma U^\top, \end{array}$$

[6] Notice that this separation might be a "trivial" one in the sense that we may want all the points to be +1 or all to be -1 i.e. when all $|y_i| < s$ or when all $|y_i| \geq s$ respectively.

from which $A^\top A = V\Sigma^2 V^\top$ and $(A^\top A)^+ = V_N \Sigma_N^{-2} V_N^\top$, where Σ_N^{-1} denotes the $N \times N$ matrix whose elements are the inverse of the nonzero eigenvalues. After some computations equation (3.58) can be written as:

$$\mathbf{z}^0 = V\Sigma_N^{-1} U_N^\top \mathbf{t} = (V\Sigma_N U_N^\top)(U_N \Sigma_N^{-2} U_N^\top)\mathbf{t} = AG^{-1}\mathbf{t}. \tag{3.59}$$

Using the definition of $\mathbf{z}^0$ we have that

$$\sum_{n=1}^{\infty} w_n^0 \phi_n(\mathbf{x}) = \sum_{n=1}^{\infty} \sum_{i=1}^{N} \sqrt{\lambda_n}\phi_n(\mathbf{x}) A_{ni}\alpha_i. \tag{3.60}$$

Finally, using the definition of A_{in} we get:

$$\sum_{n=1}^{\infty} w_n^0 \phi_n(\mathbf{x}) = \sum_{i=1}^{N} K(\mathbf{x}, \mathbf{x}_i)\alpha_i$$

which completes the proof of the lemma.

Given this lemma, we consider only functions of the form $\sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x})$. We show that the function of this form that solves the system of equations (3.56) has norm larger than A^2. Therefore any other solution has norm larger than A^2 which implies we cannot shatter N points using functions of our hypothesis space.

The solution $\boldsymbol{\alpha} = G^{-1}\boldsymbol{t}$ needs to satisfy the constraint:

$$\boldsymbol{\alpha}^T G \boldsymbol{\alpha} = \boldsymbol{t}^T G^{-1} \boldsymbol{t} \leq A^2$$

Let λ_{max} be the largest eigenvalue of matrix G. Then $\boldsymbol{t}^T G^{-1}\boldsymbol{t} \geq \frac{\boldsymbol{t}^T\boldsymbol{t}}{\lambda_{max}}$. Since $\lambda_{max} \leq M^2$, $\boldsymbol{t}^T G^{-1}\boldsymbol{t} \geq \frac{\boldsymbol{t}^T\boldsymbol{t}}{M^2}$. Moreover, because of the choice of the separation, $\boldsymbol{t}^T\boldsymbol{t} \geq N\gamma^2$ (for example, for the points in class 1 which contribute to $\boldsymbol{t}^T\boldsymbol{t}$ an amount equal to $(y_i + s + \gamma + \delta_i)^2$: $|y_i| < s \Rightarrow y_i + s > 0$, and since $\gamma + \delta_i \geq \gamma > 0$, then $(y_i + s + \gamma + \delta_i)^2 \geq \gamma^2$. Similarly each of the other points "contribute" to $\boldsymbol{t}^T\boldsymbol{t}$ at least γ^2, so $\boldsymbol{t}^T\boldsymbol{t} \geq N\gamma^2$). So:

$$\boldsymbol{t}^T G^{-1}\boldsymbol{t} \geq \frac{N\gamma^2}{M^2} > A^2$$

since we assumed that $N > \frac{M^2 A^2}{\gamma^2}$. This is a contradiction, so we conclude that we cannot get this particular separation.

Proof of part 2.

Consider N points that can be shattered. This means that for any separation, for points in the first class there are $\delta_i \geq 0$ such that $|f(x_i) + b - y_i| = s + \gamma + \delta_i$. For points in the second class there are $\delta_i \in [0, s - \gamma]$ such that $|f(x_i) + b - y_i| = s - \gamma - \delta_i$. As in the case $b = 0$ we can remove the absolute values by considering for each class two types of points (we call them type 1 and type 2). For class 1, type 1 are points for which $f(x_i) = y_i + s + \gamma + \delta_i - b = t_i - b$. Type 2 are points for which $f(x_i) = y_i - s - \gamma - \delta_i - b = t_i - b$. For class 2, type 1 are points for which $f(x_i) = y_i + s - \gamma - \delta_i - b = t_i - b$. Type 2 are points for which $f(x_i) = y_i - s + \gamma + \delta_i - b = t_i - b$. Variables t_i are as in the case $b = 0$. Let $S_{11}, S_{12}, S_{-11}, S_{-12}$ denote the four sets of points (S_{ij} are points of class i type j). Using lemma 3.4.1, we only need to consider functions of the form $f(x) = \sum_{i=1}^N \alpha_i K(x_i, x)$. The coefficients α_i are given by $\boldsymbol{\alpha} = G^{-1}(\boldsymbol{t} - \boldsymbol{b})$ there $\boldsymbol{b}$ is a vector of b's. As in the case $b = 0$, the RKHS norm of this function is at least

$$\frac{1}{M^2}(\boldsymbol{t} - \boldsymbol{b})^T(\boldsymbol{t} - \boldsymbol{b}). \tag{3.61}$$

The b that minimizes (3.61) is $\frac{1}{N}(\sum_{i=1}^N t_i)$. So (3.61) is at least as large as (after replacing b and doing some simple calculations) $\frac{1}{2NM^2}\sum_{i,j=1}^N (t_i - t_j)^2$.

We now consider a particular separation. Without loss of generality assume that $y_1 \leq y_2 \leq \ldots \leq y_N$ and that N is even (if odd, consider $N - 1$ points). Consider the separation where class 1 consists only of the "even" points $\{N, N-2, \ldots, 2\}$. The following lemma holds:

Lemma 3.4.2 *For the separation considered,* $\sum_{i,j=1}^N (t_i - t_j)^2$ *is at least as large as* $\frac{\gamma^2(N^2-4)}{2}$.

Proof of lemma

Consider a point (x_i, y_i) in S_{11} and a point (x_j, y_j) in S_{-11} such that $y_i \geq y_j$ (if such a pair does not exist we can consider another pair from the cases listed below). For these points $(t_i - t_j)^2 = (y_i + s + \gamma + \delta_i - y_j - s + \gamma + \delta_j)^2 = ((y_i - y_j) + 2\gamma + \delta_i + \delta_j)^2 \geq 4\gamma^2$. In a similar way (taking into account the constraints on the δ_i's and on s) the inequality $(t_i - t_j)^2 \geq 4\gamma^2$ can be shown to hold in the following two cases:

$$\begin{array}{lll} (x_i, y_i) \in S_{11}, & (x_j, y_j) \in S_{-11} \bigcup S_{-12}, & y_i \geq y_j \\ (x_i, y_i) \in S_{12}, & (x_j, y_j) \in S_{-11} \bigcup S_{-12}, & y_i \leq y_j \end{array} \tag{3.62}$$

Moreover

$$\begin{array}{rl} \sum_{i,j=1}^N (t_i - t_j)^2 \geq & 2\left[\sum_{i \in S_{11}} \left(\sum_{j \in S_{-11} \bigcup S_{-12}, y_i \geq y_j} (t_i - t_j)^2\right)\right] + \\ & 2\left[\sum_{i \in S_{12}} \left(\sum_{j \in S_{-11} \bigcup S_{-12}, y_i \leq y_j} (t_i - t_j)^2\right)\right]. \end{array} \tag{3.63}$$

since in the right hand side we excluded some of the terms of the left hand side. Using the fact that for the cases considered $(t_i - t_j)^2 \geq 4\gamma^2$, the right hand side is at least

$$\begin{array}{l} 8\gamma^2 \sum_{i \in S_{11}} (\text{number of points j in class } -1 \text{ with } y_i \geq y_j) + \\ +8\gamma^2 \sum_{i \in S_{12}} (\text{number of points j in class } -1 \text{ with } y_i \leq y_j) \end{array} \tag{3.64}$$

Let I_1 and I_2 be the cardinalities of S_{11} and S_{12} respectively. Because of the choice of the separation it is clear that (3.64) is at least

$$8\gamma^2 \left((1 + 2 + \ldots + I_1)) + (1 + 2 + \ldots + (I_2 - 1))\right)$$

(for example if $I_1 = 2$ in the worst case points 2 and 4 are in S_{11} in which case the first part of (3.64) is exactly 1+2). Finally, since $I_1 + I_2 = \frac{N}{2}$, (3.64) is at least

$8\gamma^2 \frac{N^2-4}{16} = \frac{\gamma^2(N^2-4)}{2}$, which proves the lemma.

Using lemma 3.4.2 we get that the norm of the solution for the considered separation is at least as large as $\frac{\gamma^2(N^2-4)}{4NM^2}$. Since this has to be $\leq A^2$ we get that $N - \frac{4}{N} \leq 4\left[\frac{M^2A^2}{\gamma^2}\right]$, which completes the proof (assume $N > 4$ and ignore additive constants less than 1 for simplicity of notation).

In the case of L_p loss functions, using the same argument as in the proof of theorem 3.4.2 we get that the V_γ dimension in infinite dimensional RKHS is bounded by $\frac{(pB)^2M^2A^2}{\gamma^2}$ in the first case of theorem 3.4.3, and by $4\frac{(pB)^2M^2A^2}{\gamma^2}$ in the second case of theorem 3.4.3. Finally for L_ϵ loss functions the bound on the V_γ dimension is the same as that for L_1 loss function, again using the argument in the proof of theorem 3.4.2. □

Empirical V_γ dimension

Theorem 3.4.3 assumes a bound on the eigenvalues of *any* finite dimensional matrix G. However such a bound may not be known a priori, or it may not even exist. In practice we can still use the method presented above to measure the empirical V_γ dimension given a set of ℓ training points. This can provide an upper bound on the random entropy of our hypothesis space [Vapnik, 1998].

More precisely, given a set of ℓ training points we build the $\ell \times \ell$ matrix G as before, and compute it's largest eigenvalue $\lambda_{\max}$. We can then substitute M^2 with $\lambda_{\max}$ in the computation above to get an upper bound of what we call the empirical V_γ dimension. This can be used directly to get bounds on the random entropy (or number of ways that the ℓ training points can be separated using rules (3.48)) of our hypothesis space. Finally the statistical properties of our learning machine can be studied using the estimated empirical V_γ dimension (or the random entropy), in a way similar in spirit as in [Williamson *et al.*, 1998, Shawe-Taylor *et al.*, 1998]. We discuss this issue further in chapter 6.

3.4.3 A theoretical analysis of SVM classification

It is interesting to notice that a similar analysis can be used for the problem of classification. In this case the following theorem holds:

Theorem 3.4.4 *The V_γ dimension h for $|1-yf(\mathbf{x})|_+^\sigma$ in hypothesis spaces $\mathcal{H}_A = \{f \in \mathcal{H}\ ;\ \|f\|_K^2 \leq A^2\}$ (of the set of function $\{|1-yf(\mathbf{x})|_+^\sigma\ ;\ f \in \mathcal{H}_A\}$) and $y \in \{-1, 1\}$, is finite for $\forall\ 0 < \gamma$. If D is the dimensionality of the RKHS $\mathcal{H}$, and R^2 is the radius of the smallest sphere centered at the origin containing the data $\mathbf{x}$ in the RKHS, then h is upper bounded by:*

- *$O(min(D, \frac{R^2A^2}{\gamma^{\frac{2}{\sigma}}}))$ for $\sigma < 1$*
- *$O(min(D, \frac{\sigma^2R^2A^2}{\gamma^2}))$ for $\sigma \geq 1$*

Proof

The proof is based on the following theorem [Gurvits, 1997] (proved for the fat-shattering dimension, but as mentioned in chapter 2, we use it for the "equivalent" V_γ one).

Theorem [Gurvits, 1997]: The V_γ dimension h of the set of functions[7] $\mathcal{H}_A = \{f \in \mathcal{H} \; ; \; \|f\|_K^2 \leq A^2\}$ *is finite for* $\forall \; \gamma > 0$. *If D is the dimensionality of the RKHS, then* $h \leq O(min(D, \frac{R^2A^2}{\gamma^2}))$*, where R^2 is the radius of the smallest sphere in the RKHS centered at the origin where the data belong to.*

Let $2N$ be the largest number of points $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{2N}, y_{2N})\}$ that can be shattered using the rules:

$$\begin{array}{rl} \text{class } 1 \text{ if} & \theta(1 - y_i f(\mathbf{x}_i))(1 - y_i f(\mathbf{x}_i))^\sigma \geq s + \gamma \\ \text{class } -1 \text{ if} & \theta(1 - y_i f(\mathbf{x}_i))(1 - y_i f(\mathbf{x}_i))^\sigma \leq s - \gamma \end{array} \tag{3.65}$$

for some s with $0 < \gamma \leq s$. After some simple algebra these rules can be decomposed as:

$$\begin{array}{rl} \text{class } 1 \text{ if} & f(\mathbf{x}_i) - 1 \leq -(s+\gamma)^{\frac{1}{\sigma}} \text{ (for } y_i = 1 \text{)} \\ \text{or} & f(\mathbf{x}_i) + 1 \geq (s+\gamma)^{\frac{1}{\sigma}} \text{ (for } y_i = -1 \text{)} \\ \text{class } -1 \text{ if} & f(\mathbf{x}_i) - 1 \geq -(s-\gamma)^{\frac{1}{\sigma}} \text{ (for } y_i = 1 \text{)} \\ \text{or} & f(\mathbf{x}_i) + 1 \leq (s-\gamma)^{\frac{1}{\sigma}} \text{ (for } y_i = -1 \text{)} \end{array} \tag{3.66}$$

From the $2N$ points at least N are either all class -1, or all class 1. Consider the first case (the other case is exactly the same), and for simplicity of notation let's assume the first N points are class -1. Since we can shatter the $2N$ points, we can also shatter the first N points. Substituting y_i with 1, we get that we can shatter the N points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ using rules:

$$\begin{array}{rl} \text{class } 1 \text{ if} & f(\mathbf{x}_i) + 1 \geq (s+\gamma)^{\frac{1}{\sigma}} \\ \text{class } -1 \text{ if} & f(\mathbf{x}_i) + 1 \leq (s-\gamma)^{\frac{1}{\sigma}} \end{array} \tag{3.67}$$

Notice that the function $f(\mathbf{x}_i) + 1$ has RKHS norm bounded by A^2 plus a constant C (equal to the inverse of the eigenvalue corresponding to the constant basis function in the RKHS - if the RKHS does not include the constant functions, we can define a new RKHS with the constant and use the new RKHS norm). Furthermore there is a "margin" between $(s+\gamma)^{\frac{1}{\sigma}}$ and $(s-\gamma)^{\frac{1}{\sigma}}$ which we can lower bound as follows.

For $\sigma < 1$, assuming $\frac{1}{\sigma}$ is an integer (if not, we can take the closest lower integer),

$$\frac{1}{2}\left((s+\gamma)^{\frac{1}{\sigma}} - (s-\gamma)^{\frac{1}{\sigma}}\right) = \frac{1}{2}((s+\gamma) - (s-\gamma))\left(\sum_{k=0}^{\frac{1}{\sigma}-1}(s+\gamma)^{\frac{1}{\sigma}-1-k}(s-\gamma)^k\right) \geq$$

$$\geq \gamma\gamma^{\frac{1}{\sigma}-1} = \gamma^{\frac{1}{\sigma}}.$$

[7] In this case we can consider $V(y, f(\mathbf{x})) = f(\mathbf{x})$.

For $\sigma \geq 1$, σ integer (if not, we can take the closest upper integer) we have that:

$$2\gamma = \left((s+\gamma)^{\frac{1}{\sigma}}\right)^{\sigma} - \left((s-\gamma)^{\frac{1}{\sigma}}\right)^{\sigma} =$$

$$= ((s+\gamma)^{\frac{1}{\sigma}} - (s-\gamma)^{\frac{1}{\sigma}}) \left(\sum_{k=0}^{\sigma-1}((s+\gamma)^{\frac{1}{\sigma}})^{\sigma-1-k}((s-\gamma)^{\frac{1}{\sigma}})^{k}\right) \leq$$

$$\leq ((s+\gamma)^{\frac{1}{\sigma}} - (s-\gamma)^{\frac{1}{\sigma}})\sigma B^{\frac{\sigma-1}{\sigma}}$$

where B is an upper bound on the values of the loss function (which exists because of the constraints on $\mathbf{x}$ and $\|f\|_K^2$),

from which we obtain:

$$\frac{1}{2}\left((s+\gamma)^{\frac{1}{\sigma}} - (s-\gamma)^{\frac{1}{\sigma}}\right) \geq \frac{\gamma}{\sigma B^{\frac{\sigma-1}{\sigma}}} \tag{3.68}$$

Therefore N cannot be larger than the V_γ dimension of the set of functions with RKHS norm $\leq A^2 + C$ and margin at least $\gamma^{\frac{1}{\sigma}}$ for $\sigma < 1$ (from eq. (3.68)) and $\frac{\gamma}{\sigma B^{\frac{\sigma-1}{\sigma}}}$ for $\sigma \geq 1$ (from eq. (3.68)). Using Gurvits' theorem, and ignoring constant factors (also ones because of C) the theorem is proved. □

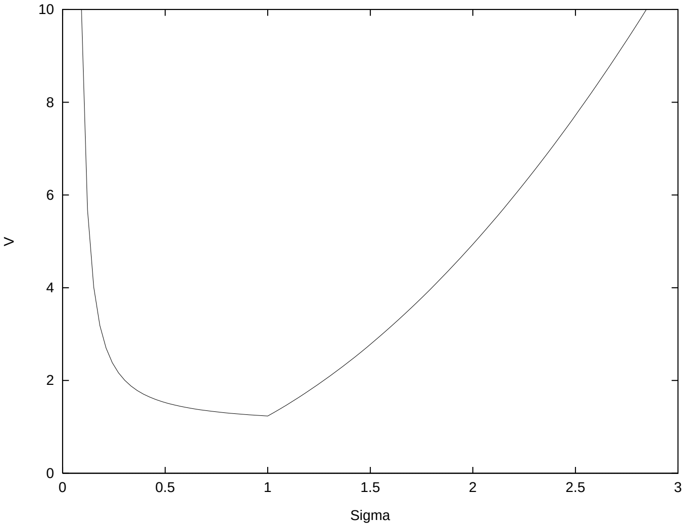

Figure 3-2: Plot of the V_γ dimension as a function of σ for $\gamma = .9$

Figure 3-2 shows the V_γ dimension for $R^2A^2 = 1$ and $\gamma = 0.9$, and D infinite. Notice that as $\sigma \rightarrow 0$, the dimension goes to infinity. For $\sigma = 0$ the V_γ dimension becomes the same as the VC dimension of hyperplanes, which is infinite for infinite dimensional RKHS [Vapnik, 1998]. For σ increasing above 1, the dimension also increases: intuitively the margin γ becomes smaller relatively to the values of the loss function.

