Preface

Statistical Learning Theory now plays a more active role: after the general analysis of learning processes, the research in the area of synthesis of optimal algorithms was started. These studies, however, do not belong to history yet. They are a subject of today's research activities.

Vladimir Vapnik (1995)

The Support Vector Machine has recently been introduced as a new technique for solving a variety of learning and function estimation problems. During a workshop at the annual Neural Information Processing Systems (NIPS) conference, held in Breckenridge, Colorado in December 1997, a snapshot of the state of the art in Support Vector learning was recorded. A variety of people helped in this, among them our co-organizer Léon Bottou, the NIPS workshop chairs Steve Nowlan and Rich Zemel, and all the workshop speakers and attendees who contributed to lively discussions. After the workshop, we decided that it would be worthwhile to invest some time to have the snapshot printed.

We invited all the speakers as well as other researchers to submit papers for this collection, and integrated the results into the present book. We believe that it covers the full range of current Support Vector research at an early point in time. This is possible for two reasons. First, the field of SV learning is in its early (and thus exciting) days. Second, this book gathers expertise from all contributers, whom we wholeheartedly thank for all the work they have put into our joint effort. Any single person trying to accomplish this task would most likely have failed: either by writing a book which is less comprehensive, or by taking more time to complete the book.

It is our hope that this outweighs the shortcomings of the book, most notably the fact that a collection of chapters can never be as homogeneous as a book conceived by a single person. We have tried to compensate for this by the selection and refereeing process of the submissions. In addition, we have written an introductory chapter describing the SV algorithm in some detail (chapter 1), and added a roadmap (chapter 2) which describes the actual contributions which are to follow in chapters 3 through 20.

Bernhard Schölkopf, Christopher J.C. Burges, Alexander J. Smola

Berlin, Holmdel, July 1998

1 Introduction to Support Vector Learning

The goal of this chapter, which describes the central ideas of SV learning, is twofold. First, we want to provide an introduction for readers unfamiliar with this field. Second, this introduction serves as a source of the basic equations for the chapters of this book. For more exhaustive treatments, we refer the interested reader to Vapnik (1995); Schölkopf (1997); Burges (1998).

1.1 Learning Pattern Recognition from Examples

Training Data

Let us start with the problem of learning how to recognize patterns. Formally, we want to estimate a function $f : \mathbb{R}^N \to \{\pm 1\}$ using input-output training data

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell) \in \mathbb{R}^N \times \{\pm 1\}, \tag{1.1}$$

such that f will correctly classify unseen examples $(\mathbf{x}, y)$, i.e. $f(\mathbf{x}) = y$ for examples $(\mathbf{x}, y)$ that were generated from the same underlying probability distribution $P(\mathbf{x}, y)$ as the training data. If we put no restriction on the class of functions that we choose our estimate f from, however, even a function which does well on the training data, e.g. by satisfying $f(\mathbf{x}_i) = y_i$ for all $i = 1, \ldots, \ell$, need not generalize well to unseen examples. To see this, note that for each function f and any test set $(\bar{\mathbf{x}}_1, \bar{y}_1), \ldots, (\bar{\mathbf{x}}_{\bar{\ell}}, \bar{y}_{\bar{\ell}}) \in \mathbb{R}^N \times \{\pm 1\}$, satisfying $\{\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_{\bar{\ell}}\} \cap \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\} = \{\}$, there exists another function f^* such that $f^*(\mathbf{x}_i) = f(\mathbf{x}_i)$ for all $i = 1, \ldots, \ell$, yet $f^*(\bar{\mathbf{x}}_i) \neq f(\bar{\mathbf{x}}_i)$ for all $i = 1, \ldots, \bar{\ell}$. As we are only given the training data, we have no means of selecting which of the two functions (and hence which of the completely different sets of test outputs) is preferable. Hence, only minimizing the training error (or *empirical risk*),

Test Data

Empirical Risk

$$R_{emp}[f] = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{1}{2} |f(\mathbf{x}_i) - y_i|, \tag{1.2}$$

Risk

does not imply a small test error (called *risk*), averaged over test examples drawn from the underlying distribution $P(\mathbf{x}, y)$,

$$R[f] = \int \frac{1}{2} |f(\mathbf{x}) - y| \, dP(\mathbf{x}, y). \tag{1.3}$$

Statistical learning theory (Vapnik and Chervonenkis, 1974; Vapnik, 1979), or VC (Vapnik-Chervonenkis) theory, shows that it is imperative to restrict the class of functions that f is chosen from to one which has a *capacity* that is suitable for the amount of available training data. VC theory provides *bounds* on the test error. The minimization of these bounds, which depend on both the empirical risk and the capacity of the function class, leads to the principle of *structural risk minimization* (Vapnik, 1979). The best-known capacity concept of VC theory is the *VC dimension*, defined as the largest number h of points that can be separated in all possible ways using functions of the given class (cf. chapter 4). An example of a VC bound is the following: if $h < \ell$ is the VC dimension of the class of functions that the learning machine can implement, then for all functions of that class, with a probability of at least $1 - \eta$, the bound

VC dimension

$$R(\alpha) \leq R_{emp}(\alpha) + \phi\left(\frac{h}{\ell}, \frac{\log(\eta)}{\ell}\right) \tag{1.4}$$

holds, where the *confidence term* ϕ is defined as

$$\phi\left(\frac{h}{\ell}, \frac{\log(\eta)}{\ell}\right) = \sqrt{\frac{h\left(\log\frac{2\ell}{h} + 1\right) - \log(\eta/4)}{\ell}}. \tag{1.5}$$

Tighter bounds can be formulated in terms of other concepts, such as the *annealed VC entropy* or the *Growth function*. These are usually considered to be harder to evaluate (cf., however, chapter 9), but they play a fundamental role in the conceptual part of VC theory (Vapnik, 1995). Alternative capacity concepts that can be used to formulate bounds include the *fat shattering dimension*, cf. chapter 4.