We now study classification kernel machines of the form (3.2), namely:

$$\begin{aligned} &\min_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) \\ &\qquad\qquad \|f\|_K^2 \leq A_m^2 \end{aligned} \tag{3.69}$$

for particular loss functions V:

- Misclassification loss function:

$$V(y, f(\mathbf{x})) = V^{msc}(yf(\mathbf{x})) = \theta(-yf(\mathbf{x})) \tag{3.70}$$

- Hard margin loss function:

$$V(y, f(\mathbf{x})) = V^{hm}(yf(\mathbf{x})) = \theta(1 - yf(\mathbf{x})) \tag{3.71}$$

- Soft margin loss function:

$$V(y, f(\mathbf{x})) = V^{sm}(yf(\mathbf{x})) = |1 - yf(\mathbf{x})|_+, \tag{3.72}$$

where θ is the Heavyside function. Loss functions (3.71) and (3.72) are "margin" ones because the only case they do not penalize a point $(\mathbf{x}, y)$ is if $yf(\mathbf{x}) \geq 1$. For a given f, these are the points that are correctly classified *and* have distance $\frac{|f(\mathbf{x})|}{\|f\|_K^2} \geq \frac{1}{\|f\|_K^2}$ from the surface $f(\mathbf{x}) = 0$ (hyperplane in the feature space induced by the kernel K [Vapnik, 1998]). For a point $(\mathbf{x}, y)$, quantity $\frac{yf(\mathbf{x})}{\|f\|_K}$ is its margin, and the probability of having $\frac{yf(\mathbf{x})}{\|f\|_K} \geq \delta$ is called the *margin distribution* of hypothesis f. In the case of SVM Classification, quantity $|1 - y_i f(\mathbf{x}_i)|_+$ is known as the *slack variable* corresponding to training point $(\mathbf{x}_i, y_i)$ [Vapnik, 1998].

We will also consider the following family of margin loss functions (nonlinear soft margin loss functions):

$$V(y, f(\mathbf{x})) = V^{\sigma}(yf(\mathbf{x})) = |1 - yf(\mathbf{x})|_+^{\sigma}. \tag{3.73}$$

Loss functions (3.71) and (3.72) correspond to the choice of $\sigma = 0, 1$ respectively. Figure 3-3 shows some of the possible loss functions for different choices of the parameter σ.

We first study the loss functions (3.70) - (3.73). For classification machines the quantity we are interested in is the expected misclassification error of the solution f of machine 3.69. With some notation overload we note this with $I^{msc}[f]$. Similarly we will note with $I^{hm}[f]$, $I^{sm}[f]$, and $I^{\sigma}[f]$ the expected risks of f using loss functions (3.71), (3.72) and (3.73), respectively, and with $I_{emp}^{hm}[f;\ell]$, $I_{emp}^{sm}[f;\ell]$, and $I_{emp}^{\sigma}[f;\ell]$,

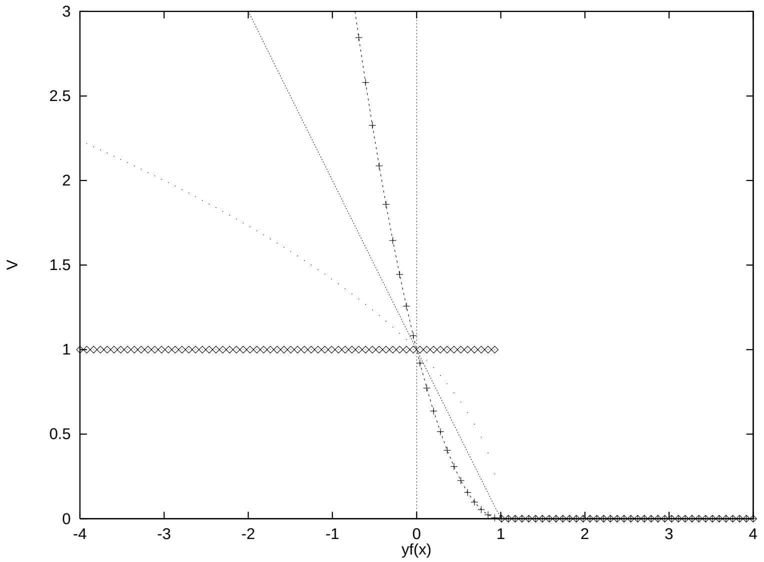

Figure 3-3: Hard margin loss (line with diamond-shaped points), soft margin loss (solid line), nonlinear soft margin with $\sigma = 2$ (line with crosses), and $\sigma = \frac{1}{2}$ (dotted line)

the corresponding empirical errors. We will not consider kernel machines (3.69) with V^{msc} as the loss function, for a clear reason: the solution of the optimization problem:

$$\begin{array}{rc} \min & \sum_{i=1}^{\ell} \theta(-y_i f(\mathbf{x}_i)) \\ \text{subject to} & \|f\|_K^2 \leq A^2 \end{array}$$

is independent of A, since for any solution f we can always rescale f and have the same cost $\sum_{i=1}^{\ell} \theta(-y_i f(\mathbf{x}_i))$.

Using theorems 3.4.4 and 2.1.6 we can bound the expected error of the solution f of machines (3.45):

$$Pr\left\{|I_{\text{emp}}[f;\ell] - I[f]| > \epsilon\right\} \leq \mathcal{G}(\epsilon, \ell, h_\gamma), \tag{3.74}$$

where the error is measured using either V^{sm} or V^{σ} with h_γ being the corresponding V_γ dimension given by theorem 3.4.4. To get a bound on the expected misclassification error $I^{msc}[f]$ we use the following simple observation:

$$V^{msc}(y, f(\mathbf{x})) \quad \leq \quad V^{\sigma}(y, f(\mathbf{x})) \;\; \text{for} \quad \forall\ \sigma, \tag{3.75}$$

So we can bound the expected misclassification error of the solution of machine (3.69) under V^{sm} and V^{σ} using the V_γ dimension of these loss functions and the empirical error of f measured using again these loss functions. In particular we get that for $\forall \sigma$, with probability $1 - \mathcal{G}(\epsilon, \ell, h_\gamma)$:

$$I^{msc}[f] \leq I^{\sigma}_{emp}[f;\ell] + \epsilon \tag{3.76}$$

where ϵ and γ are related as stated in theorem 2.1.6, and h_γ is the V_γ dimension given by theorem 3.4.4 for the V^{σ} loss function.

Unfortunately we cannot use theorems 3.4.4 and 2.1.6 for the V^{hm} or V^{msc} loss functions. For these loss functions, since they are binary-valued, the V_γ dimension is the same as the VC-dimension, which is not appropriate to use in this case: it is not influenced by A, and in the case that $\mathcal{H}$ is an infinite dimensional RKHS, the VC-dimension of these binary-valued loss functions turns out to be infinite (see for example [Vapnik, 1998]). This implies that for infinite dimensional RKHS, since the VC-dimension of the indicator functions V^{hm} and V^{msc} is infinite *no uniform convergence takes place*, and furthermore the bounds of chapter 2 cannot be used to bound the expected misclassification (or hard margin) error in terms of the empirical misclassification (or hard margin) one.

Notice, however, that for $\sigma \to 0$, V^σ approaches V^{hm} pointwise (from theorem 3.4.4 the V_γ dimension also increases towards infinity). Regarding the empirical error, this implies that $R^\sigma \to R^{hm}$, so, theoretically, we can still bound the misclassification error of the solution of machines with V^{hm} using:

$$R^{msc}(f) \leq R^{hm}_{emp}(f) + \epsilon + \max(R^\sigma_{emp}(f) - R^{hm}_{emp}(f), 0), \tag{3.77}$$

where $R^\sigma_{emp}(f)$ is measured using V^σ for some σ. Notice that changing σ we get a family of bounds on the expected misclassification error. Finally, as a last remark, it could be interesting to extend theorem 3.4.4 to loss functions of the form $g(|1 - yf(\mathbf{x})|_+)$, with g any continuous monotone function.

3.4.4 Discussion

In recent years there has been significant work on bounding the generalization performance of classifiers using scale-sensitive dimensions of real-valued functions out of which indicator functions can be generated through thresholding (see [Bartlett and Shawe-Taylor, 1998a, Shawe-Taylor and Cristianini, 1998, Shawe-Taylor *et al.*, 1998],[Bartlett, 1998] and references therein). This is unlike the "standard" statistical learning theory approach where classification is typically studied using the theory of indicator functions (binary valued functions) and their VC-dimension [Vapnik, 1998]. The approach taken in this chapter is similar in spirit with that of [Bartlett, 1998], but significantly different as we now briefly discuss.

In [Bartlett, 1998] a theory was developed to justify machines with "margin". The idea was that a "better" bound on the generalization error of a classifier can be derived by excluding training examples on which the hypothesis found takes a value close to zero (classification is performed after thresholding a real valued function). Instead of measuring the empirical misclassification error, as suggested by the standard statistical learning theory, what was used was the number of misclassified training points *plus* the number of training points on which the hypothesis takes a value close to zero. Only points classified correctly with some "margin" are considered correct. In [Bartlett, 1998] a different notation was used: the parameter A of machines (3.69) was fixed to 1, while a margin ψ was introduced inside the hard margin loss, i.e $\theta(\psi - yf(x))$. Notice that the two notations are equivalent: given a value A in our notation we have $\psi = A^{-1}$ in the notation of [Bartlett, 1998]. Below we adapt the results in [Bartlett, 1998] to the setup of this paper, that is, we set $\psi = 1$ and let A vary. Two main theorems were proven in [Bartlett, 1998].

Theorem 3.4.5 (Bartlett, 1998) *For a given A, with probability $1-\delta$, every function f with $\|f\|_K^2 \leq A^2$ has expected misclassification error $I^{msc}[f]$ bounded as:*

$$I^{msc}[f] < I_{emp}^{hm}[f;\ell] + \sqrt{\frac{2}{\ell}(dln(34e\ell/d)\log_2(578\ell) + ln(4/\delta)}, \tag{3.78}$$

where d is the fat-shattering dimension fat_γ of the hypothesis space $\{f : \|f\|_K^2 \leq A^2\}$ for $\gamma = \frac{1}{16A}$.

Unlike in this thesis, in [Bartlett, 1998] this theorem was proved without using theorem 2.1.6. Although practically both bound (3.78) and the bounds derived above are not tight and therefore not practical, bound (3.78) seems easier to use than the ones presented in this paper.

It is important to notice that, like bounds (3.74), (3.76), and (3.77), theorem 3.4.5 holds for a fixed A [Bartlett, 1998]. In [Bartlett, 1998] theorem 3.4.5 was extended to the case where the parameter A (or ψ in the notations of [Bartlett, 1998]) is not fixed, which means that the bound holds for all functions in the RKHS. In particular the following theorem gives a bound on the expected misclassification error of a machine that holds *uniformly* over all functions:

Theorem 3.4.6 (Bartlett, 1998) *For any f with $\|f\|_K < \infty$, with probability $1-\delta$, the misclassification error $I^{mcs}(f)$ of f is bounded as:*

$$I^{msc}[f] < I_{emp}^{hm}[f;\ell] + \sqrt{\frac{2}{\ell}(dln(34e\ell/d)\log_2(578\ell) + ln(8\|f\|/\delta)}, \tag{3.79}$$

where d is the fat-shattering dimension fat_γ of the hypothesis space consisting of all functions in the RKHS with norm $\leq \|f\|_K^2$, and with $\gamma = \frac{1}{32\|f\|}$.

Notice that the only differences between (3.78) and (3.79) are the $ln(8\|f\|/\delta)$ instead of $ln(4/\delta)$, and that $\gamma = \frac{1}{32\|f\|}$ instead of $\gamma = \frac{1}{16A}$.

So far we studied machines of the form (3.69), where A is fixed *a priori*. In practice learning machines used, like SVM, do not have A fixed a priori. For example in the case of SVM the problem is formulated [Vapnik, 1998] as minimizing:

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{\ell} |1 - y_i f(\mathbf{x}_i)|_+ + \lambda \|f\|_K^2 \tag{3.80}$$

where λ is known as the *regularization parameter*. In the case of machines (3.80) we do not know the norm of the solution $\|f\|_K^2$ before actually solving the optimization problem, so it is not clear what the "effective" A is. Since we do not have a fixed upper bound on the norm $\|f\|_K^2$ *a priori*, we **cannot** use the bounds of chapter 2 or theorem 3.4.5 for machines of the form (3.80). Instead, we need to use bounds that hold uniformly for *all* A (or ψ if we follow the setup of [Bartlett, 1998]), for example the bound of theorem 3.4.6, so that the bound also holds for the solution of (3.80) we find. In fact theorem 3.4.6 has been used directly to get bounds on the performance of SVM [Bartlett and Shawe-Taylor, 1998a]. It is a (simple) conjecture that a straightforward applications of the methods used to extend theorem 3.4.5 to 3.4.6 can also be used to extend the bounds of chapter 2 to the case where A is not fixed (and therefore hold for all f with $\|f\|_K^2 < \infty$).

There is another way to see the similarity between machines (3.69) and (3.80). Notice that the formulation (3.69) the regularization parameter λ of (3.80) can be seen as the *Lagrange multiplier* used to solve the constrained optimization problem (3.69). That is, problem (3.69) is equivalent to:

$$\max_{\lambda}\min_{f\in\mathcal{H}} \quad \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda(\|f\|_K^2 - A^2) \tag{3.81}$$

for $\lambda \geq 0$, which is similar to problem (3.80) that is solved in practice. However in the case of (3.81) the Lagrange multiplier λ is not known before having the training data, unlike in the case of (3.80).

So, to summarize, for the machines (3.2) studied in this thesis, A is fixed a priori and the "regularization parameter" λ is not known a priori, while for machines (3.80) the parameter λ is known a priori, but the norm of the solution (or the effective A) is not known a priori. As a consequence we can use the theorems of chapter 2 for machines (3.2) but not for (3.80). To do the second we need a technical extension of the results of chapter 2 similar to the extension of theorem 3.4.5 to 3.4.6 done in [Bartlett, 1998]. On the practical side, the important issue for both machines (3.2) and (3.80) is how to choose A or λ. In general the theorems and bounds discussed in chapter 2 cannot be practically used for this purpose. Criteria for the choice of the regularization parameter exist in the literature - such as cross validation and generalized cross validation - (for example see [Vapnik, 1998, Wahba, 1990] and references therein), and is the topic of ongoing research.

Finally, for the case of classification, as theorem 3.4.4 indicates, the generalization performance of the learning machines can be bounded using any function of the slack variables and therefore of the margin distribution. Is it, however, the case that the slack variables (margin distributions or any functions of these) are *the* quantities that control the generalization performance of the kernel machines, or there are other important geometric quantities involved? The next chapter follows a different approach to studying learning machines that leads to different type of bounds that are practically shown to be tight and also depend on parameters other than the margin distribution of the data.

Chapter 4

Learning with Ensembles of Kernel Machines

This chapter studies the problem of learning using ensembles of kernel machines, for the case of classification. Two types of ensembles are defined: voting combinations of classifiers, and adaptive combinations of classifiers. Special cases considered are bagging and Support Vector Machines. New theoretical bounds on the generalization performance of voting ensembles of kernel machines are presented. Experimental results supporting the theoretical bounds are shown. Finally, both voting and adaptive combinations of kernel machines are further characterized experimentally. Among others, the experiments suggest how such ensembles can be used to partially solve the problem of parameter selection for kernel machines (by combining machines with different parameters), and for fast training with very large datasets (by combining machines each trained on small subsets of the original training data).

4.1 Introduction

Two major recent advances in learning theory are Support Vector Machines (SVM) [Vapnik, 1998] and ensemble methods such as boosting and bagging [Breiman, 1996, Shapire *et al.*, 1998]. Distribution independent bounds on the generalization performance of these two techniques have been suggested recently [Vapnik, 1998, Shawe-Taylor and Cristianini, 1998, Bartlett, 1998, Shapire *et al.*, 1998], and similarities between these bounds in terms of a geometric quantity known as the *margin* (see chapter 3) have been proposed. More recently bounds on the generalization performance of SVM based on cross-validation have been derived [Vapnik, 1998, Chapelle and Vapnik, 1999]. These bounds depend also on geometric quantities other than the margin (such as the radius of the smallest sphere containing the support vectors).