The bound (1.4) deserves some further explanatory remarks. Suppose we wanted to learn a "dependency" where $P(\mathbf{x}, y) = P(\mathbf{x}) \cdot P(y)$, i.e. where the pattern $\mathbf{x}$ contains no information about the label y, with uniform $P(y)$. Given a training sample of fixed size, we can then surely come up with a learning machine which achieves zero training error (provided we have no examples contradicting each other). However, in order to reproduce the random labellings, this machine will necessarily require a large VC dimension h. Thus, the confidence term (1.5), increasing monotonically with h, will be large, and the bound (1.4) will *not* support possible hopes that due to the small training error, we should expect a small test error. This makes it understandable how (1.4) can hold independent of assumptions about the underlying distribution $P(\mathbf{x}, y)$: it always holds (provided that $h < \ell$), but it does not always make a nontrivial prediction — a bound on an error rate becomes void if it is larger than the maximum error rate. In order to get nontrivial predictions from (1.4), the function space must be restricted such that the capacity (e.g. VC dimension) is small enough (in relation to the available amount of data).

1.2 Hyperplane Classifiers

To design learning algorithms, one thus needs to come up with a class of functions whose capacity can be computed. Vapnik and Lerner (1963) and Vapnik and

Chervonenkis (1964) considered the class of hyperplanes

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \quad \mathbf{w} \in \mathbb{R}^N, b \in R, \tag{1.6}$$

corresponding to decision functions

$$f(\mathbf{x}) = \operatorname{sgn}((\mathbf{w} \cdot \mathbf{x}) + b), \tag{1.7}$$

and proposed a learning algorithm for separable problems, termed the *Generalized Portrait*, for constructing f from empirical data. It is based on two facts. First, among all hyperplanes separating the data, there exists a unique one yielding the maximum margin of separation between the classes,

Optimal Hyperplane

$$\max_{\mathbf{w},b} \min\{\|\mathbf{x} - \mathbf{x}_i\| : \mathbf{x} \in \mathbb{R}^N, (\mathbf{w} \cdot \mathbf{x}) + b = 0, i = 1, \ldots, \ell\}. \tag{1.8}$$

Second, the capacity decreases with increasing margin.

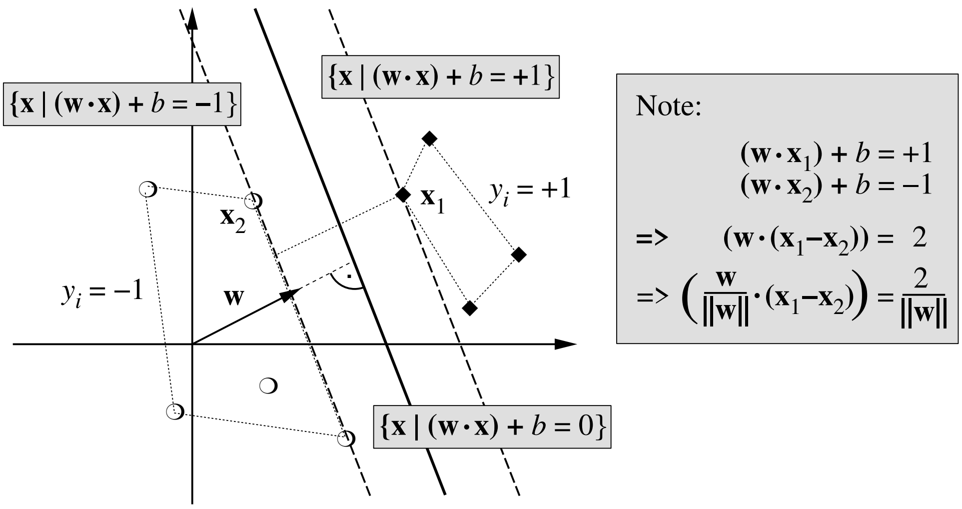

Figure 1.1 A binary classification toy problem: separate balls from diamonds. The *optimal hyperplane* is orthogonal to the shortest line connecting the convex hulls of the two classes (dotted), and intersects it half-way between the two classes. The problem being separable, there exists a weight vector $\mathbf{w}$ and a threshold b such that $y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) > 0$ $(i = 1, \ldots, \ell)$. Rescaling $\mathbf{w}$ and b such that the point(s) closest to the hyperplane satisfy $|(\mathbf{w} \cdot \mathbf{x}_i) + b| = 1$, we obtain a *canonical* form $(\mathbf{w}, b)$ of the hyperplane, satisfying $y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1$. Note that in this case, the *margin*, measured perpendicularly to the hyperplane, equals $2/\|\mathbf{w}\|$. This can be seen by considering two points $\mathbf{x}_1, \mathbf{x}_2$ on opposite sides of the margin, i.e. $(\mathbf{w} \cdot \mathbf{x}_1) + b = 1$, $(\mathbf{w} \cdot \mathbf{x}_2) + b = -1$, and projecting them onto the hyperplane normal vector $\mathbf{w}/\|\mathbf{w}\|$.

To construct this *Optimal Hyperplane* (cf. figure 1.1), one solves the following optimization problem:

$$\text{minimize} \quad \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \tag{1.9}$$

$$\text{subject to} \quad y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1, \quad i = 1, \dots, \ell. \tag{1.10}$$

Lagrangian

This constrained optimization problem is dealt with by introducing Lagrange multipliers $\alpha_i \geq 0$ and a Lagrangian

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i \left(y_i \cdot ((\mathbf{x}_i \cdot \mathbf{w}) + b) - 1 \right). \tag{1.11}$$

The Lagrangian L has to be minimized with respect to the *primal variables* $\mathbf{w}$ and b and maximized with respect to the *dual variables* α_i (i.e. a saddle point has to be found). Let us try to get some intuition for this. If a constraint (1.10) is violated, then $y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) - 1 < 0$, in which case L can be increased by increasing the corresponding α_i. At the same time, $\mathbf{w}$ and b will have to change such that L decreases. To prevent $-\alpha_i \left(y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) - 1 \right)$ from becoming arbitrarily large, the change in $\mathbf{w}$ and b will ensure that, provided the problem is separable, the constraint will eventually be satisfied. Similarly, one can understand that for all constraints which are not precisely met as equalities, i.e. for which $y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) - 1 > 0$, the corresponding α_i must be 0: this is the value of α_i that maximizes L. The latter is the statement of the Karush-Kuhn-Tucker complementarity conditions of optimization theory (Karush, 1939; Kuhn and Tucker, 1951; Bertsekas, 1995).

KKT Conditions

The condition that at the saddle point, the derivatives of L with respect to the primal variables must vanish,

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \quad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \tag{1.12}$$

leads to

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0 \tag{1.13}$$

and

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i. \tag{1.14}$$

Support Vector

The solution vector thus has an expansion in terms of a subset of the training patterns, namely those patterns whose α_i is non-zero, called *Support Vectors*. By the Karush-Kuhn-Tucker complementarity conditions

$$\alpha_i \cdot [y_i((\mathbf{x}_i \cdot \mathbf{w}) + b) - 1] = 0, \quad i = 1, \dots, \ell, \tag{1.15}$$

the Support Vectors lie on the margin (cf. figure 1.1). All remaining examples of the training set are irrelevant: their constraint (1.10) does not play a role in the optimization, and they do not appear in the expansion (1.14). This nicely captures our intuition of the problem: as the hyperplane (cf. figure 1.1) is completely determined by the patterns closest to it, the solution should not depend on the other examples.

By substituting (1.13) and (1.14) into L, one eliminates the primal variables and arrives at the Wolfe dual of the optimization problem (e.g. Bertsekas, 1995): find multipliers α_i which

Dual Optimization Problem

$$\text{maximize} \qquad W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{1.16}$$

$$\text{subject to} \qquad \alpha_i \geq 0, \quad i = 1, \ldots, \ell, \text{ and } \sum_{i=1}^{\ell} \alpha_i y_i = 0. \tag{1.17}$$

The hyperplane decision function can thus be written as

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} y_i \alpha_i \cdot (\mathbf{x} \cdot \mathbf{x}_i) + b \right) \tag{1.18}$$

where b is computed using (1.15).

The structure of the optimization problem closely resembles those that typically arise in Lagrange's formulation of mechanics (e.g. Goldstein, 1986). Also there, often only a subset of the constraints become active. For instance, if we keep a ball in a box, then it will typically roll into one of the corners. The constraints corresponding to the walls which are not touched by the ball are irrelevant, the walls could just as well be removed.

Seen in this light, it is not too surprising that it is possible to give a mechanical interpretation of optimal margin hyperplanes (Burges and Schölkopf, 1997): If we assume that each support vector $\mathbf{x}_i$ exerts a perpendicular force of size α_i and sign y_i on a solid plane sheet lying along the hyperplane, then the solution satisfies the requirements of mechanical stability. The constraint (1.13) states that the forces on the sheet sum to zero; and (1.14) implies that the torques also sum to zero, via $\sum_i \mathbf{x}_i \times y_i \alpha_i \cdot \mathbf{w}/\|\mathbf{w}\| = \mathbf{w} \times \mathbf{w}/\|\mathbf{w}\| = 0$.

There are several theoretical arguments supporting the good generalization performance of the optimal hyperplane (Vapnik and Chervonenkis (1974); Vapnik (1979), cf. chapters 3 and 4). In addition, it is computationally attractive, since it can be constructed by solving a quadratic programming problem. But how can this be generalized to the case of decision functions which, unlike (1.7), are nonlinear in the data?

1.3 Feature Spaces and Kernels

To construct SV machines, the optimal hyperplane algorithm had to be augmented by a method for computing dot products in feature spaces nonlinearly related to input space (Aizerman et al., 1964; Boser et al., 1992). The basic idea is to map the data into some other dot product space (called the *feature space*) F via a nonlinear

Feature Space

map

$$\Phi : \mathbb{R}^N \to F, \tag{1.19}$$

and perform the above linear algorithm in F.

For instance, suppose we are given patterns $\mathbf{x} \in \mathbb{R}^N$ where most information is contained in the d-th order products (monomials) of entries x_j of $\mathbf{x}$, i.e. $x_{j_1} \cdot \ldots \cdot x_{j_d}$, where $j_1, \ldots, j_d \in \{1, \ldots, N\}$. In that case, we might prefer to extract these monomial features first, and work in the feature space F of all products of d entries. This approach, however, fails for realistically sized problems: for N-dimensional input patterns, there exist $(N + d - 1)!/(d!(N - 1)!)$ different monomials. Already 16×16 pixel input images (e.g. in character recognition) and a monomial degree $d = 5$ yield a dimensionality of 10^{10}.

This problem can be overcome by noticing that both the construction of the optimal hyperplane in F (cf. (1.16)) and the evaluation of the corresponding decision function (1.18) only require the evaluation of dot products $(\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$, and never the mapped patterns $\Phi(\mathbf{x})$ in explicit form. This is crucial, since in some cases, the dot products can be evaluated by a simple kernel

Mercer Kernel

$$k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})). \tag{1.20}$$

For instance, the polynomial kernel

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d \tag{1.21}$$

can be shown to correspond to a map Φ into the space spanned by all products of exactly d dimensions of $\mathbb{R}^N$ (Poggio (1975); Boser et al. (1992); Burges (1998); for a proof, see chapter 20). For $d = 2$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$, e.g., we have (Vapnik, 1995)

$$(\mathbf{x} \cdot \mathbf{y})^2 = (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2)(y_1^2, y_2^2, \sqrt{2}\, y_1 y_2)^\top = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})), \tag{1.22}$$

defining $\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2)$.