The goal of this chapter is to study ensemble methods for the particular case of kernel machines. As in the previous chapter, the kernel machines considered are learning machines of the form (3.5), namely:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}_i, \mathbf{x}), \tag{4.1}$$

where, as discussed at the beginning of chapter 3, the coefficients c_i are learned by solving the following optimization problem:

$$\min_{f\in\mathcal{H}} \frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda\|f\|_K^2 \tag{4.2}$$

It turns out that for particular choices of the loss function V, the minimization problem (4.2) is equivalent to the *dual* one:

$$\begin{aligned} &\max_c H(c) = \textstyle\sum_{i=1}^{\ell} S(c_i) - \frac{1}{2}\sum_{i,j=1}^{\ell} c_i c_j y_i y_i K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{subject to}: \ 0 \le c_i \le C \end{aligned} \tag{4.3}$$

where $S(\cdot)$ is a cost function depending on V, and C a constant depending on V and λ [Jaakkola and Haussler, 1998a]. For example, in the particular case of SVM (that we are mostly interested in here), $S(c_i) = c_i$ and $C = \frac{1}{2\lambda}$ as discussed in chapter 3. The theoretical results presented in this chapter hold only for loss functions for which machines (4.2) and (4.3) are equivalent. Notice that the bias term (threshold b in the general case of machines $f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}_i, \mathbf{x}) + b$) is often incorporated in the kernel K (see also chapter 3 for a discussion on this issue).

The types of ensembles considered are linear combinations of the individual machines, that is, the "overall" machine $F(\mathbf{x})$ is of the form:

$$F(\mathbf{x}) = \sum_{t=1}^{T} \beta_t f^{(t)}(\mathbf{x}) \tag{4.4}$$

where T is the number of machines in the ensemble, and $f^{(t)}(\mathbf{x})$ is the t^{th} machine. Two types of ensembles are considered:

1. *Voting Combination of Classifiers* (VCC): this is the case where the coefficients β_t in 4.4 are not learned (i.e. $\beta_t = \frac{1}{T}$).

2. *Adaptive Combinations of Classifiers* (ACC): these are ensembles of the form (4.4) with the coefficients β_t also learned (adapted) from the training data.

The first part of the chapter presents new bounds on the generalization performance of voting ensembles of kernel machines (4.3). The bounds are derived using cross-validation arguments, so they can be seen as generalizations of the bounds for single kernel machines derived in [Vapnik, 1998, Jaakkola and Haussler, 1998a]. Particular cases considered are that of bagging kernel machines each trained on different subsamples of the initial training set, or that of voting kernel machines each using a different kernel (also different subsets of features/components of the initial input features). Among others, the bounds presented can be used, for example, for model selection in such cases.

For the case of adaptive combinations no bounds are proven (it remains an open question). However model selection can still be done by using a validation set. This chapter also shows experimental results showing that a validation set can be used for model selection for kernel machines and their ensembles, without having to decrease the training set size in order to create a validation set. Finally, both voting and adaptive combinations are further characterized experimentally.

4.1.1 Ensemble learning methods: Bagging and Boosting

Before presenting the theoretical and experimental analysis of ensembles of kernel machines, an overview of some common ensembles of learning machines is given briefly.

Combining machines instead of using a single one is an idea used by many researchers in recent years [Breiman, 1993, Breiman, 1996, Shapire *et al.*, 1998, Friedman *et al.*, 1998]. In the spirit of VCC, bagging [Breiman, 1996] is a particular ensemble architecture where a voting combination of a number of learning machines each trained using a subset with replacement of the initial training data is used. The size of the subsample is equal to the size of the original training set, but repetitions of points occur. Bagging has also been used for regression in which case the average real output of the individual regressors is taken.

Although there does not exist a well-accepted theory of bagging, there are some theoretical and experimental indications that bagging works better when the individual classifiers are "unstable" [Breiman, 1996]. Instability in this case means that the solution of the learning machine changes a lot with changes of the training data. A theoretical view of bagging developed by Breiman [Breiman, 1996] suggests that the overall performance of a "bagging system" depends on the instability of the individual classifiers (called *variance* in [Breiman, 1996]) as well as on the so called *bias* of the classifiers used, which is the error of the "best" possible classifier in the hypothesis space in which the individual classifiers belong to (i.e. the "best" decision tree, or linear classifier). According to the theory, bagging decreases the variance, therefore improving performance. This analysis of bagging is asymptotic, in the sense that the theory does not provide small-size characterizations of the performance of bagging, for example in the forms of bounds on the generalization error of bagging machines like the ones discussed in chapter 2.

A different type of ensembles of learning machines is that of boosted machines. Boosting [Shapire *et al.*, 1998, Friedman *et al.*, 1998] has been used to describe the following method of training and combining machines: each machine is trained on all the data, but each of the data points has a weight that signifies the "importance" that each of the machines gives to that point. Initially, that is for the first learning machine, all points have typically the same weight. After each machine is trained on the weighted data, the weights of the data are updated typically in such a way that future machines put more weight on the points that previous machines made a mistake on. After a number of such iterations the overall machine consists of a weighted vote of the individual machines, where the weights are computed typically in such a way that machines with large training error have small weights. Various "update" and "weight" rules have been proposed (see for example [Shapire *et al.*, 1998]). Theoretical analysis of boosting has lead to non-asymptotic bounds of the boosted (overall) machine of the form discussed in chapter 2 [Shapire *et al.*, 1998]. More recently, boosting has been studied in the context of gradient descent minimization of cost functions, where each of the individual machines (iterations) corresponds to a gradient descent step [Friedman *et al.*, 1998]. Within this framework new boosting methods have been proposed.

The approach taken in this chapter defers from both aforementioned ones. In particular, the theoretical analysis of ensembles of machines leads to bounds on the expected error of the ensembles (like in the case of boosting), but the bounds shown depend on quantities different from the ones in [Shapire *et al.*, 1998]. The bounds are derived using the leave-one-out approach similarly to [Vapnik, 1998, Jaakkola and Haussler, 1998a].

4.2 Generalization performance of voting ensembles

The theoretical results of the chapter are based on the cross-validation (or leave-one-out) error. The cross-validation procedure consists of removing from the training set one point at a time, training a machine on the remaining points and then testing on the removed one. The number of errors counted throughout this process, $\mathcal{L}\left((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell)\right)$, is called the cross-validation error [Wahba, 1980, Kearns *et al.*, 1995]. It is known that this quantity provides an estimate of the generalization performance of a machine [Wahba, 1980, Vapnik, 1998]. In particular the expectation of the generalization error of a machine trained using ℓ points is bounded by the expectation of the cross validation error of a machine trained on $\ell+1$ points (Luntz and Brailovsky theorem [Vapnik, 1998]).

We begin with some known results on the cross-validation error of kernel machines. The following theorem is from [Jaakkola and Haussler, 1998a]:

Theorem 4.2.1 *The cross-validation error of a kernel machine (4.3) is upper bounded as:*

$$\mathcal{L}\left((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell)\right) \leq \sum_{i=1}^{\ell} \theta(c_i K(\mathbf{x}_i, \mathbf{x}_i) - y_i f(\mathbf{x}_i)) \tag{4.5}$$

where θ is the Heavyside function, and the c_i are found by solving maximization problem (4.3).

In the particular case of SVM where the data are separable (4.5) can be bounded by geometric quantities, namely [Vapnik, 1998]:

$$\mathcal{L}\left((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell)\right) \leq \sum_{i=1}^{\ell} \theta(c_i K(\mathbf{x}_i, \mathbf{x}_i) - y_i f(\mathbf{x}_i)) \leq \frac{D_{sv}^2}{\rho^2} \tag{4.6}$$

where D_{sv} is the radius of the smallest sphere in the feature space induced by kernel K [Wahba, 1990, Vapnik, 1998] centered at the origin containing the support vectors, and ρ is the margin ($\rho^2 = \frac{1}{\|f\|_K^2}$) of the SVM.

Using this result, the following theorem is a direct application of the Luntz and Brailovsky theorem [Vapnik, 1998]:

Theorem 4.2.2 *The average generalization error of an SVM (with zero threshold b, and in the separable case) trained on ℓ points is upper bounded by*

$$\frac{1}{\ell+1} E\left(\frac{D_{sv(\ell)}^2}{\rho^2(\ell)}\right),$$

where the expectation E is taken with respect to the probability of a training set of size ℓ.

Notice that this result shows that the performance of SVM does not depend only on the margin, but also on other geometric quantities, namely the radius D_{sv}. In the non-separable case, it can be shown (the proof is similar to that of corollary 4.2.2 below) that equation (4.6) can be written as:

$$\mathcal{L}\left((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell)\right) \leq \sum_{i=1}^{\ell} \theta(c_i K(\mathbf{x}_i, \mathbf{x}_i) - y_i f(\mathbf{x}_i)) \leq EE_1 + \frac{D_{sv}^2}{\rho^2} \tag{4.7}$$

where EE_1 is the hard margin empirical error of the SVM (the number of training points with $yf(\mathbf{x}) < 1$, that is $\sum_{i=1}^{\ell} \theta(1 - y_i f(\mathbf{x}_i))$).

We now extend these results to the case of ensembles of kernel machines. We consider the general case where each of the machines in the ensemble uses a different kernel. Let T be the number of machines, and let $K^{(t)}$ be the kernel used by machine t. Notice that, as a special case, appropriate choices of $K^{(t)}$ lead to machines that may have different subsets of features from the original ones. Let $f^{(t)}(\mathbf{x})$ be the optimal solution of machine t (real-valued), and $c_i^{(t)}$ the optimal weight that machine t assigns to point $(\mathbf{x}_i, y_i)$ (after solving problem (4.3)). We consider ensembles that are linear combinations of the individual machines. In particular, the separating surface of the ensemble is:

$$F(\mathbf{x}) = \sum_{t=1}^{T} \beta_t f^{(t)}(\mathbf{x}) \tag{4.8}$$

and the classification is done by taking the sign of this function. The coefficients β_t are not learned (i.e. $\beta_t = \frac{1}{T}$), and $\sum_{t=1}^{T} \beta_t = 1$ (for scaling reasons), $\beta_t > 0$. All parameters (C's and kernels) are fixed before training. In the particular case of bagging, the subsampling of the training data should be deterministic. With this we mean that when the bounds are used for model (parameter) selection, for each model the same subsample sets of the data need to be used. These subsamples, however, are still random ones. It is a conjecture that the results presented below also hold (with minor modifications) in the general case that the subsampling is always random. We now consider the cross-validation error of such ensembles.

Theorem 4.2.3 *The cross-validation error $\mathcal{L}\left((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell)\right)$ of a kernel machines ensemble is upper bounded by:*

$$\mathcal{L}\left((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell)\right) \leq \sum_{i=1}^{\ell} \theta(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i)) \tag{4.9}$$

The proof of this theorem is based on the following lemma shown in [Vapnik, 1998, Jaakkola and Haussler, 1998a]:

Lemma 4.2.1 *Let c_i be the coefficient of the solution $f(\mathbf{x})$ of machine (4.3) corresponding to point $(\mathbf{x}_i, y_i)$, $c_i \neq 0$. Let $f_i(\mathbf{x})$ be the solution of machine (4.3) found when point $(\mathbf{x}_i, y_i)$ is removed from the training set. Then: $y_i f_i(\mathbf{x}_i) \geq y_i f(\mathbf{x}_i) - c_i K(\mathbf{x}_i, \mathbf{x}_i)$.*

Using lemma 4.2.1 we can now prove theorem 4.2.3.

Proof of theorem 4.2.3

Let $F_i(\mathbf{x}) = \sum_{t=1}^{T} \beta_t f_i^{(t)}(\mathbf{x})$ be the final machine trained with all initial training data except $(\mathbf{x}_i, y_i)$. Lemma 4.2.1 gives that

$$\begin{aligned} y_i F_i(\mathbf{x}_i) = y_i \sum_{t=1}^{T} \beta_t f_i^{(t)}(\mathbf{x}_i) \quad &\geq \quad y_i \sum_{t=1}^{T} \beta_t f^{(t)}(\mathbf{x}) - \sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) = \\ = y_i F(\mathbf{x}_i) - \sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \quad &\Rightarrow \quad \theta(-y_i F_i(\mathbf{x}_i)) \leq \theta(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i)) \end{aligned}$$

therefore the leave one out error $\sum_{i=1}^{\ell} \theta(-y_i F_i(\mathbf{x}_i))$ is not more than $\sum_{i=1}^{\ell} \theta(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i))$, which proves the theorem. □

Notice that the bound has the same form as bound (4.5): for each point $(\mathbf{x}_i, y_i)$ we only need to take into account its corresponding parameter $c_i^{(t)}$ and "remove" the effects of $c_i^{(t)}$ from the value of $F(\mathbf{x}_i)$.

The cross-validation error can also be bounded using geometric quantities. To this purpose we introduce one more parameter that we call the *ensemble margin* (in contrast to the margin of a single SVM). For each point $(\mathbf{x}_i, y_i)$ we define its ensemble margin to be simply $y_i F(\mathbf{x}_i)$. This is exactly the definition of margin in [Shapire *et al.*, 1998]. For any given $\delta > 0$ we define EE_δ to be the number of training points with ensemble margin $< \delta$ (empirical error with margin δ), and by N_δ the set of the remaining training points - the ones with ensemble margin $\geq \delta$. Finally, we note by $D_{t(\delta)}$ to be the radius of the smallest sphere in the feature space induced by kernel $K^{(t)}$ centered at the origin containing the points of machine t with $c_i^{(t)} \neq 0$ and ensemble margin larger than δ (in the case of SVM, these are the support vectors of machine t with ensemble margin larger than δ). A simple consequence of theorem 4.2.3 and of the inequality $K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \leq D_{t(\delta)}^2$ for points $\mathbf{x}_i$ with $c_i^{(t)} \neq 0$ and ensemble margin $y_i F(\mathbf{x}_i) \geq \delta$ is the following:

Corollary 4.2.1 *For any $\delta > 0$ the cross-validation error $\mathcal{L}((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell))$ of a kernel machines ensemble is upper bounded by:*

$$\mathcal{L}((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell)) \leq EE_\delta + \frac{1}{\delta} \left(\sum_{t=1}^{T} \beta_t D_{t(\delta)}^2 (\sum_{i \in N_\delta} c_i^{(t)}) \right) \tag{4.10}$$

Proof:

For each training point $(\mathbf{x}_i, y_i)$ with ensemble margin $y_i F(\mathbf{x}_i) < \delta$ we upper bound $\theta(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i))$ with 1 (this is a trivial bound). For the remaining points (the points in N_δ) we show that:

$$\theta(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i)) \leq \frac{1}{\delta} \left(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \right) \tag{4.11}$$

If $\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) < 0$ then (trivially):

$$\theta\left(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) \right) = 0 \leq \frac{1}{\delta} \sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i).$$

On the other hand, if $\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i) \geq 0$ then

$$\theta(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i)) = 1$$

while

$$\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \geq y_i F(\mathbf{x}_i) \geq \delta \Rightarrow \frac{1}{\delta} \sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) \geq 1.$$

So in both cases inequality (4.11) holds. Therefore:

$$\sum_{i=1}^{\ell} \theta(\sum_{t=1}^{T} \beta_t c_i^{(t)} K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) - y_i F(\mathbf{x}_i)) \leq EE_\delta + \frac{1}{\delta} \left(\sum_{i \in N_\delta} \sum_{t=1}^{T} \beta_t K^{(t)}(\mathbf{x}_i, \mathbf{x}_i) c_i^{(t)} \right) \leq$$
$$EE_\delta + \frac{1}{\delta} \left(\sum_{t=1}^{T} \beta_t D_{t(\delta)}^2 (\sum_{i \in N_\delta} c_i^{(t)}) \right)$$

which proves the corollary. □

Notice that equation (4.10) holds for any $\delta > 0$, so the best bound is obtained for the minimum of the right hand side with respect to $\delta > 0$. Using the Luntz and Brailovsky theorem, theorems 4.2.3 and 4.2.1 provide bounds on the generalization performance of general kernel machines ensembles like that of theorem 4.2.2.

We now consider the particular case of SVM ensembles. In this case, for example choosing $\delta = 1$ (4.10) becomes:

Corollary 4.2.2 *The leave-one-out error of an ensemble of SVMs is upper bounded by:*

$$\mathcal{L}\left((\mathbf{x}_i, y_1), \ldots (\mathbf{x}_\ell, y_\ell)\right) \leq EE_1 + \sum_{t=1}^{T} \beta_t \frac{D_t^2}{\rho_t^2} \tag{4.12}$$

where EE_1 is the margin empirical error with ensemble margin 1, D_t is the radius of the smallest sphere centered at the origin, in the feature space induced by kernel $K^{(t)}$, containing the support vectors of machine t, and ρ_t is the margin of SVM t.

This is because clearly $D_t \geq D_{t(\delta)}$ for any δ, and $\sum_{i \in N_\delta} c_i^{(t)} \leq \sum_{i=1}^{\ell} c_i^{(t)} = \frac{1}{\rho_t^2}$ (see [Vapnik, 1998] for a proof of this equality). A number of remarks can be made from equation (4.12).

First notice that the generalization performance of the SVM ensemble now depends on the "average" (convex combination of) $\frac{D^2}{\rho^2}$ of the individual machines. In some cases this may be smaller than the $\frac{D^2}{\rho^2}$ of a single SVM. For example, suppose we train many SVMs on different subsamples of the training points and we want to compare such an ensemble with a single SVM using all the points. If all SVMs (the single one, as well as the individual ones of the ensemble) use most of their training points as support vectors, then clearly the D^2 of each SVM in the ensemble is smaller than that of the single SVM. Moreover the margin of each SVM in the ensemble is expected to be larger than that of the single SVM using all the points. So the "average" $\frac{D^2}{\rho^2}$ in this case is expected to be smaller than that of the single SVM. Another case where an ensemble of SVMs may be better than a single SVM is the one where there are outliers among the training data: if the individual SVMs are trained on subsamples of the training data, some of the machines may have smaller $\frac{D^2}{\rho^2}$ because they do not use some outliers. In general it is not clear when ensembles of kernel machines are better than single machines. The bounds in this section may provide some insight to this question.