By using $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + c)^d$ with $c > 0$, we can take into account all product of order up to d (i.e. including those of order smaller than d).

More generally, the following theorem of functional analysis shows that kernels k of positive integral operators give rise to maps Φ such that (1.20) holds (Mercer, 1909; Aizerman et al., 1964; Boser et al., 1992):

Theorem 1.1 (Mercer)
If k is a continuous symmetric kernel of a positive integral operator T, i.e.

$$(Tf)(\mathbf{y}) = \int_{\mathcal{C}} k(\mathbf{x}, \mathbf{y}) f(\mathbf{x})\, d\mathbf{x} \tag{1.23}$$

with

$$\int_{\mathcal{C} \times \mathcal{C}} k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y})\, d\mathbf{x}\, d\mathbf{y} \geq 0 \tag{1.24}$$

for all $f \in L_2(\mathcal{C})$ ($\mathcal{C}$ being a compact subset of $\mathbb{R}^N$), it can be expanded in a uniformly convergent series (on $\mathcal{C} \times \mathcal{C}$) in terms of T's eigenfunctions ψ_j and positive eigenvalues λ_j,

$$k(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{N_F} \lambda_j \psi_j(\mathbf{x}) \psi_j(\mathbf{y}), \tag{1.25}$$

where $N_F \leq \infty$ is the number of positive eigenvalues.

Note that originally proven for the case where $\mathcal{C} = [a, b]$ ($a < b \in \mathbb{R}$), this theorem also holds true for general compact spaces (Dunford and Schwartz, 1963).

An equivalent way to characterize Mercer kernels is that they give rise to positive matrices $K_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$ for all $\{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ (Saitoh, 1988). One of the implications that need to be proven to show this equivalence follows from the fact that K_{ij} is a Gram matrix: for $\boldsymbol{\alpha} \in \mathbb{R}^\ell$, we have $(\boldsymbol{\alpha} \cdot K\boldsymbol{\alpha}) = \| \sum_{i=1}^{\ell} \alpha_i \Phi(\mathbf{x}_i) \|^2 \geq 0$.

From (1.25), it is straightforward to construct a map Φ into a potentially infinite-dimensional l_2 space which satisfies (1.20). For instance, we may use

$$\Phi(\mathbf{x}) = (\sqrt{\lambda_1}\psi_1(\mathbf{x}), \sqrt{\lambda_2}\psi_2(\mathbf{x}), \ldots). \tag{1.26}$$

Rather than thinking of the feature space as an l_2 space, we can alternatively represent it as the Hilbert space $\mathcal{H}_k$ containing all linear combinations of the functions $f(.) = k(\mathbf{x}_i, .)$ ($\mathbf{x}_i \in \mathcal{C}$). To ensure that the map $\Phi : \mathcal{C} \to \mathcal{H}_k$, which in this case is defined as

$$\Phi(\mathbf{x}) = k(\mathbf{x}, .), \tag{1.27}$$

satisfies (1.20), we need to endow $\mathcal{H}_k$ with a suitable dot product $\langle ., . \rangle$. In view of the definition of Φ, this dot product needs to satisfy

$$\langle k(\mathbf{x}, .), k(\mathbf{y}, .) \rangle = k(\mathbf{x}, \mathbf{y}), \tag{1.28}$$

Reproducing Kernel

which amounts to saying that k is a *reproducing kernel* for $\mathcal{H}_k$. For a Mercer kernel (1.25), such a dot product does exist. Since k is symmetric, the ψ_i ($i = 1, \ldots, N_F$) can be chosen to be orthogonal with respect to the dot product in $L_2(C)$, i.e. $(\psi_j, \psi_n)_{L_2(C)} = \delta_{jn}$, using the Kronecker δ_{jn}. From this, we can construct $\langle ., . \rangle$ such that

$$\langle \sqrt{\lambda_j}\psi_j, \sqrt{\lambda_n}\psi_n \rangle = \delta_{jn}. \tag{1.29}$$

Substituting (1.25) into (1.28) then proves the desired equality (for further details, see chapter 6 and Aronszajn (1950); Wahba (1973); Girosi (1998); Schölkopf (1997)).

Besides (1.21), SV practicioners use sigmoid kernels

$$k(\mathbf{x}, \mathbf{y}) = \tanh(\kappa(\mathbf{x} \cdot \mathbf{y}) + \Theta) \tag{1.30}$$

for suitable values of gain κ and threshold Θ (cf. chapter 7), and radial basis function kernels, as for instance (Aizerman et al., 1964; Boser et al., 1992; Schölkopf et al., 1997b)

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|^2/(2\,\sigma^2)\right), \tag{1.31}$$

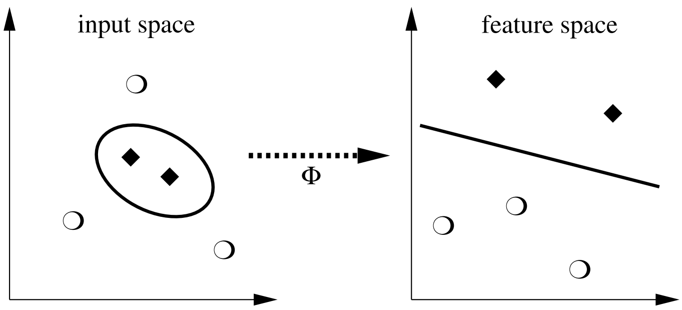

Figure 1.2 The idea of SV machines: map the training data nonlinearly into a higher-dimensional feature space via Φ, and construct a separating hyperplane with maximum margin there. This yields a nonlinear decision boundary in input space. By the use of a kernel function (1.20), it is possible to compute the separating hyperplane without explicitly carrying out the map into the feature space.

with $\sigma > 0$. Note that when using Gaussian kernels, for instance, the feature space $\mathcal{H}_k$ thus contains all superpositions of Gaussians on $\mathcal{C}$ (plus limit points), whereas by definition of Φ (1.27), only single bumps $k(\mathbf{x}, .)$ do have pre-images under Φ.

1.4 Support Vector Machines

To construct SV machines, one computes an optimal hyperplane in feature space. To this end, we substitute $\Phi(\mathbf{x}_i)$ for each training example $\mathbf{x}_i$. The weight vector (cf. (1.14)) then becomes an expansion in feature space, and will thus typically no more correspond to the image of a single vector from input space (cf. Schölkopf et al. (1998c) for a formula how to compute the pre-image if it exists). Since all patterns only occur in dot products, one can substitute Mercer kernels k for the dot products (Boser et al., 1992; Guyon et al., 1993), leading to decision functions of the more general form (cf. (1.18))

Decision Function

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} y_i\alpha_i \cdot (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i)) + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} y_i\alpha_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b\right) \tag{1.32}$$

and the following quadratic program (cf. (1.16)):

$$\text{maximize} \quad W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{1.33}$$

$$\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, \dots, \ell, \text{ and } \sum_{i=1}^{\ell} \alpha_i y_i = 0. \tag{1.34}$$

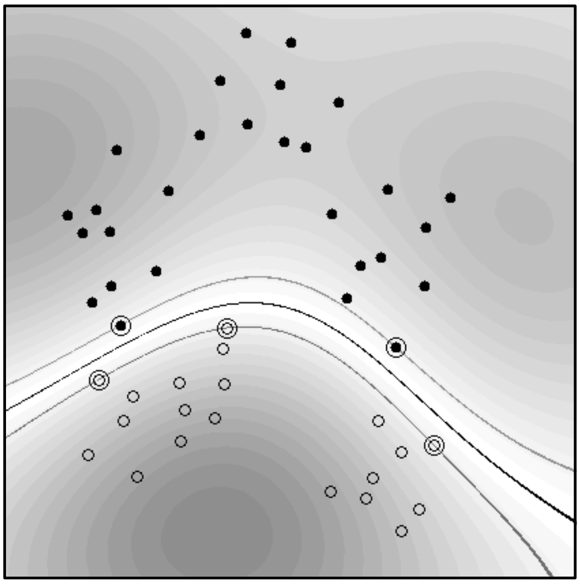

Figure 1.3 Example of a Support Vector classifier found by using a radial basis function kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2)$. Both coordinate axes range from -1 to +1. Circles and disks are two classes of training examples; the middle line is the decision surface; the outer lines precisely meet the constraint (1.10). Note that the Support Vectors found by the algorithm (marked by extra circles) are not centers of clusters, but examples which are critical for the given classification task. Grey values code the modulus of the argument $\sum_{i=1}^{\ell} y_i \alpha_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b$ of the decision function (1.32). (From Schölkopf et al. (1996a), see also Burges (1998).)

Soft Margin Hyperplane

In practice, a separating hyperplane may not exist, e.g. if a high noise level causes a large overlap of the classes. To allow for the possibility of examples violating (1.10), one introduces slack variables (Cortes and Vapnik, 1995; Vapnik, 1995)

$$\xi_i \geq 0, \quad i = 1, \ldots, \ell, \tag{1.35}$$

along with relaxed constraints

$$y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, \ell. \tag{1.36}$$

A classifier which generalizes well is then found by controlling both the classifier capacity (via $\|\mathbf{w}\|$) and the number of training errors, minimizing the objective function

$$\tau(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{\ell} \xi_i \tag{1.37}$$

subject to the constraints (1.35) and (1.36), for some value of the constant $C > 0$ determining the trade-off. Here and below, we use boldface greek letters as a shorthand for corresponding vectors $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_\ell)$. Incorporating kernels, and rewriting it in terms of Lagrange multipliers, this again leads to the problem of maximizing (1.33), subject to the constraints

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, \ell, \text{ and } \sum_{i=1}^{\ell} \alpha_i y_i = 0. \tag{1.38}$$

The only difference from the separable case is the upper bound C on the Lagrange multipliers α_i. This way, the influence of the individual patterns (which could always be outliers) gets limited. As above, the solution takes the form (1.32). The threshold b can be computed by exploiting the fact that for all SVs $\mathbf{x}_i$ with $\alpha_i < C$, the slack variable ξ_i is zero (this again follows from the Karush-Kuhn-Tucker complementarity conditions), and hence

$$\sum_{j=1}^{\ell} y_j \alpha_j \cdot k(\mathbf{x}_i, \mathbf{x}_j) + b = y_i. \tag{1.39}$$

If one uses an optimizer that works with the double dual (e.g. Vanderbei, 1997), one can also recover the value of the primal variable b directly from the corresponding double dual variable.