Notice also how the ensemble margin δ plays a role for the generalization performance of kernel machine ensembles. This margin is also shown to be important for boosting [Shapire *et al.*, 1998]. Finally, notice that all the results discussed hold for the case that there is no bias (threshold b), or the case where the bias is included in the kernel (as discussed in the introduction). In the experiments discussed below we use the results also for cases where the bias is not regularized, which is common in practice. It may be possible to use recent theoretical results [Chapelle and Vapnik, 1999] on the leave-one-out bounds of SVM when the bias b is taken into account in order to study the generalization performance of kernel machines ensembles with the bias b.

4.3 Experiments

To test how tight the bounds we presented are, we conducted a number of experiments using datasets from UCI[1], as well as the US Postal Service (USPS) dataset [LeCun *et al.*, 1989]. We show results for some of the sets in figures 4-1-4-5. For each dataset we split the overall set in training and testing (the sizes are shown in the figures) in 50 different (random) ways, and for each split:

1. We trained one SVM with $b = 0$ using all training data, computed the leave-one-bound given by theorem 4.2.1, and then compute the test performance using the test set.

2. We repeated (1) this time with with $b \neq 0$.

[1] Available from `http://www.ics.uci.edu/ mlearn/MLRepository.html`

3. We trained 30 SVMs with $b = 0$ each using a random subsample of size 40% of the training data (bagging), computed the leave-one-bound given by theorem 4.2.3 using $\beta_t = \frac{1}{30}$, and then compute the test performance using the test set.

4. We repeated (3) this time with with $b \neq 0$.

We then averaged over the 50 training-testing splits the test performances and the leave-one-out bounds found, and computed the standard deviations. All machines were trained using a Gaussian kernel $K(\mathbf{x}, \mathbf{y}) = e^{\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}}$, and we repeated the procedure for a number of different σ's of the Gaussian, and for a *fixed* C (show in the figures). We show the averages and standard deviations of the results in the figures. In all figures we use the following notation: top left figure: bagging with $b = 0$; top right figure: single SVM with $b = 0$; bottom left figure: bagging with $b \neq 0$; and bottom right figure: single SVM with $b \neq 0$. In all plots the solid line is the mean test performance and the dashed line is the error bound computed using the leave-one-out theorems (theorems 4.2.1 and 4.2.3). The dotted line is the validation set error discussed below. For simplicity, only one error bar (standard deviation over the 50 training-testing splits) is shown (the others were similar). The cost parameter C used is given in each of the figures. The horizontal axis is the natural logarithm of the σ of the Gaussian kernel used, while the vertical axis is the error.

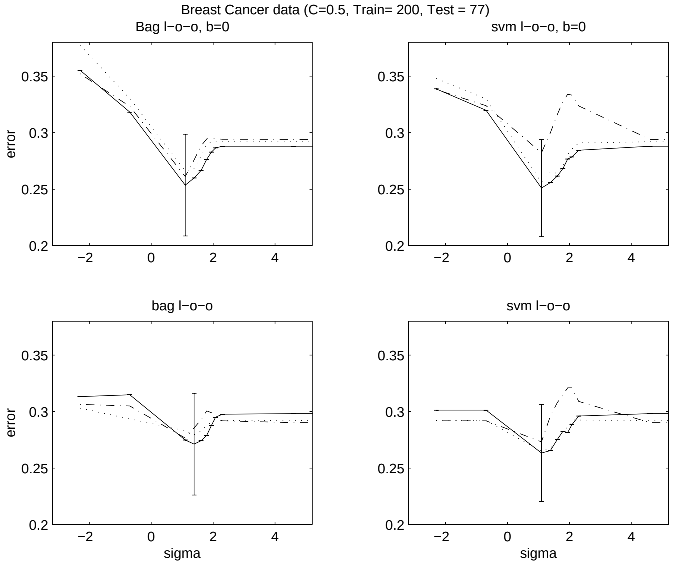

Figure 4-1: Breast cancer data: see text for description.

An interesting observation is that *the bounds are always tighter for the case of bagging than they are for the case of a single SVM.* This is an interesting experimental finding for which we do not have a theoretical explanation. It may be because the generalization performance of a machine is related to the *expected* leave-one-out error of the machine [Vapnik, 1998], and by combining many machines each using a different (random) subset of the training data we better approximate the "expected" leave-one-out than we do when we only compute the leave-one-out of a single machine. This

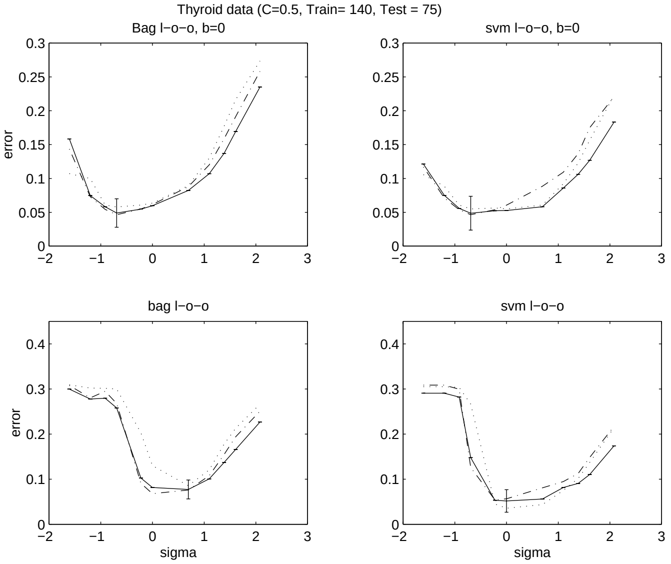

Figure 4-2: Thyroid data: see text for description.

finding can practically justify the use of ensembles of machines for model selection: parameter selection using the leave-one-out bounds presented in this chapter is easier for ensembles of machines than it is for single machines.

Another interesting observation is that the bounds seem to work similarly in the case that the bias b is not 0. In this case, as before, the bounds are tighter for ensembles of machines than they are for single machines.

Experimentally we found that the bounds presented here do not work well in the case that the C parameter used is large. An example is shown in figure 4-6. Consider the leave-one-out bound for a single SVM given by theorem 4.2.1. Let $(\mathbf{x}_i, y_i)$ be a support vector for which $y_i f(\mathbf{x}_i) < 1$. It is known [Vapnik, 1998] that for these support vectors the coefficient c_i is C. If C is such that $CK(\mathbf{x}_i, \mathbf{x}_i) > 1$ (for example consider Gaussian kernel with $K(\mathbf{x}, \mathbf{x}) = 1$ and any $C > 1$), the clearly $\theta(CK(\mathbf{x}_i, \mathbf{x}_i) - y_i f(\mathbf{x}_i)) = 1$. In this case the bound of theorem 4.2.1 effectively counts *all support vectors in the margin* (plus some of the ones *on* the margin - $yf(\mathbf{x}) = 1$). This means that for "large" C (in the case of Gaussian kernels this can be for example for any $C > 1$), the bounds of this chapter effectively are similar (not larger than) to another known leave-one-out bound for SVMs, namely one that uses the number of all support vectors to bound generalization performance [Vapnik, 1998]. So effectively the experimental results show that *the number of support vectors does not provide a good estimate of the generalization performance of the SVMs and their ensembles.*

4.4 Validation set for model selection

Instead of using bounds on the generalization performance of learning machines like the ones discussed above, an alternative approach for model selection is to use a validation set to choose the parameters of the machines. We consider first the simple

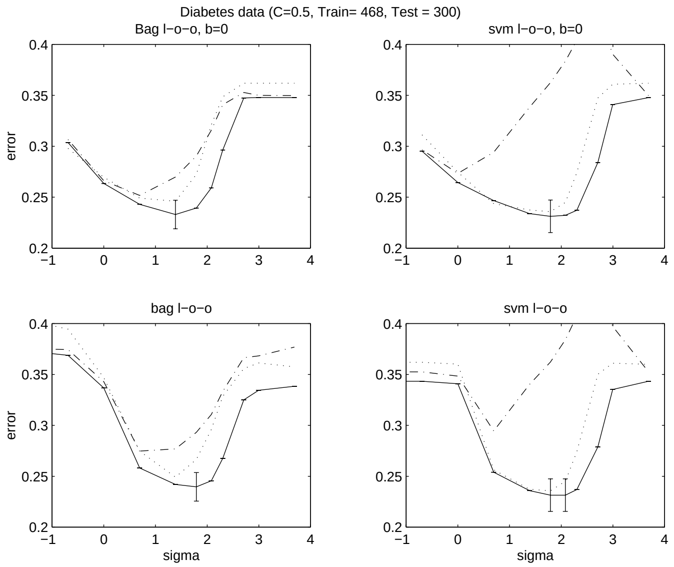

Figure 4-3: Diabetes data: see text for description.

case where we have N machines and we choose the "best" one based on the error they make on a fixed validation set of size V. This can be thought of as a special case where we consider as the hypothesis space to be the set of the N machines, and then we "train" by simply picking the machine with the smallest "empirical" error (in this case this is the validation error). It is known that if VE_i is the validation error of machine i and TE_i is its true test error, then for all N machines simultaneously the following bound holds with probability $1 - \eta$ [Devroye *et al.*, 1996, Vapnik, 1998]:

$$TE_i \leq VE_i + \sqrt{\frac{\log(N) - \log(\frac{\eta}{4})}{V}} \tag{4.13}$$

So how "accurately" we pick the best machine using the validation set depends, as expected, on the number of machines N and on the size V of the validation set. The bound suggests that a validation set can be used to accurately estimate the generalization performance of a relatively small number of machines (i.e. small number of parameter values examined), as done often in practice.

We used this observation for parameter selection for SVM and for their ensembles. Experimentally we followed a slightly different procedure from what is suggested by bound (4.13): for each machine (that is, for each σ of the Gaussian kernel in this case, both for a single SVM and for an ensemble of machines) we split the training set (for each training-testing split of the overall dataset as described above) into a smaller training set and a validation set (70-30% respectively). We trained each machine using the new, smaller training set, and measured the performance of the machine on the validation set. Unlike what bound (4.13) suggests, instead of comparing the validation performance found with the generalization performance of the machines trained on the smaller training set (which is the case for which bound (4.13) holds), we compared the validation performance with the test performance of the machine trained using *all* the initial (larger) training set. This way *we did not have to use less*

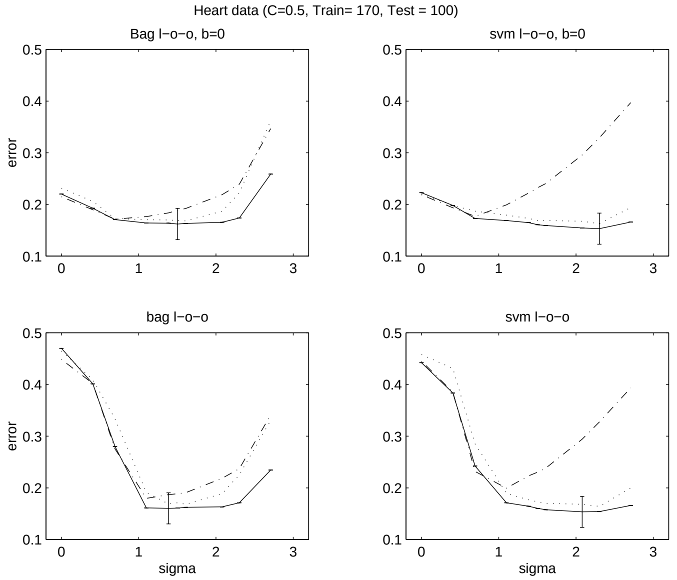

Figure 4-4: Heart data: see text for description.

points for training the machines, which is a typical drawback of using a validation set, and we could compare the validation performance with the leave-one-out bounds and the test performance of the *exact same* machines used in the previous section.

We show the results of these experiments in figures 4-1-4-5: see the dotted lines in the plots. We observe that *although the validation error is that of a machine trained on a smaller training set, it still provides a very good estimate of the test performance of the machines trained on the whole training set.* In all cases, including the case of $C > 1$ for which the leave-one-out bounds discussed above did not work well, the validation set error provided a very good estimate of the test performance of the machines.

4.5 Adaptive combinations of classifiers

The ensemble kernel machines (4.8) considered so far are voting combinations where the coefficients β_t in (4.8) of the linear combination of the machines are fixed. We now consider the case where these coefficients are also learned from the training subsets. In particular we consider the following architecture:

- A number T of kernel machines is trained as before (for example using different training data, or different parameters).

- The T outputs (real valued in the experiments, but could also be thresholded - binary) of the machines at each of the training points are computed.

- A linear machine (i.e. linear SVM) is trained using as inputs the outputs of the T machines on the training data, and as labels the original training labels.

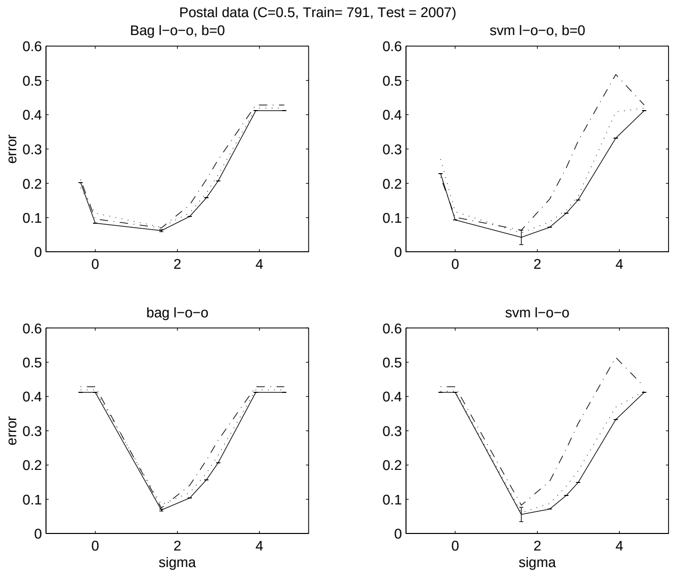

Figure 4-5: USPS data: see text for description.

> The solution is used as the coefficients β_t of the linear combination of the T machines.

Notice that for this type of machines the leave-one-out bound of theorem 4.2.3 does not hold since the theorem assumes fixed coefficients β_t. A validation set can still be used for model selection for these machines. On the other hand, an important characteristic of this type of ensembles is that independent of what kernels/parameters each of the individual machines of the ensemble use, the "second layer" machine (which finds coefficients β_t) uses always a linear kernel. This may imply that *the overall architecture may not be very sensitive to the kernel/parameters of the machines of the ensemble*. This hypothesis is experimentally tested by comparing how the test performance of this type of machines changes with the σ of the Gaussian kernel used from the individual machines of the ensemble, and compared the behavior with that of single machines and ensembles of machines with fixed β_t. Figure 4-7 shows two example. In the experiments, for all datasets except from one, learning the coefficients β_t of the combination of the machines using a linear machine (a linear SVM is used in the experiments) made the overall machine *less sensitive* to changes of the parameters of the individual machines (σ of the Gaussian kernel). This can be practically a useful characteristic of the architecture outlined in this section: for example the choice of the kernel parameters of the machines of the ensembles need not be tuned accurately.

4.6 Ensembles versus single machines

So far we concentrated on the theoretical and experimental characteristics of ensembles of kernel machines. We now discuss how ensembles compare with single machines.

Table 4.1 shows the test performance of one SVM compared with that of an ensemble of 30 SVMs combined with $\beta_t = \frac{1}{30}$ and an ensemble of 30 SVMs combined

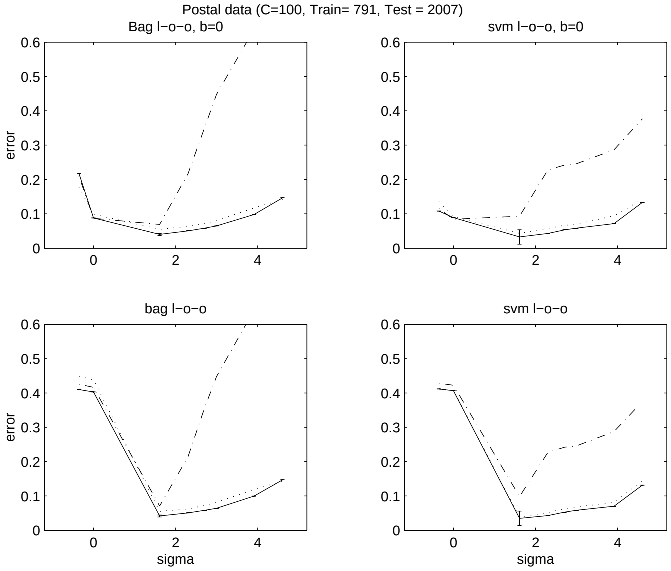

Figure 4-6: USPS data: using a large C (C=50). In this case the bounds do not work - see text for an explanation.

using a linear SVM for some UCI datasets (characteristic results). We only consider SVM and ensembles of SVMs with the threshold b. The table shows mean test errors and standard deviations for the best (decided using the validation set performance in this case) parameters of the machines (σ's of Gaussians *and* parameter C - hence different from figures 4-1-4-5 which where for a given C). As the results show, the best SVM and the best ensembles we found have about the same test performance. Therefore, with appropriate tuning of the parameters of the machines, combining SVM's does not lead to performance improvement compared to a single SVM.