1.5 Support Vector Regression

The concept of the margin is specific to pattern recognition. To generalize the SV algorithm to regression estimation (Vapnik, 1995), an analogue of the margin is constructed in the space of the target values y (note that in regression, we have $y \in \mathbb{R}$) by using Vapnik's ε-insensitive loss function (figure 1.4)

$$|y - f(\mathbf{x})|_\varepsilon := \max\{0, |y - f(\mathbf{x})| - \varepsilon\}. \tag{1.40}$$

To estimate a linear regression

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b \tag{1.41}$$

with precision ε, one minimizes

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{\ell} |y_i - f(\mathbf{x}_i)|_\varepsilon. \tag{1.42}$$

Written as a constrained optimization problem, this reads (Vapnik, 1995):

$$\begin{aligned}
\text{minimize} \quad & \tau(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) && (1.43)\\
\text{subject to} \quad & ((\mathbf{w} \cdot \mathbf{x}_i) + b) - y_i \leq \varepsilon + \xi_i && (1.44)\\
& y_i - ((\mathbf{w} \cdot \mathbf{x}_i) + b) \leq \varepsilon + \xi_i^* && (1.45)\\
& \xi_i, \xi_i^* \geq 0 && (1.46)
\end{aligned}$$

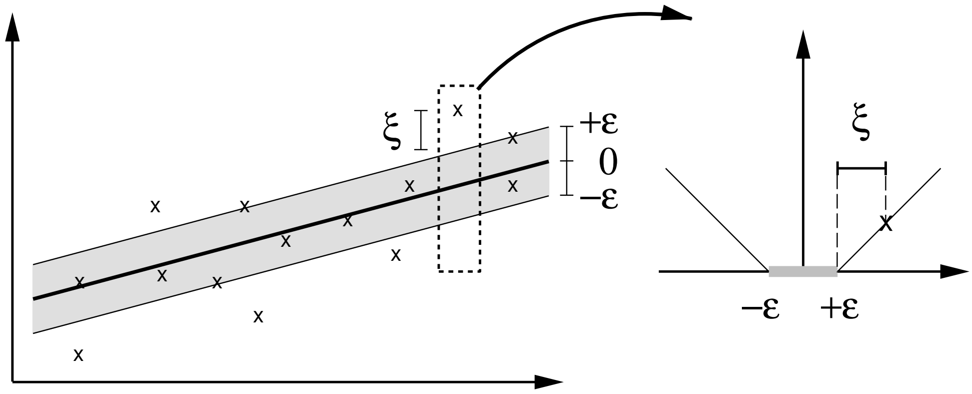

Figure 1.4 In SV regression, a desired accuracy ε is specified a priori. One then attempts to fit a tube with radius ε to the data. The trade-off between model complexity and points lying outside of the tube (with positive slack variables ξ) is determined by minimizing (1.43).

for all $i = 1, \ldots, \ell$. Note that according to (1.44) and (1.45), any error smaller than ε does not require a nonzero ξ_i or ξ_i^*, and hence does not enter the objective function (1.43).

Generalization to nonlinear regression estimation is carried out using kernel functions, in complete analogy to the case of pattern recognition. Introducing Lagrange multipliers, one thus arrives at the following optimization problem: for $C > 0, \varepsilon \geq 0$ chosen a priori,

$$\text{maximize} \quad W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = -\varepsilon \sum_{i=1}^{\ell} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) y_i - \frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(\mathbf{x}_i, \mathbf{x}_j) \tag{1.47}$$

$$\text{subject to} \quad 0 \leq \alpha_i, \alpha_i^* \leq C, \quad i = 1, \ldots, \ell, \text{ and } \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) = 0. \tag{1.48}$$

Regression Function

The regression estimate takes the form

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) k(\mathbf{x}_i, \mathbf{x}) + b, \tag{1.49}$$

where b is computed using the fact that (1.44) becomes an equality with $\xi_i = 0$ if $0 < \alpha_i < C$, and (1.45) becomes an equality with $\xi_i^* = 0$ if $0 < \alpha_i^* < C$.

Several extensions of this algorithm are possible. From an abstract point of view, we just need some target function which depends on the vector $(\mathbf{w}, \boldsymbol{\xi})$ (cf. (1.43)). There are multiple degrees of freedom for constructing it, including some freedom how to penalize, or regularize, different parts of the vector, and some freedom how to use the kernel trick. For instance, more general loss functions can be used for

$\boldsymbol{\xi}$, leading to problems that can still be solved efficiently (Smola and Schölkopf, 1998b; Smola et al., 1998a). Moreover, norms other than the 2-norm $\|.\|$ can be used to regularize the solution (cf. chapters 18 and 19). Yet another example is that polynomial kernels can be incorporated which consist of multiple layers, such that the first layer only computes products within certain specified subsets of the entries of $\mathbf{w}$ (Schölkopf et al., 1998d).

Finally, the algorithm can be modified such that ε need not be specified a priori. Instead, one specifies an upper bound $0 \le \nu \le 1$ on the fraction of points allowed to lie outside the tube (asymptotically, the number of SVs) and the corresponding ε is computed automatically. This is achieved by using as primal objective function

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\left(\nu\ell\varepsilon + \sum_{i=1}^{\ell} |y_i - f(\mathbf{x}_i)|_\varepsilon\right) \tag{1.50}$$

instead of (1.42), and treating $\varepsilon \ge 0$ as a parameter that we minimize over (Schölkopf et al., 1998a).

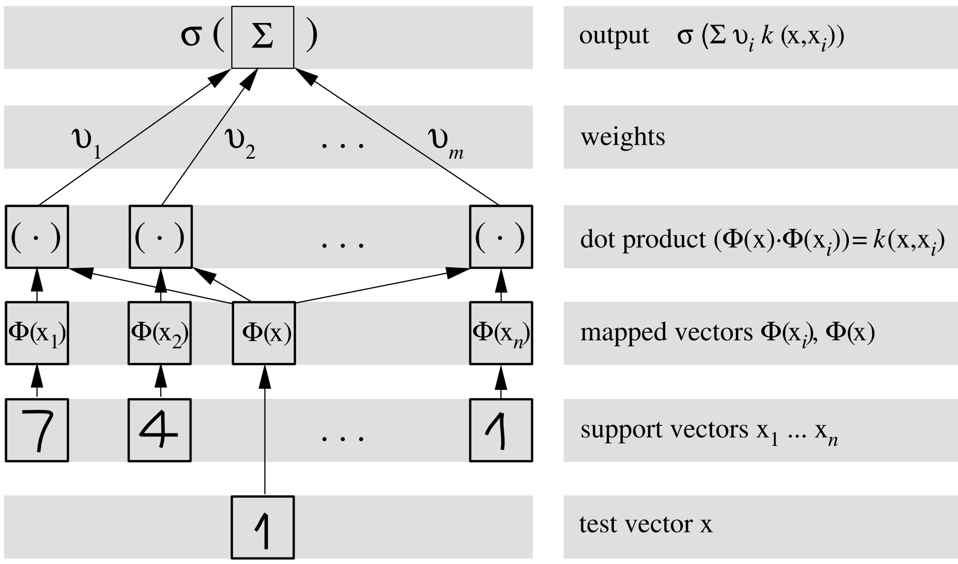

Figure 1.5 Architecture of SV machines. The input $\mathbf{x}$ and the Support Vectors $\mathbf{x}_i$ are nonlinearly mapped (by Φ) into a feature space F, where dot products are computed. By the use of the kernel k, these two layers are in practice computed in one single step. The results are linearly combined by weights υ_i, found by solving a quadratic program (in pattern recognition, $\upsilon_i = y_i\alpha_i$; in regression estimation, $\upsilon_i = \alpha_i^* - \alpha_i$). The linear combination is fed into the function σ (in pattern recognition, $\sigma(x) = \mathrm{sgn}(x + b)$; in regression estimation, $\sigma(x) = x + b$).

1.6 Empirical Results, Implementations, and Further Developments

Having described the basics of SV machines, we now summarize empirical findings and theoretical developments which were to follow. We cannot report all contributions that have advanced the state of the art in SV learning since the time the algorithm was first proposed. Not even the present book can do this job, let alone a single section. Presently, we merely give a concise overview.

By the use of kernels, the optimal margin classifier was turned into a classifier which became a serious competitor of high-performance classifiers. Surprisingly, it was noticed that when different kernel functions are used in SV machines (specifically, (1.21), (1.30), and (1.31)), they lead to very similar classification accuracies and SV sets (Schölkopf et al., 1995). In this sense, the SV set seems to characterize (or *compress*) the given task in a manner which up to a certain degree is independent of the type of kernel (i.e. the type of classifier) used.

Initial work at AT&T Bell Labs focused on OCR (optical character recognition), a problem where the two main issues are classification accuracy and classification speed. Consequently, some effort went into the improvement of SV machines on these issues, leading to the *Virtual SV* method for incorporating prior knowledge about transformation invariances by transforming SVs, and the *Reduced Set* method for speeding up classification. This way, SV machines became competitive with the best available classifiers on both OCR and object recognition tasks (Schölkopf et al., 1996a; Burges, 1996; Burges and Schölkopf, 1997; Schölkopf, 1997). Two years later, the above are still topics of ongoing research, as shown by chapter 16 and (Schölkopf et al., 1998b), proposing alternative Reduced Set methods, as well as by chapter 7 and (Schölkopf et al., 1998d), constructing kernel functions which incorporate prior knowledge about a given problem.

Another initial weakness of SV machines, less apparent in OCR applications which are characterized by low noise levels, was that the size of the quadratic programming problem scaled with the number of Support Vectors. This was due to the fact that in (1.33), the quadratic part contained at least all SVs — the common practice was to extract the SVs by going through the training data in chunks while regularly testing for the possibility that some of the patterns that were initially not identified as SVs turn out to become SVs at a later stage (note that without chunking, the size of the matrix would be $\ell \times \ell$, where ℓ is the number of all training examples). What happens if we have a high-noise problem? In this case, many of the slack variables ξ_i will become nonzero, and all the corresponding examples will become SVs. For this case, a decomposition algorithm was proposed (Osuna et al., 1997a), which is based on the observation that not only can we leave out the non-SV examples (i.e. the $\mathbf{x}_i$ with $\alpha_i = 0$) from the current chunk, but also some of the SVs, especially those that hit the upper boundary (i.e. $\alpha_i = C$). In fact, one can use chunks which do not even contain all SVs, and maximize over the corresponding sub-problems. Chapter 12 explores an extreme case, where the sub-problems are chosen so small that one can solve them analytically. Most of the current implementations use larger subproblems, and employ some quadratic optimizer to solve these problems. Among the optimizers used are LOQO (Vanderbei, 1997), MINOS (Murtagh and Saunders, 1993), and variants of conjugate gradient descent, such as the optimizers of Bottou (cf. Saunders et al., 1998) and Burges (1998). Several public domain SV packages and optimizers are listed on the web page http://svm.first.gmd.de. For more details on the optimization problem, see chapters 10, 11, and 12.

Once the SV algorithm had been generalized to regression, researchers started applying it to various problems of estimating real-valued functions. Very good results were obtained on the Boston housing benchmark (Drucker et al. (1997) and chapter 17), and on problems of times series prediction (see Müller et al. (1997); Mukherjee et al. (1997), as well as chapters 13 and 14). Moreover, the SV method was applied to the solution of inverse function estimation problems (Vapnik et al. (1997); cf. chapters 3 and 18).