Dataset	**SVM**	**VCC**	**ACC**
Breast	25.5 ± 4.3	25.6 ± 4.5	25 ± 4
thyroid	5.1 ± 2.5	5.1 ± 2.1	4.6 ± 2.7
diabetes	23 ± 1.6	23.1 ± 1.4	23 ± 1.8
heart	15.4 ± 3	15.9 ± 3	15.9 ± 3.2

Table 4.1: Average errors and standard deviations (percentages) of the "best" machines (best σ of the Gaussian kernel and best C) - chosen according to the validation set performances. The performances of the machines are about the same. VCC and ACC use 30 SVM classifiers.

Although the "best" SVM and the "best" ensemble (that is, after accurate parameter tuning) perform similarly, an important difference of the ensembles compared to a single machine is that the training of the ensemble consists of a large number of (parallelizable) small-training-set kernel machines - in the case of bagging. This implies that one can gain performance similar to that of a single machine by training many faster machines using smaller training sets. This can be an important practical

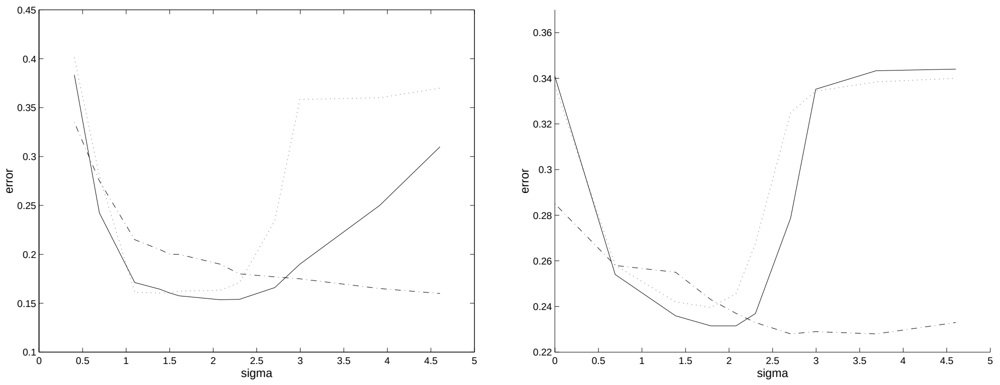

Figure 4-7: When the coefficients of the second layer are learned using a linear SVM the system is less sensitive to changes of the σ of the Gaussian kernel used by the individual machines of the ensemble. Solid line is one SVM, dotted is ensemble of 30 SVMs with fixed $\beta_t = \frac{1}{30}$, and dashed line is ensemble of 30 SVMs with the coefficients β_t learned. The horizontal axis shows the natural logarithm of the σ of the Gaussian kernel. Left is the heart dataset, and right is the diabetes one. The threshold b is non-zero for these experiments.

advantage of ensembles of machines especially in the case of large datasets. Table 4.2 compares the test performance of a single SVM with that of an ensemble of SVM each trained with as low as 1% of the initial training set (for one dataset). For fixed β_t the performance decreases only slightly in all cases (thyroid, that we show, was the only dataset found in the experiments for which the change was significant for the case of VCC), while in the case of ACC even with 1% training data the performance does not decrease: this is because the linear machine used to learn coefficients β_t uses all the training data. Even in this last case the overall machine can still be faster than a single machine, since the second layer learning machine is a linear one, and fast training methods for the particular case of linear machines exist [Platt, 1998]. Finally, it may be the case that ensembles of machines perform better [Osuna *et al.*, 1997b] for some problems in the presence of outliers (as discussed

Dataset	**VCC 10%**	**VCC 5%**	**VCC 1%**	**SVM**
Diabetes	23.9	26.2	-	23 ± 1.6
Thyroid	6.5	22.2	-	5.1 ± 2.5
Faces	.2	.2	.5	.1
Dataset	**ACC 10%**	**ACC 5%**	**ACC 1%**	**SVM**
Diabetes	24.9	24.5	-	23 ± 1.6
Thyroid	4.6	4.6	-	5.1 ± 2.5
Faces	.1	.2	.2	.1

Table 4.2: Comparison between error rates of a single SVM v.s. error rates of VCC and ACC of 100 SVMs for different percentages of subsampled data. The last dataset is from (Osuna *et al.*, 1997).

above), or, if the ensemble consists of machines that use different kernels and/or different input features, in the presence of irrelevant features. The leave-one-out bounds presented in this chapter may be used for finding these cases and for better understanding how bagging and general ensemble methods work [Breiman, 1996, Shapire *et al.*, 1998].

4.7 Summary

This chapter presented theoretical bounds on the generalization error of voting ensembles of kernel machines. The results apply to the quite general case where each of the machines in the ensemble is trained on different subsets of the training data and/or uses different kernels or input features. Experimental results supporting the theoretical findings have been shown. A number of observations have been made from the experiments:

1. The leave-one-out bounds for ensembles of machines have a form similar to that of single machines. In the particular case of SVMs, the bounds are based on an "average" geometric quantity of the individual SVMs of the ensemble (average margin and average radius of the sphere containing the support vectors).

2. The leave-one-out bounds presented are experimentally found to be tighter than the equivalent ones for single machines.

3. For SVM, the leave-one-out bounds based on the number of support vectors are experimentally found not to be tight.

4. It is experimentally found that a validation set can be used for accurate model selection without having to decrease the size of the training set used in order to create a validation set.

5. With accurate parameter tuning (model selection) single SVMs and ensembles of SVMs perform similarly.

6. Ensembles of machines for which the coefficients of combining the machines are also learned from the data are less sensitive to changes of parameters (i.e. kernel) than single machines are.

7. Fast (parallel) training without significant loss of performance relatively to single whole-large-training-set machines can be achieved using ensembles of machines.

A number of questions and research directions are open. An important theoretical question is how the bounds presented in this chapter can be used to better characterize ensembles of machines such as bagging [Breiman, 1996]. On the practical side, further experiments using very large datasets are needed to support the experimental finding that the ensembles of machines can be used for fast training without significant loss in performance. Finally, other theoretical questions are how to extend the bounds or VCC to the case of ACC, and how to use the more recent leave-one-out bounds for SVM [Chapelle and Vapnik, 1999] to better characterize the performance of ensembles of machines.

Chapter 5

Object Detection: a Case Study on Representations for Learning

Two important choices when using a kernel machine are that of the data representation and of the kernel of the machine. These choices are clearly problem specific, and a general method for making them is unlikely to exist. This chapter discusses the issues of data representation and kernel selection for the particular problem of object detection in images. It presents experimental comparisons of various image representations for object detection using kernel classifiers. In particular it discusses the use of support vector machines (SVM) for object detection using as image representations raw pixel values, projections onto principal components, and Haar wavelets. General linear transformations of the images through the choice of the kernel of the SVM are considered. Experiments showing the effects of histogram equalization, a non-linear transformation, are presented. Image representations derived from probabilistic models of the class of images considered, through the choice of the kernel of the SVM, are also evaluated. Finally, the chapter presents a feature selection heuristic using SVMs.

5.1 Introduction

Detection of real-world objects in images, such as faces and people, is a problem of fundamental importance in many areas of image processing: for face recognition, the face must first be detected before being recognized; for autonomous navigation, obstacles and landmarks need to be detected; effective indexing into image and video databases relies on the detection of different classes of objects. The detection of objects poses challenging problems: the objects are difficult to model, there is significant variety in color and texture, and the backgrounds against which the objects lie are unconstrained.

This chapter considers a learning based approach to object detection and focuses on the use of Support Vector Machines (SVM) classifiers [Vapnik, 1998] as the core engine of these systems [Papageorgiou *et al.*, 1998b]. A major issue in such a system is choosing an appropriate image representation. This chapter presents experimental results comparing different image representations for object detection, in particular for detecting faces and people.

Initial work on object detection used template matching approaches with a set of rigid templates or handcrafted parameterized curves, Betke & Makris[Betke and Makris, 1995], Yuille, et al.[Yuille *et al.*, 1992]. These approaches are difficult to extend to more complex objects such as people, since they involve a significant amount of prior information and domain knowledge. In recent research the detection problem has been solved using learning-based techniques that are data driven. This approach was used by Sung and Poggio[Sung and Poggio, 1994] and Vaillant, et al. [Vaillant *et al.*, 1994] for the detection of frontal faces in cluttered scenes, with similar architectures later used by Moghaddam and Pentland [Moghaddam and Pentland, 1995], Rowley, et al.[Rowley *et al.*, 1995], and Osuna et al.[Osuna *et al.*, 1997b]. The image representations used were either projections onto principal components (i.e. eigenfaces [Moghaddam and Pentland, 1995]), projections on wavelets, raw pixel values, or, finally, features derived from probabilistic models [Sung and Poggio, 1994]. This chapter compares these representations, and also links them through the choice of kernels in the SVM classifier.

5.2 A trainable system for object detection

The trainable system for object detection used in this chapter is based on [Papageorgiou *et al.*, 1998b] and can be used to learn any class of objects. The overall framework has been motivated and successfully applied in the past [Papageorgiou *et al.*, 1998b]. The system consists of three parts:

- A set of (positive) example images of the object class considered (i.e. images of frontal faces) and a set of negative examples (i.e. any non-face image) are collected.
- The images are transformed into vectors in a chosen representation (i.e. a vector of the size of the image with the values at each pixel location - below this is called the “pixel” representation).
- The vectors (examples) are used to train a SVM classifier to learn the classification task of separating positive from negative examples. A new set of examples is used to test the system. The full architecture involves scanning an (test) image over different positions and scales. [Papageorgiou *et al.*, 1998b] has more information.

Two choices need to be made: the representation in the second stage, and the kernel of the SVM (see below) in the third stage. This chapter focuses on these two choices.

In the experiments described below, the following data have been used:

- For the face detection systems, 2,429 positive images of frontal faces of size 19x19 (see figure 5-1), and 13,229 negative images randomly chosen from large images not containing faces have been used. The systems were tested on new data consisting of 105 positive images and around 4 million negative ones.
- For the people detection systems, 700 positive images of people of size 128x64 (see figure 5-1), and 6,000 negative images have been used. The systems were

tested on new data consisting of 224 positive images and 3,000 negative ones (for computational reasons, only for Figure 5-6 more test points have been used: 123 positive and around 800,000 negative ones).

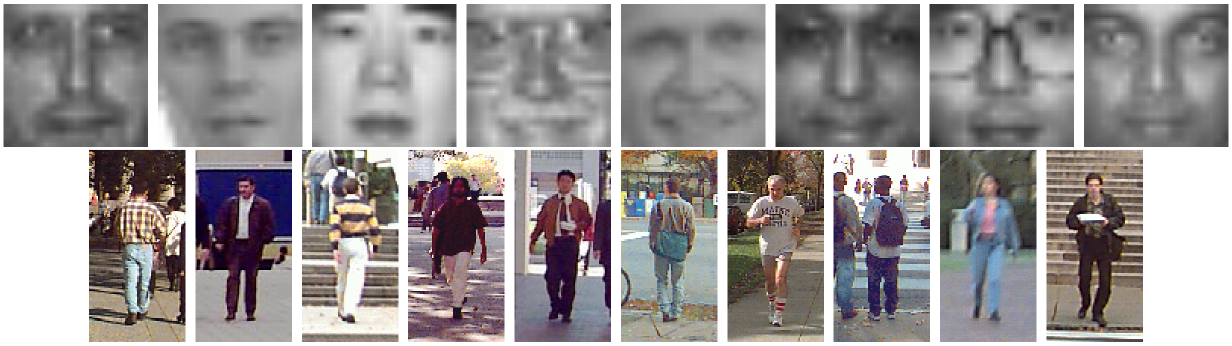

Figure 5-1: *Top row:* examples of images of faces in the training database. The images are 19x19 pixels in size. *Bottom row:* examples of images of people in the training database. The images are 128x64 pixels in size.

The performances are compared using ROC curves [Papageorgiou *et al.*, 1998b] generated by moving the hyperplane of the SVM solution by changing the threshold b (see below), and computing the percentage of false positives and false negatives for each choice of b. In the plots presented the vertical axis shows the percentage of positive test images correctly detected (1 - false negative percentage), while the horizontal axis shows one false positive detection per number of negative test images correctly classified.

5.3 Comparison of representations for face and people detection

5.3.1 Pixels, principal components and Haar wavelets

Using the experimental setup described above, experiments to compare the discriminative power of three different image representations have been conducted:

- The *pixel representation*: train an SVM using the raw pixel values scaled between 0 and 1 (i.e. for faces this means that the inputs to the SVM machine are $19 \cdot 19 = 361$ dimensional vectors with values from 0 to 1 - before scaling it was 0 to 255).

- The *eigenvector (principal components) representation:* compute the correlation matrix of the positive examples (their pixel vectors) and find its eigenvectors. Then project the pixel vectors on the computed eigenvectors. We can either do a full rotation by taking the projections on all 361 eigenvectors, or use the projections on only the first few principal components. The projections are rescaled to be between 0 and 1.

- The *wavelet representation:* consider a set of Haar wavelets at different scales and locations (see Figure 5-2), and compute the projections of the image on the chosen wavelets. For the face detection experiments all wavelets (horizontal, vertical and diagonal) at scales 4×4 and 2×2 have been used, since their dimensions correspond to typical features for the size of the face images considered. This gives a total of 1,740 coefficients for each image. For the people detection system wavelets at scales 32×32 and 16×16 shifted by 8 and 4 pixels respectively have been considered. This gives a total of 1,326 coefficients. The outputs of the projections were rescaled to be between 0 and 1.

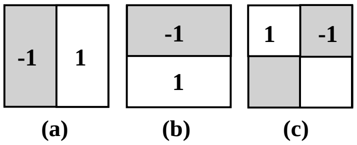

Figure 5-2: The 3 types of 2-dimensional non-standard Haar wavelets; (a) “vertical”, (b) “horizontal”, (c) “diagonal”.

Experiments

Figure 5-3 shows the results of the experiments comparing the representations described above. In all these experiments a second order polynomial kernel was used. The motivation for using such a kernel is based on the experimental results of [Osuna *et al.*, 1997a, Papageorgiou *et al.*, 1998b]. Throughout the chapter, notice the range of the axis in all the plots in the figures: the range varies in order to show clearer the important parts of the curves.

These experiments suggest a few remarks. First notice that both the pixel and eigenvector representations give almost identical results (small differences due to the way the ROC curves are produced are ignored). This is an observation that has a theoretical justification discussed below. Second, notice that for faces the wavelet representation performs about the same as the other two, but in the case of people, the wavelet representation is significantly better than the other two. This is a finding that was expected [Papageorgiou *et al.*, 1998b, Oren *et al.*, 1997]: for people pixels may not be very informative (i.e. people may have different color clothes), while wavelets capture intensity differences that discriminate people from other patterns [Papageorgiou *et al.*, 1998b]. On the other hand, for faces at the scale used, pixel values seem to capture enough of the information that characterizes faces. Notice that all the three representations considered so far are linear transformations of the pixels representation. This takes us to the next topic.

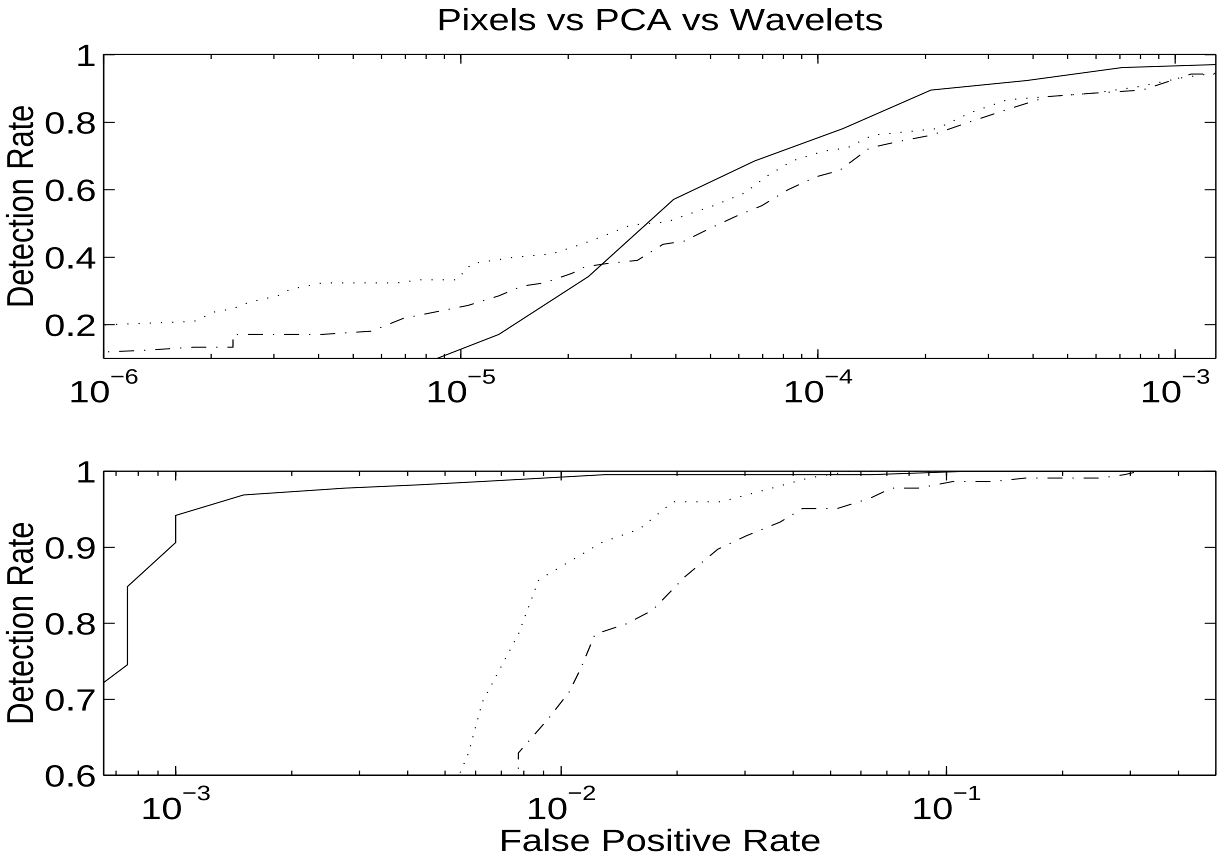

Figure 5-3: ROC curves for face (top) and people (bottom) detection: solid lines are for the wavelet representation, dashed lines for pixel representation, and dotted line for eigenvector representation (all 361 eigenvectors).

5.3.2 Linear transformations and kernels

As presented in chapter 3, a key issue when using a SVM (and generally any kernel machine) is the choice of the kernel K in equation (3.5), namely:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i), \tag{5.1}$$

The kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ defines a dot product between the projections of two inputs $\mathbf{x}_i, \mathbf{x}_j$, in a feature space (see chapter 3). Therefore the choice of the kernel is very much related to the choice of the "effective" image representation. So we can study different representations of the images through the study of different kernels for SVM.

In particular there is a simple relation between linear transformations of the original images, such as the ones considered above, and kernels. A point (image) $\mathbf{x}$ is linearly decomposed in a set of features $\mathbf{g} = g_1, \ldots, g_m$ by $\mathbf{g} = A\mathbf{x}$, with A a real matrix (we can think of the features $\mathbf{g}$ as the result of applying a set of linear filters to the image $\mathbf{x}$). If the kernel used is a polynomial of degree m[1] (as in the experiments),

[1] Generally this holds for any kernel for which only dot products between input arguments are

then $K(\mathbf{x}_i, \mathbf{x}_j) = (1+\mathbf{x}_i^\top \cdot \mathbf{x}_j)^m$, while $K(\mathbf{g}_i, \mathbf{g}_j) = (1+\mathbf{g}_i^\top \cdot \mathbf{g}_j)^m = (1+\mathbf{x}_i^\top (A^\top A)\mathbf{x}_j)^m$. So using a polynomial kernel in the "**g**" representation is the same as using a kernel $(1 + \mathbf{x}_i^\top (A^\top A)\mathbf{x}_j)^m$ in the original one. This implies that one can consider any linear transformation of the original images by choosing the appropriate square matrix $A^T A$ in the kernel K of the SVM.

As a consequence of this observation, we have a theoretical justification of why the pixel and eigenvector representations lead to the same performance: in this case the matrix A is orthonormal, therefore $A^T A = I$ which implies that the SVM finds the same solution in both cases. On the other hand, if we choose only some of the principal components (like in the case of eigenfaces [Turk and Pentland, 1991]), or if we project the images onto a non-orthonormal set of Haar wavelets, the matrix A is no longer orthonormal, so the performance of the SVM may be different.

5.3.3 Histogram Equalization

We now discuss the experimental finding that histogram equalization (H.E.), a non-linear transformation of the "pixel" representation, improves the performance of the detection system. Given an image, H.E. is performed in two steps: first the pixel values (numbering 0 to 255) are grouped into the smallest number of bins so that the distribution of the number of pixels in the image is uniform among the bins; then we replace the pixel values of the original image with the values (rank) of the bins they fall into. More information on H.E. can be found in the literature (i.e. [Jain, 1989]). Figure 5-5 shows an image of a face used for training, and the same face after H.E.

Experiments with the object detection system using the aforementioned representations have been conducted, this time after performing H.E. on every input image. Only for the wavelet representation, instead of projecting histogram equalized images on the chosen wavelets, we transformed the outputs of the projections of the original images on the wavelets using a sigmoid function. This operation is (almost) equivalent to first performing H.E. and then projecting on the wavelet filters the histogram equalized image (assuming Gaussian-like histogram of the original image). Figure 5-4 shows the performance of the detection system. Both for face and people detection the performance increased dramatically.

H.E. has been extensively used for image compression and in this case it is straightforward to show that H.E. is a form of Vector Quantization [Gersho and Gray, 1991] and is an effective coding scheme. Classification is however different from compression and it is an open question of why H.E. seems to improve so much the performance of the SVM classifier. Here is a conjecture:

Suppose that a transformation satisfies the following two conditions:

- it is a legal transformation of the input vector, that is it preserves the class label;
- it increases the entropy of the input set, leading to a more compressed representation.

needed - i.e. also for Radial Basis Functions.

It is a *conjecture* that such a transformation will improve the performance of a SVM classifier.

Notice that H.E. is a transformation of the images that satisfies the above conditions: i.e. faces remain faces, and non-faces remain non-faces (of course one can design images where this does not hold, but such images are very unlikely and are not expected to exist among the ones used for training and/or testing). Moreover H.E. leads to a more compressed representation of the images (in terms of bits needed to describe them). Of course the first condition relies on prior information. In the case of H.E. applied to images it is known a priori that H.E. is a transformation embedding "illumination invariance": images of the same face under different illuminations can be mapped into the same vector under H.E. Thus performing an H.E. transformation is roughly equivalent to using a larger training set containing many "virtual examples" generated from the real examples [Girosi and Chan, 1995, Vetter *et al.*, 1994] by changing the global illumination (mainly the dynamic range). Of course a larger training set in general improves the performance of a classifier. So this may explain the improvement of the system.

In the case of the SVM classifier used, it is likely that H.E. makes the space of images "more discrete": there are fewer possible images. This may correspond to a better (more uniform) geometry of the training data (for example, after H.E. there may be a larger margin between the two classes) that leads to a better separating surface found by the SVM.

Thus the conjecture claims that H.E. improves classification performance because it does not change, say, faces into non-faces: this is "a priori" information about the illumination invariance of face images. H.E. exploits this information. One may be able to formalize this either through a compression argument, or through the equivalence with virtual face examples, or possibly through a geometric argument. This "a priori" information is true for face and people images, but may not hold for other ones, in which case H.E. might not improve performance. In general, because of the same arguments outlined above, it is expected that any transformation of the original images that "effectively" takes advantage of prior information about the class of images considered and compresses their signatures, is expected to improve the performance of the system.

5.4 Input feature selection using SVM

This section addresses the following questions: can feature selection improve performance of the SVM classifier? can SVM perform well even when many (possibly irrelevant) features are used? One important problem with features selection is the multiple use of the data: the same training set is used first to train the system using all the features, then to select the important features, and finally to retrain the system using only the selected features. The multiple use of training data may lead to overfitting, so it is unclear a priori that selecting features can improve performance.

In order to investigate these issues, several experiments where the object detection systems were trained with different numbers of input features have been performed.

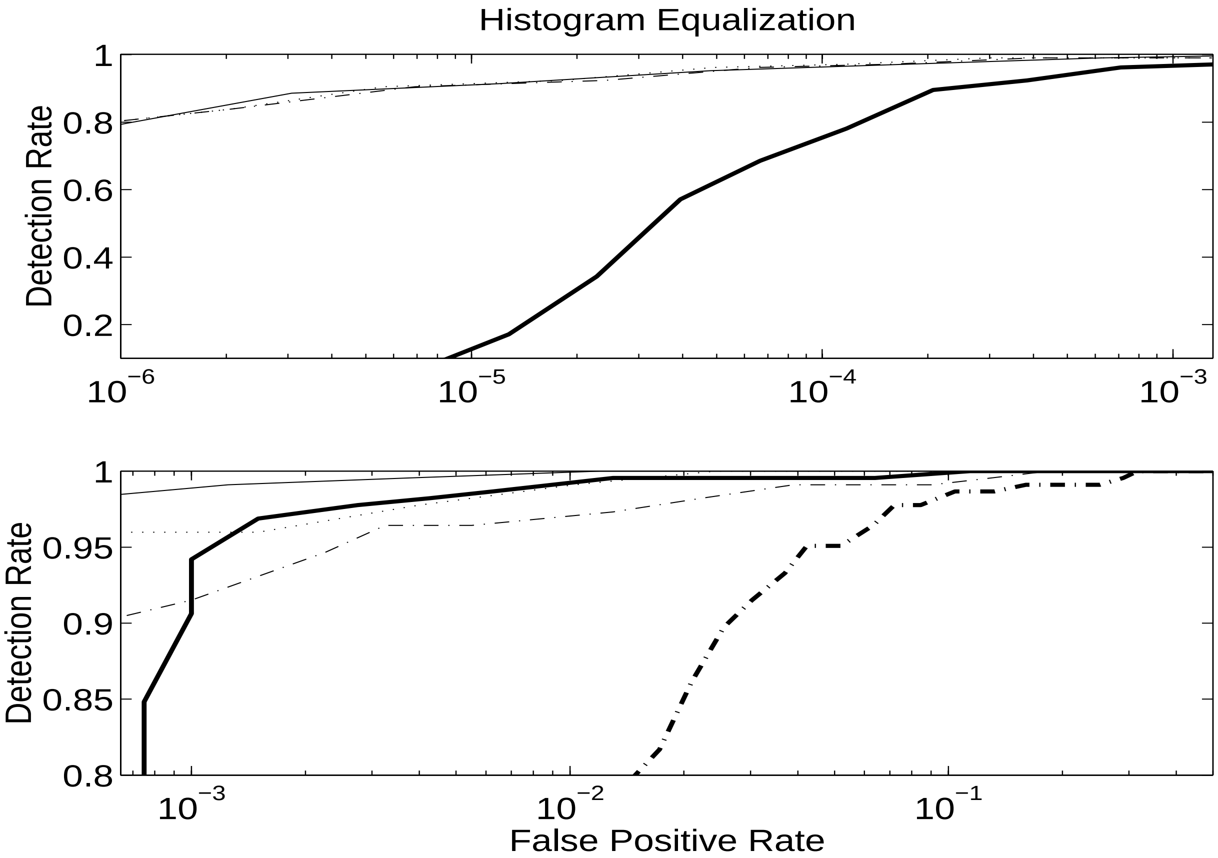

Figure 5-4: ROC curves for face (top) and people (bottom) detection after histogram equalization: solid lines are for the wavelet representation, dashed lines for the pixel representation, and dotted line for the eigenvector representation. The ROC curve for the wavelet representation without histogram equalization (like in Figure 5-3) is also shown; this is the bottom thick solid line. For people, the bottom thick dashed line shows the performance of pixels without H.E..

To this purpose a method for automatically selecting a subset of the input features within the framework of SVM has been developed.

5.4.1 A heuristic for feature selection using SVM

The idea of the proposed feature selection method is based on the intuition that the most important input features are the ones for which, if removed or modified, the separating boundary $f(\mathbf{x}) = 0$ changes the most. Instead of the change of the boundary we can consider the average change of the value of the function $f(\mathbf{x})$ in a region around the boundary (variations of f will affect classification only for points near the boundary). To do so, we compute the derivative of $f(\mathbf{x})$ with respect to an input feature x^r and integrate the absolute value (we are interested in the magnitude

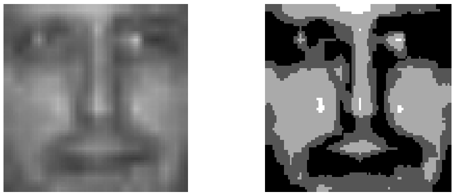

Figure 5-5: An original image of a face on the left. The same image after histogram equalization on the right.

of the derivative) in a volume V around the boundary:

$$I^r = \int_V dP(\mathbf{x}) \left| \frac{df}{dx^r} \right| .$$

where P is the (unknown) probability distribution of the input data. In practice we cannot compute this quantity because we do not know the probability distribution $P(\mathbf{x})$. Instead we can approximate I_r with the sum over the support vectors[2]:

$$I^r \approx \sum_{i=1}^{N_{sv}} \left| \frac{df}{dx_i^r} \right| = \sum_{i=1}^{N_{sv}} \left| \sum_{j=1}^{N_{sv}} c_j K^r(\mathbf{x}_j, \mathbf{x}_i) \right| . \tag{5.2}$$

where N_{sv} is the number of support vectors and $K^r(\mathbf{x}_j, \mathbf{x}_i)$ is the derivative of the kernel with respect to the r^{th} dimension evaluated at $\mathbf{x}_i$. For example for $K(\mathbf{x}_j, \mathbf{x}_i) = (1+\mathbf{x}_j \cdot \mathbf{x}_i)^2$, this is equal to $K^r(\mathbf{x}_j, \mathbf{x}_i) = (1+\mathbf{x}_j \cdot \mathbf{x}_i)\mathbf{x}_i^r$ where $\mathbf{x}_i^r$ is the r^{th} component of vector $\mathbf{x}_i$. Notice that this is only an approximation to the actual derivative: changing the value of a feature may also lead to different solution of the SVM, namely different c_i's in (5.1). We assume that this change is small and we neglect it.

To summarize, the feature selection method consists of computing the quantity in (5.2) for all the input features and selecting the ones with the largest values I^r.

5.4.2 Experiments

For people detection, using the proposed heuristic, 29 of the initial set of 1,326 wavelet coefficients have been selected. An SVM using only the 29 selected features was trained, and the performance of the machine was compared with that of an SVM trained on 29 coefficients selected using a manual method as described in [Papageorgiou *et al.*, 1998b]. The results are shown in Figure 5-6 (bottom plot).

The same heuristic was also tested for face detection. A total of30 of the initial 1,740 wavelet coefficients have been selected, and the performance of an SVM trained using only these 30 features was compared with the performance of a SVM that uses 30 randomly selected features out of the initial 1,740. The performance of the system when 500 of the wavelets were chosen is also shown. Notice that using the proposed method we can select about a third (500) of the original input dimensions without

[2] For separable data these are also the points nearest to the separating surface. For non-separable data we can take the sum over only the support vectors near the boundary.

significantly decreasing the performance of the system. The result is also shown in Figure 5-6 (top plot). Finally for the eigenvector representation, the system was also tested using few principal components. The results are also shown in Figure 5-6 (middle plot).

From all the experiments shown in Figure 5-6 we observe that SVMs are not sensitive to large numbers of input dimensions. In fact in all cases, when using all input dimensions (all wavelets or all eigenvectors) the system performed better (or about the same) than when using few of the input features. This experimental finding gives a first answer, although not general and theoretical, to the questions asked at the beginning of this section: it confirms the difficulty of the feature selection problem, and indicates that SVMs work well even when the input data are high-dimensional, by automatically dealing with irrelevant features.

5.5 Features from probabilistic models

In this section we take a different approach to the problem of finding image representations. Consider a specific class of images (i.e. faces, people, cars) and assume that they are sampled according to a generative probabilistic model $P(\mathbf{x}|\beta)$, where β indicates a set of parameters. As an example consider a Gaussian distribution:

$$P(\mathbf{x}|\beta) = \frac{1}{2\pi^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^\top \Sigma^{-1}(\mathbf{x}-\mathbf{x}_0)\right\} \tag{5.3}$$

where the parameters β are the average image $\mathbf{x}_0$ and the covariance Σ.

Recent work [Papageorgiou *et al.*, 1998a] shows how the assumption of a specific probabilistic model of the form (5.3) suggests a choice of the kernel – and therefore of the "features" – to be used in order to reconstruct, through the SVM regression algorithm, images of faces and people. The relevant features are the principal components $\mathbf{u}_n$ of the set of examples (i.e. faces or people) scaled by the corresponding eigenvalues λ_n. However, [Papageorgiou *et al.*, 1998a] left open the question of what features to choose in order to do classification as opposed to regression, that to discriminate faces (or people) from non-faces (non-people), once the probabilistic model is decided.

Very recently a general approach to the problem of constructing features for classification starting from probabilistic models describing the training examples has been suggested [Jaakkola and Haussler, 1998b]. The choice of the features was made implicitly through the choice of the kernel to be used for a kernel classifier. In [Jaakkola and Haussler, 1998b] a probabilistic model for both the classes to be discriminated was assumed, and the results were also used when a model of only one class was available - which is the case we have.

Let us denote with $L(\mathbf{x}|\beta)$ the log of the probability function and define the Fisher information matrix

$$I = \int d\mathbf{x} P(\mathbf{x}|\beta)\partial_i L(\mathbf{x}|\beta)\partial_j L(\mathbf{x}|\beta),$$

where ∂_i indicates the derivative with respect to the parameter β_i. A natural set of features, ϕ_i, is found by taking the gradient of L with respect to the set of parameters,

$$\phi_i(\mathbf{x}) = I^{-\frac{1}{2}} \frac{\partial L(\mathbf{x}|\beta))}{\partial \beta}. \tag{5.4}$$

These features were theoretical motivated in [Jaakkola and Haussler, 1998b] and shown to lead to kernel classifiers which are at least as discriminative as the Bayes classifier based on the given generative model. We have assumed the generative model (5.3) and rewritten it with respect to the average image $\mathbf{x}_0$ and the eigenvalues λ_n and obtain the set of features according to equation (5.4). For simplicity the principal components were kept fixed in the model. The features obtained in this way were then used as a new input representation in the learning system. The resulting linear kernel obtained by taking the dot product between the features (dot product for the implicitly chosen representation) of a pair of images $\mathbf{x}_i$ and $\mathbf{x}_j$ is:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{n=1}^{N} [-\lambda_n^{-1}(c_n(\mathbf{x}_i) - c_n(\mathbf{x}_j))^2 + \lambda_n^2 c_n(\mathbf{x}_i) c_n(\mathbf{x}_j)], \tag{5.5}$$

where $c_n(x) = (\mathbf{x} - \mathbf{x}_0)^\top \mathbf{u}_n$ and N is the total number of eigenvectors (principal components) used. The parameters $\mathbf{x}_0, \mathbf{u}_n, \lambda_n$ were estimated using the training examples of faces (or people). Note that the training data were used multiple times: once for estimating the parameters of the probabilistic model (5.3), and once to train an SVM classifier.

Notice also that the new features are a non-linear transformation of the pixel representation and the eigenvalues appear in the denominator of each term in the kernel. This is not surprising as the smaller principal components may be important for discrimination: for example, in the problem of face recognition we expect that to discriminate between two faces, we can get more benefit by looking at small details in the image which may not be captured by the larger principal components. Similarly, in the problem of face detection, the non-image class is expected to have the same energy on each principal component, so the small principal components may still be useful for classification. On the other hand, when the goal is to reconstruct or denoise the *in-class images*, we deal with a regression-type problem; in such a case the top principal components capture the most important coefficients for reconstructing the images and only few of them need to be consider.

Equation (5.5) indicates that only a limited number of principal components can be used in practice because small λ_n create numerical instabilities in the learning algorithm. Several experiments were performed by changing the number of principal components used in the model (see Figure 5-7). The results were compared with the image representations discussed above. Notice that the proposed representation performs slightly better than the other ones (when 100 principal components were used), but not significantly better. It may be the case that features from other (more realistic) probabilistic models lead to better systems.

5.6 Summary

This chapter presented experiments for face and people detection with different image representations and kernels using SVM. The main points can be summarized as follows:

- For face detection, pixels, principal components, and Haar wavelets perform almost equally well.

- For people detection, the Haar wavelet representation significantly outperforms the other two.

- We can capture all linear transformation of the original images through the kernel of the SVM.

- For both faces and for people, histogram equalization increases performance dramatically for all the representations tested. Explanations for this result were suggested.

- A feature selection method was proposed and tested.

- New image representations are derived from generative models. In particular, starting from a Gaussian model for the images (i.e. for faces) suggested by the regularization model for regression, new features, that are different from eigenfaces, are derived. These features may have a slight advantage compared to the other three representations tested.

A number of questions and future research directions are still open. What nonlinear transformations of the images (other than histogram equalization) can improve the performance of the system? How can we include prior knowledge through such transformations? It may be possible to design kernels that incorporate such transformations/prior knowledge. Regarding the probabilistic features, it may be interesting to derive such features from other probabilistic models. There is no reason to believe that one Gaussian is enough to model the space of faces images. For example in [Sung and Poggio, 1994] a mixture of six Gaussians was used and shown to be satisfactory.

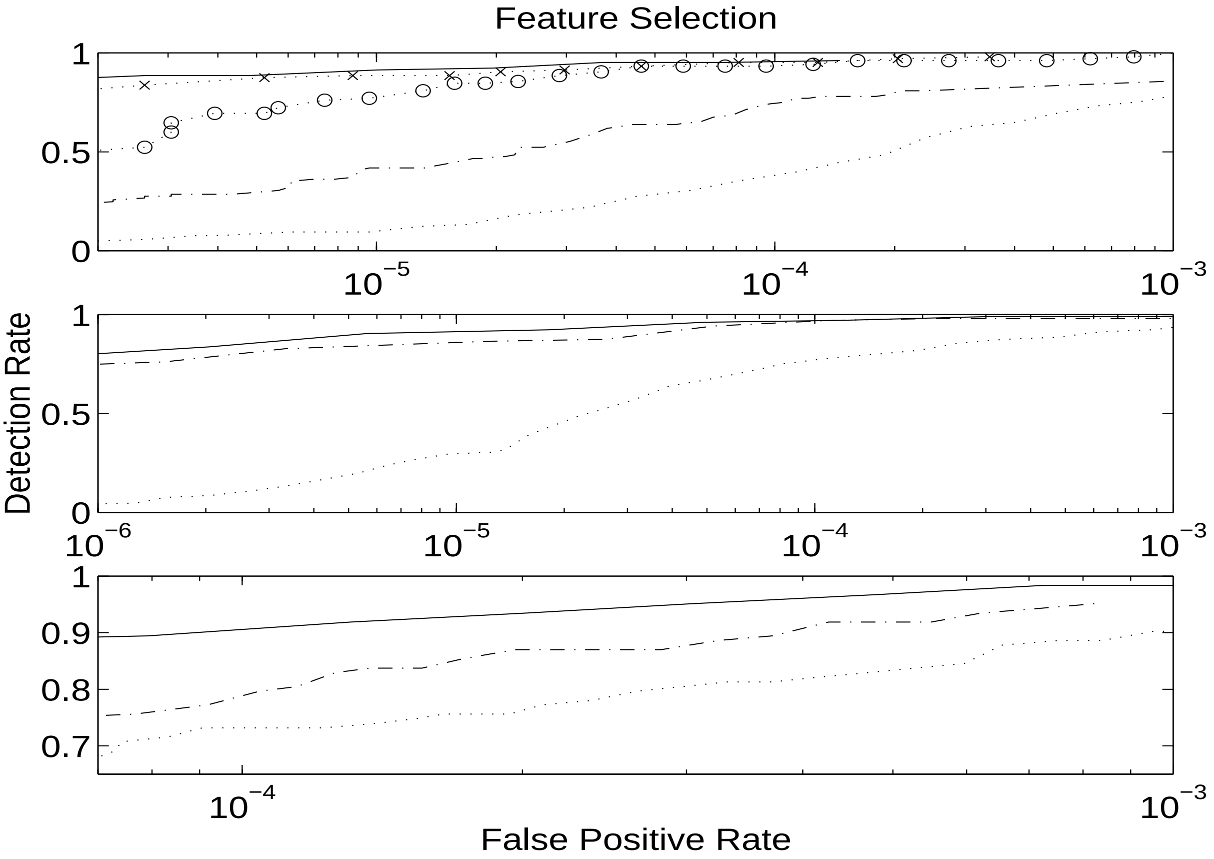

Figure 5-6: *Top figure:* solid line is face detection with all 1,740 wavelets, dashed line is with 30 wavelets chosen using the proposed method, and dotted line is with 30 randomly chosen wavelets. The line with ×'s is with 500 wavelets, and the line with ∘'s is with 120 wavelets, both chosen using the method based on equation (5.2). *Middle figure:* solid line is face detection with all eigenvectors, dashed line is with the 40 principal components, and dotted line is with the 15 principal components. *Bottom figure:* solid line is people detection using all 1,326 wavelets, dashed line is with the 29 wavelets chosen by the method based on equation 5.2 , and dotted line is with the 29 wavelets chosen in ..

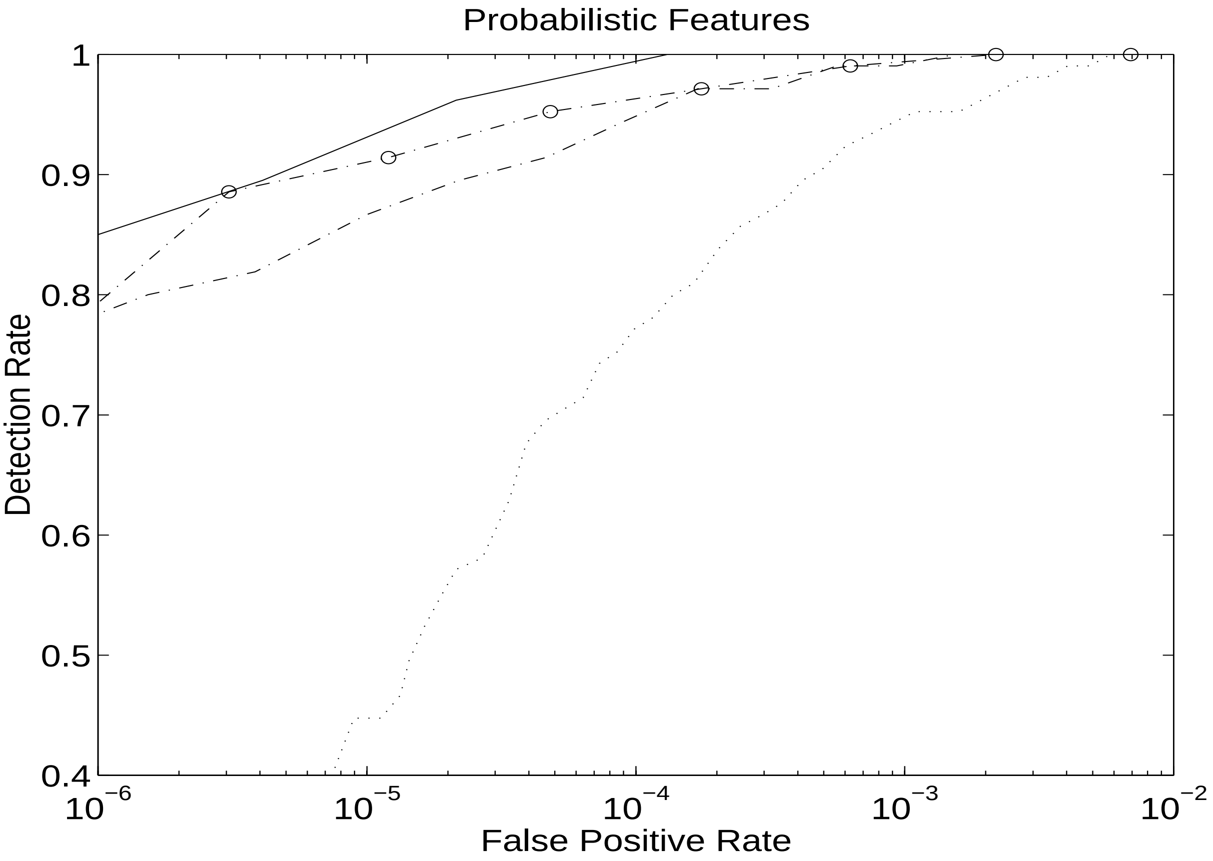

Figure 5-7: Face experiments: Solid line indicates the probabilistic features using 100 principal components, dashed line is for 30 principal components, and dotted for 15. The ROC curves with all wavelets (line with circles) is also shown for comparison. Histogram equalization was performed on the images.

Chapter 6

Further Remarks

The first part of the thesis presented a framework, based on the Statistical Learning Theory (SLT) of Vapnik, within which the problem of learning can be studied and a large family of learning methods can be analyzed. However, SLT, although it provides powerful ideas and formal methods for analyzing learning methods, is not the only approach to learning, as the second part of the thesis also suggests. In this second part, learning was approached through the study of an important quantity describing learning methods, namely the *leave-one-out* error. This approach is independent of the SLT one. It is interesting to study the relations between the two: for example, how can quantities, such as VC-dimension, suggested by SLT can be used to better understand the behavior of the leave-one-out error of a learning machine? Preliminary studies on this issue have been already done [Kearns and Ron, 1999], but more work is needed. Generally, it is important to study the relations between SLT and other approaches to learning (i.e. compression-based approaches [Floyd and Warmuth, 1995], the luckiness function approach [Shawe-Taylor *et al.*, 1998], or the maximum entropy approach [Jaakkola *et al.*, 2000]) as well as to develop new ones.

6.1 Summary and Contributions

The thesis consisted of three main parts. First (chapter 2) some basic theoretical tools were reviewed. In particular, standard Statistical Learning Theory (SLT) and a technical extension of it were presented in chapter 2. Within the extended SLT a theoretical justification and statistical analysis of a large family of learning machines, namely kernel learning machines, was provided. Within this family of machines, two important types were analyzed: Support Vector Machines (SVM) and Regularization Networks (RN) (chapter 3). In the second part of the thesis (chapter 4), the problem of learning was studied not using the tools of SLT but instead using the leave-one-out error characterization of learning machines. Using this quantity other learning architectures, namely ensembles of learning machines, were investigated. Finally, the last part of the thesis discussed an application of learning to object detection. This provided a testbed to discuss important practical issues involved in using learning machines, in particular the problem of finding appropriate data representations (chapter 5).

The contributions of the thesis, as outlined in the introduction, can be summarized as follows:

1. The thesis reviewed standard Statistical Learning Theory and developed an extension within which a *new* (unified) theoretical justification of a number of kernel machines, including RN and SVM, was provided.

2. Within the extended SLT framework, *new* bounds on the expected error (performance) of a large class of kernel machines and particularly SVM, the main learning machines considered in the thesis, were proven.

3. In the second part ensembles of machines were studied. Two types of ensembles were defined: voting combinations, and adaptive combinations. *New* theoretical results on the statistical properties of voting ensembles of kernel machines for classification were shown.

4. The new theoretical findings on voting ensembles of machines were experimentally validated. Both voting and adaptive combinations of machines were further characterized experimentally.

5. The third part discussed an important practical issue, namely the problem of finding appropriate data representations for learning. A trainable system for object detection in images provided the main experimental setup where ideas were tested and discussed.

6.2 Extensions and conjectures

A number of conjectures have been suggested throughout the thesis. These, as well as suggestions for future research, are listed below:

- A theoretical question is related to Kolmogorov's lemmas and theorems (theorems 2.1.1 and 2.1.2, and lemma 2.1.1). As discussed in chapter 2, these theorems and lemmas are about hypothesis spaces that consist of orthants. A first question is whether lemma 2.1.1 can be extended to hypothesis spaces consisting of other types of functions. If that is the case, then one would be able to prove distribution independent bounds on the distance between empirical and expected error (in the spirit of theorem 2.1.2 and of SLT) for the class of hypothesis spaces for which lemma 2.1.1 still holds. That approach could lead to a new analysis of a class of learning methods.

- Conjectures about the "extended" SRM: chapters 2 and 3 discussed a number of conjectures on this framework that are listed again below:

  - From section 2.2.1: we *conjecture* that as (the number of training data) $\ell \to \infty$, for appropriate choice of (the hypothesis space H_n in the defined structure) $n(\ell, \epsilon)$ with $n(\ell, \epsilon) \to \infty$ as $\ell \to \infty$, the expected risk of the solution of the "extended" SRM method converges in probability to a value less than 2ϵ away from the minimum expected risk in $\mathcal{H} = \bigcup_{i=1}^{\infty} H_i$.

  - From section 2.2.1: we *conjecture* that if the target function f_0 belongs to the closure of $\mathcal{H}$, then as $\ell \to \infty$, with appropriate choices of ϵ, $n(\ell, \epsilon)$ and $n^*(\ell, \epsilon)$ the solution of the "extended" SRM method can be proven to satisfy eq. (2.4) in probability. Finding appropriate forms of ϵ, $n(\ell, \epsilon)$ and $n^*(\ell, \epsilon)$ is an open theoretical problem (which is mostly a technical matter)
  - From section 3.4.4: it is a (simple) conjecture that a straightforward applications of the methods used to extend theorem 3.4.5 to 3.4.6 can also be used to extend the bounds of chapter 2 to the case where A is not fixed (and therefore hold for all f with $\|f\|_K^2 < \infty$).

- Quantities characterizing kernel machines: a standard quantity used to study kernel machines is that of the margin. The margin has been also studied in the framework of "luckiness functions" [Shawe-Taylor *et al.*, 1998] and boosting [Shapire *et al.*, 1998]. A possible direction of research is towards finding other (geometric) quantities (or luckiness functions in the framework of [Shawe-Taylor *et al.*, 1998]) that describe learning methods. The radius of the support vectors, as discussed in chapter 4, is such a quantity, but others may be more appropriate.

- Compression and learning: as mentioned in chapter 4, an estimate of the expected error of an SVM is given in terms of the number of support vectors [Vapnik, 1998]. The support vectors can be seen as a *compressed representation of the training data*. In fact, if one deletes the non-support vectors and trains an SVM using only the support vectors, one gets the exact same solution (same classifier) [Vapnik, 1998]. Therefore, the generalization error of an SVM is closely related to the *compression* achieved measured as the percentage of training data that are support vectors (in fact it is only the essential support vectors that are important [Vapnik, 1998]). This is a particular case where the performance of a classifier is related to a form of compression, the compression of the training data, called *sample compression*. This statement can be rephrased in terms of bits used to describe the data: for ℓ training data one needs ℓ bits. If n is the sample compression of a learning machine, then one needs only n bits to describe the training data without influencing the classifier. It has been generally shown that for any classifier, if one can find n out of the ℓ training data such that retraining the classifier using only those n points leads to the exact same solution with the one found using all ℓ training points, then the generalization performance of the learning machine is related to the *sample compression rate* $\frac{n}{\ell}$ [Floyd and Warmuth, 1995]. From a different point of view, the complexity of a hypothesis space can also be seen in the framework of compression: the growth function (VC-dimension) of a hypothesis space effectively measures how many functions there are in the hypothesis space, therefore how many bits one needs to enumerate these functions. Again, the performance of a learning machine depends on the complexity of the machine, which implies that it depends (as in the case of sample compression) on the number of bits required to describe the hypothesis space of the machine. Instead of using ℓ bits to describe the data, we only need h bits to describe the function corresponding to the data, where h is the number of bits required to describe the hypothesis space (the complexity of the space). According to SLT, the generalization performance of a learning machine depends on the ratio of the complexity of the hypothesis space over the number ℓ of training data, which is similar to the aforementioned sample compression rate $\frac{n}{\ell}$: in both cases the performance of the learning machine depends on the compression achieved (instead of using ℓ bits to describe the data we use n in the first case or h in the second). An interesting question is to find the relation between the sample compression rate of a classifier and the complexity (i.e. VC-dimension) of the hypothesis space $\mathcal{H}$ that the classifier uses: one would expect that the sample complexity of a learning machine is related to the complexity (i.e. VC dimension) of the machine (this is shown to be the case in [Floyd and Warmuth, 1995] for a particular type of hypothesis spaces). The direction of relating notions of compression, be it sample compression or compressed representations of hypothesis spaces, with the performance of a learning machine is an open one that may lead to new ideas and views on learning.

- Learning with ensembles versus single machines: a question left open in chapter 4 is whether the bounds on the expected error of ensemble machines derived can be used to describe under which conditions ensembles of machines is expected to work better than single machines. It may be possible to show that the "average geometry" (average D over ρ) is better than the geometry (D over ρ) of a single machine - for example in the case that the training data contain a lot of outliers.

- In chapter 5, a conjecture about the influence of data transformations on the performance of a learning method has been made. In particular, given data $D_\ell \equiv \{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^{\ell}$ and a transformation $g(\mathbf{x})$ (i.e. histogram equalization of images, as discussed in chapter 5), we conjecture the following:

  Suppose that a transformation g satisfies the following two conditions:

  - it is a legal transformation of the input vector, that is it preserves the class label: if $\mathbf{x}$ has label y, then $g(\mathbf{x})$ has the same label y;
  - it increases the entropy of the input set, leading to a more compressed representation.

  It is a *conjecture* that such a transformation will improve the performance of a learning method (classifier). It may be possible to approach this conjecture within the recently developed Maximum Entropy Discrimination (MED) framework for learning [Jaakkola *et al.*, 2000].

- Finally, starting from the conjecture above, and possibly within the MED framework, it is important to further study the twin problems of data representation and learning in a more principled way. It is common experience that the choice of a "good" data representation is as important as the (closely related) choice of the learning method used (as chapter 5 also discussed).

Bibliography

[Allen, 1974] D. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16:125–127, 1974.

[Alon *et al.*, 1993] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Symposium on Foundations of Computer Science*, 1993.

[Aronszajn, 1950] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 686:337–404, 1950.

[Bartlett and Shawe-Taylor, 1998a] P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machine and other patern classifiers. In C. Burges B. Scholkopf, editor, *Advances in Kernel Methods–Support Vector Learning*. MIT press, 1998.

[Bartlett and Shawe-Taylor, 1998b] P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machine and other patern classifiers. In ation performance of support vector machine B. Scholkopf, C. Burges and other patern classifiers, editors, *Advances in Kernel Methods–Support Vector Learning*. MIT press, 1998.

[Bartlett *et al.*, 1996] P. Bartlett, P.M. Long, and R.C. Williamson. Fat-shattering and the learnability of real-valued functions. *Journal of Computer and Systems Sciences*, 52(3):434–452, 1996.

[Bartlett, 1998] P. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important that the size of the network. *IEEE Transactions on Information Theory*, 1998.

[Betke and Makris, 1995] M. Betke and N. Makris. Fast object recognition in noisy images using simulated annealing. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 523–20, 1995.

[Bottou and Vapnik, 1992] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, November 1992.

[Breiman, 1993] L. Breiman. Stacked regression. Technical report, University of California, Berkeley, 1993.

[Breiman, 1996] L. Breiman. Baggind predictors. *Machine Learning*, 26(2):123–140, 1996.

[Buhmann, 1990] M.D. Buhmann. Multivariate cardinal interpolation with radial basis functions. *Constructive Approximation*, 6:225–255, 1990.

[Buhmann, 1991] M.D. Buhmann. On quasi-interpolation with Radial Basis Functions. Numerical Analysis Reports DAMPT 1991/NA3, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, March 1991.

[Burges, 1998] C. Burges. A tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery.* Kluwer Academic Publishers, Boston, 1998. (Volume 2).

[Cantelli, 1933] F. P. Cantelli. Sulla determinazione empirica della leggi di probabilita. *G. Inst. Ital. Attuari*, 4, 1933.

[Chapelle and Vapnik, 1999] O. Chapelle and V. Vapnik. Model selection for support vector machines. In *Advances in Neural Information Processing Systems*, 1999.

[Cochran, 1972] J.A. Cochran. *The analysis of linear integral equations.* McGraw-Hill, New York, 1972.

[Cortes and Vapnik, 1995] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

[Courant and Hilbert, 1962] R. Courant and D. Hilbert. *Methods of mathematical physics. Vol. 2.* Interscience, London, England, 1962.

[de Boor, 1990] C. de Boor. Quasi-interpolants and approximation power of multivariate splines. In M. Gasca and C.A. Micchelli, editors, *Computation of Curves and Surfaces*, pages 313–345. Kluwer Academic Publishers, Dordrecht, Netherlands, 1990.

[DeVore, 1998] R.A. DeVore. Nonlinear approximation. *Acta Numerica*, pages 51–150, 1998.

[Devroye *et al.*, 1996] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Number 31 in Applications of mathematics. Springer, New York, 1996.

[Duda and Hart, 1973] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[Dudley *et al.*, 1991] R.M. Dudley, E. Gine, and J. Zinn. Uniform and universal glivenko-cantelli classes. *Journal of Theoretical Probability*, 4:485–510, 1991.

[Dudley, 1984] R.M. Dudley. A course on empirical processes. *Lecture Notes in Mathematics*, 1097:2–142, 1984.

[Dyn *et al.*, 1986] N. Dyn, D. Levin, and S. Rippa. Numerical procedures for surface fitting of scattered data by radial functions. *SIAM J. Sci. Stat. Comput.*, 7(2):639–659, April 1986.

[Dyn *et al.*, 1989] N. Dyn, I.R.H. Jackson, D. Levin, and A. Ron. On multivariate approximation by integer translates of a basis function. Computer Sciences Technical Report 886, University of Wisconsin–Madison, November 1989.

[Dyn, 1991] N. Dyn. Interpolation and approximation by radial and related functions. In C.K. Chui, L.L. Schumaker, and D.J. Ward, editors, *Approximation Theory, VI*, pages 211–234. Academic Press, New York, 1991.

[Evgeniou *et al.*, 1999] T. Evgeniou, M. Pontil, and T. Poggio. A unified framework for regularization networks and support vector machines. A.I. Memo No. 1654, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1999.

[Floyd and Warmuth, 1995] S. Floyd and M. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine Learning*, 21:269–304, 1995.

[Friedman *et al.*, 1998] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Stanford University, Dept. of Statistics, 1998.

[Gersho and Gray, 1991] A. Gersho and R.M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, Boston, 1991.

[Girosi and Chan, 1995] F. Girosi and N. Chan. Prior knowledge and the creation of "virtual" examples for RBF networks. In *Neural networks for signal processing, Proceedings of the 1995 IEEE-SP Workshop*, pages 201–210, New York, 1995. IEEE Signal Processing Society.

[Girosi *et al.*, 1991] F. Girosi, T. Poggio, and B. Caprile. Extensions of a theory of networks for approximation and learning: outliers and negative examples. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural information processings systems 3*, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[Girosi *et al.*, 1995] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

[Girosi, 1991] F. Girosi. Models of noise and robust estimates. A.I. Memo 1287, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1991.

[Girosi, 1997] F. Girosi. An equivalence between sparse approximation and Support Vector Machines. A.I. Memo 1606, MIT Artificial Intelligence Laboratory, 1997. (available at the URL: http://www.ai.mit.edu/people/girosi/svm.html).

[Girosi, 1998] F. Girosi. An equivalence between sparse approximation and Support Vector Machines. *Neural Computation*, 10(6):1455–1480, 1998.

[Glivenko, 1933] V. I. Glivenko. Sulla determinazione empirica di probabilita. *G. Inst. Ital. Attuari*, 4, 1933.

[Gurvits, 1997] L. Gurvits. A note on scale-sensitive dimension of linear bounded functionals in banach spaces. In *Proceedings of Algorithm Learning Theory*, 1997.

[Härdle, 1990] W. Härdle. *Applied nonparametric regression*, volume 19 of *Econometric Society Monographs.* Cambridge University Press, 1990.

[Hastie and Tibshirani, 1990] T. Hastie and R. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability.* Chapman and Hall, London, 1990.

[Hochstadt, 1973] H. Hochstadt. *Integral Equations.* Wiley Classics Library. John Wiley & Sons, 1973.

[Ivanov, 1976] V.V. Ivanov. *The Theory of Approximate Methods and Their Application to the Numerical Solution of Singular Integral Equations.* Nordhoff International, Leyden, 1976.

[Jaakkola and Haussler, 1998a] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proc. of Neural Information Processing Conference*, 1998.

[Jaakkola and Haussler, 1998b] T. Jaakkola and D. Haussler. Probabilistic kernel regression models. In *Proc. of Neural Information Processing Conference*, 1998.

[Jaakkola *et al.*, 2000] T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *Advances in Neural Information Processing Systems 12*. 2000. to appear.

[Jackson, 1988] I.R.H. Jackson. *Radial Basis Functions methods for multivariate approximation.* Ph.d. thesis, University of Cambridge, U.K., 1988.

[Jain, 1989] Anil K. Jain. *Fundamentals of digital image processing.* Prentice-Hall Information and System Sciences Series, New Jersey, 1989.

[Kearns and Ron, 1999] M. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for the leave-one-out cross validation. *Submitted*, 1999.

[Kearns and Shapire, 1994] M. Kearns and R.E. Shapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and Systems Sciences*, 48(3):464–497, 1994.

[Kearns *et al.*, 1995] M. Kearns, Y. Mansour, A. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *In Proceedings of the Eighth Annual ACM Conference on Computational Learning Theory*, 1995.

[Kolmogorov, 1933] A. N. Kolmogorov. Sulla determinazione empirica di una leggi di probabilita. *G. Inst. Ital. Attuari*, 4, 1933.

[Kolmogorov, 1992] A. N. Kolmogorov. On the empirical determination of a distribution. In S. Kotz and N. L. Johnson, editors, *Breakthroughs in statistics*. Springer-Verlag, 1992.

[LeCun *et al.*, 1989] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L.J. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[Lorentz, 1986] G. G. Lorentz. *Approximation of Functions*. Chelsea Publishing Co., New York, 1986.

[Madych and Nelson, 1990a] W.R. Madych and S.A. Nelson. Polyharmonic cardinal splines: a minimization property. *Journal of Approximation Theory*, 63:303–320, 1990a.

[Marroquin *et al.*, 1987] J. L. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *J. Amer. Stat. Assoc.*, 82:76–89, 1987.

[Mhaskar, 1993a] H.N. Mhaskar. Neural networks for localized approximation of real functions. In C.A. Kamm et al., editor, *Neural networks for signal processing III, Proceedings of the 1993 IEEE-SP Workshop*, pages 190–196, New York, 1993a. IEEE Signal Processing Society.

[Moghaddam and Pentland, 1995] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proceedings of 6th International Conference on Computer Vision*, 1995.

[Niyogi and Girosi, 1996] P. Niyogi and F. Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8:819–842, 1996.

[Oren *et al.*, 1997] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. Computer Vision and Pattern Recognition*, pages 193–199, Puerto Rico, June 16–20 1997.

[Osuna *et al.*, 1997a] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *IEEE Workshop on Neural Networks and Signal Processing*, Amelia Island, FL, September 1997.

[Osuna *et al.*, 1997b] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. A.I. Memo 1602, MIT A. I. Lab., 1997.

[Papageorgiou *et al.*, 1998a] C. Papageorgiou, F. Girosi, and T.Poggio. Sparse correlation kernel based signal reconstruction. Technical Report 1635, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1998. (CBCL Memo 162).

[Papageorgiou *et al.*, 1998b] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision*, Bombay, India, January 1998.

[Platt, 1998] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In C. Burges B. Scholkopf, editor, *Advances in Kernel Methods–Support Vector Learning.* MIT press, 1998.

[Poggio and Girosi, 1989] T. Poggio and F. Girosi. A theory of networks for approximation and learning. A.I. Memo No. 1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.

[Poggio and Girosi, 1990] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), September 1990.

[Poggio and Girosi, 1992] T. Poggio and F. Girosi. Networks for Approximation and Learning. In C. Lau, editor, *Foundations of Neural Networks*, pages 91–106. IEEE Press, Piscataway, NJ, 1992.

[Poggio and Girosi, 1998] T. Poggio and F. Girosi. A Sparse Representation for Function Approximation. *Neural Computation*, 10(6), 1998.

[Pollard, 1984] D. Pollard. *Convergence of stochastic processes.* Springer-Verlag, Berlin, 1984.

[Pontil *et al.*, 1998] M. Pontil, R. Rifkin, and T. Evgeniou. From regression to classification in support vector machines. A.I. Memo 1649, MIT Artificial Intelligence Lab., 1998.

[Powell, 1992] M.J.D. Powell. The theory of radial basis functions approximation in 1990. In W.A. Light, editor, *Advances in Numerical Analysis Volume II: Wavelets, Subdivision Algorithms and Radial Basis Functions*, pages 105–210. Oxford University Press, 1992.

[Rabut, 1991] C. Rabut. How to build quasi-interpolants. applications to polyharmonic B-splines. In P.-J. Laurent, A. Le Mehautè, and L.L. Schumaker, editors, *Curves and Surfaces*, pages 391–402. Academic Press, New York, 1991.

[Rabut, 1992] C. Rabut. An introduction to Schoenberg's approximation. *Computers Math. Applic.*, 24(12):149–175, 1992.

[Rowley *et al.*, 1995] H. Rowley, S. Baluja, and T. Kanade. Human Face Detection in Visual Scenes. Technical Report 95–158, CMU CS, July 1995. Also in *Advances in Neural Information Processing Systems* (8):875-881.

[Schoenberg, 1946a] I.J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions, part a: On the problem of smoothing of graduation, a first class of analytic approximation formulae. *Quart. Appl. Math.*, 4:45–99, 1946a.

[Schoenberg, 1969] I.J. Schoenberg. Cardinal interpolation and spline functions. *Journal of Approximation theory*, 2:167–206, 1969.

[Schumaker, 1981] L.L. Schumaker. *Spline functions: basic theory.* John Wiley and Sons, New York, 1981.

[Shapire *et al.*, 1998] R. Shapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 1998. to appear.

[Shawe-Taylor and Cristianini, 1998] J. Shawe-Taylor and N. Cristianini. Robust bounds on generalization from the margin distribution. Technical Report NeuroCOLT2 Technical Report NC2-TR-1998-029, NeuroCOLT2, 1998.

[Shawe-Taylor *et al.*, 1998] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 1998. To appear. Also: NeuroCOLT Technical Report NC-TR-96-053, 1996, ftp://ftp.dcs.rhbnc.ac.uk/pub/neurocolt/tech_reports.

[Silverman, 1984] B.W. Silverman. Spline smoothing: the equivalent variable kernel method. *The Annals of Statistics*, 12:898–916, 1984.

[Smola and Schölkopf, 1998] A. Smola and B. Schölkopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22:211 – 231, 1998.

[Stewart, 1976] J. Stewart. Positive definite functions and generalizations, an historical survey. *Rocky Mountain J. Math.*, 6:409–434, 1976.

[Sung and Poggio, 1994] K-K. Sung and T. Poggio. Example-based learning for view-based human face detection. In *Proceedings from Image Understanding Workshop*, Monterey, CA, November 1994.

[Tikhonov and Arsenin, 1977] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems.* W. H. Winston, Washington, D.C., 1977.

[Turk and Pentland, 1991] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings CVPR*, pages 586–591, Hawaii, June 1991.

[Vaillant *et al.*, 1994] R. Vaillant, C. Monrocq, and Y. Le Cun. Original approach for the localisation of objects in images. *IEEE Proc. Vis. Image Signal Process.*, 141(4), August 1994.

[Valiant, 1984] L.G. Valiant. A theory of learnable. *Proc. of the 1984 STOC*, pages 436–445, 1984.

[Vapnik and Chervonenkis, 1971] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequences of events to their probabilities. *Th. Prob. and its Applications*, 17(2):264–280, 1971.

[Vapnik and Chervonenkis, 1981] V.N. Vapnik and A. Ya. Chervonenkis. The necessary and sufficient conditions for the uniform convergence of averages to their expected values. *Teoriya Veroyatnostei i Ee Primeneniya*, 26(3):543–564, 1981.

[Vapnik and Chervonenkis, 1991] V.N. Vapnik and A. Ya. Chervonenkis. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, 1(3):283–305, 1991.

[Vapnik, 1982] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data.* Springer-Verlag, Berlin, 1982.

[Vapnik, 1995] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[Vapnik, 1998] V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

[Vetter *et al.*, 1994] T. Vetter, T. Poggio, and H. Bülthoff. The importance of symmetry and virtual views in three-dimensional object recognition. *Current Biology*, 4(1):18–23, 1994.

[Wahba, 1980] G. Wahba. Spline bases, regularization, and generalized cross-validation for solving approximation problems with large quantities of noisy data. In J. Ward and E. Cheney, editors, *Proceedings of the International Conference on Approximation theory in honour of George Lorenz*, Austin, TX, January 8–10 1980. Academic Press.

[Wahba, 1985] G. Wahba. A comparison of GCV and GML for choosing the smoothing parameter in the generalized splines smoothing problem. *The Annals of Statistics*, 13:1378–1402, 1985.

[Wahba, 1990] G. Wahba. *Splines Models for Observational Data.* Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.

[Williamson *et al.*, 1998] R. Williamson, A. Smola, and B. Scholkopf. Generalization performance of regularization networks and support vector machines via entropy numbers. Technical Report NC-TR-98-019, Royal Holloway College University of London, 1998.

[Yuille *et al.*, 1992] A. Yuille, P. Hallinan, and D. Cohen. Feature Extraction from Faces using Deformable Templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.