On the theoretical side, the least understood part of the SV algorithm initially was the precise role of the kernel, and how a certain kernel choice would influence the generalization ability. In that respect, the connection to regularization theory provided some insight. For kernel-based function expansions, one can show (Smola and Schölkopf, 1998b) that given a regularization operator P mapping the functions of the learning machine into some dot product space, the problem of minimizing the regularized risk

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2}\|Pf\|^2 \tag{1.51}$$

(with a regularization parameter $\lambda \geq 0$) can be written as a constrained optimization problem. For particular choices of the loss function, it further reduces to a SV type quadratic programming problem. The latter thus is not specific to SV machines, but is common to a much wider class of approaches. What gets lost in the general case, however, is the fact that the solution can usually be expressed in terms of a small number of SVs (cf. also Girosi (1998), who establishes a connection between SV machines and basis pursuit denoising (Chen et al., 1995)). This specific feature of SV machines is due to the fact that the type of regularization and the class of functions that the estimate is chosen from are intimately related (Girosi et al., 1993; Smola and Schölkopf, 1998a; Smola et al., 1998c): the SV algorithm is equivalent to minimizing the regularized risk on the set of functions

$$f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b, \tag{1.52}$$

provided that k and P are interrelated by

$$k(\mathbf{x}_i, \mathbf{x}_j) = ((Pk)(\mathbf{x}_i, .) \cdot (Pk)(\mathbf{x}_j, .)) \,. \tag{1.53}$$

To this end, k is chosen as a Green's function of P^*P, for in that case, the right hand side of (1.53) equals $(k(\mathbf{x}_i, .) \cdot (P^*Pk)(\mathbf{x}_j, .)) = (k(\mathbf{x}_i, .) \cdot \delta_{\mathbf{x}_j}(.)) = k(\mathbf{x}_i, \mathbf{x}_j)$. For instance, an RBF kernel thus corresponds to regularization with a functional containing a specific differential operator.

In SV machines, the kernel thus plays a dual role: firstly, it determines the class of functions (1.52) that the solution is taken from; secondly, via (1.53), the kernel determines the type of regularization that is used. The next question, naturally, is what type of regularization (i.e. kernel) we should use in order to get the best generalization performance — this is treated in chapter 9.

We conclude this section by noticing that the kernel method for computing dot products in feature spaces is not restricted to SV machines. Indeed, it has been used in 1996 to develop nonlinear generalizations of algorithms such as PCA (chapter 20), and a number of researchers have followed this example.

1.7 Notation

We conclude the introduction with a list of symbols which are used throughout the book, unless stated otherwise.

$\mathbb{R}$	the set of reals
$\mathbb{N}$	the set of natural numbers
k	Mercer kernel
F	feature space
N	dimensionality of input space
$\mathbf{x}_i$	input patterns
y_i	target values, or (in pattern recognition) classes
ℓ	number of training examples
$\mathbf{w}$	weight vector
b	constant offset (or threshold)
h	VC dimension
ε	parameter of the ε-insensitive loss function
α_i	Lagrange multiplier
$\boldsymbol{\alpha}$	vector of all Lagrange multipliers
ξ_i	slack variables
Q	Hessian of the quadratic program
$(\mathbf{x} \cdot \mathbf{y})$	dot product between patterns $\mathbf{x}$ and $\mathbf{y}$
$\|.\|$	2-norm (Euclidean distance), $\|\mathbf{x}\| := \sqrt{(\mathbf{x} \cdot \mathbf{x})}$
ln	logarithm to base e
$\log_2$	logarithm to base 2

2 Roadmap

The overall structure of this collection mirrors the development of SV learning as a field. We start with *theory*, then move on to *implementations*, to *applications*, and conclude with *extensions* of the original algorithm. We now give a brief roadmap of what the respective chapters are about.

2.1 Theory

Just as this collection of chapters should start with the theoretical ones, there is a natural beginning to the theoretical part. We start with Vladimir Vapnik, who can be singled out for his contributions to both the general development of the field and in particular of SV machines. His chapter on **Three Remarks on the Support Vector Method of Function Estimation** gives a concise authoritative account of the conceptual basis of SV machines, and explains an idea for improving generalization ability by taking into account the test points (but not their labels) during training (he calls this method *transduction*). Moreover, he shows how classical problems of statistics, such as conditional density estimation and conditional probability estimation, can be dealt with by treating them as inverse problems to be solved with SV methods. This approach is very much in line with his philosophy to solve the problem that one is interested in directly, rather than trying to solve harder problems as intermediate steps (e.g. by estimating a conditional density as the ratio of two estimated densities).

chapter 3

The seminal work of Vapnik, Chervonenkis and others has sparked a whole research field which studies the generalization ability of learning machines using various capacity concepts and different ways to make predictions about the accuracy of learning machines. Peter Bartlett and John-Shawe-Taylor, two leading researchers in this field, give a comprehensive discussion of the **Generalization Performance of Support Vector Machines and Other Pattern Classifiers**. To this end, they undertook the work of translating a variety of results from the learning theory community into a language that should be comprehensible to the emerging SV community. In particular, they describe the effect of large margins, crucial to SV machines, on the performance of classifiers. The same aspect is discussed by Nello Cristianini and John Shawe-Taylor in their work on **Bayesian Voting Schemes and Large Margin Classifiers**. Their accomplishment lies in the fact that they

chapter 4

chapter 5

have established a potentially very fruitful link to another field of machine learning which has recently attracted considerable attention, namely the field of Bayesian methods.

SV machines, and other kernel based methods, did not emerge in a vacuum. In particular, reproducing kernel Hilbert spaces (RKHS) provide a natural framework for SVMs, and show clearly how SVMs relate to other RKHS approaches, a point emphasized by Grace Wahba in **Support Vector Machines, Reproducing Kernel Hilbert Spaces and the Randomized GACV**. SVMs thus appear naturally when one chooses an appropriate likelihood functional in well known, penalized likelihood methods. In all such approaches, two (additive) terms appear in the objective function: a term that penalizes errors, and one that acts as a regularizer (and which, by the way, is also the norm of a vector in the RKHS). Loosely speaking, the latter is used to control the capacity, or variance, of the resulting learning machine. A critical choice is that of the trade-off between the size of the error penalty term versus that of the regularization ("smoothing") term. The Generalized Approximate Cross Validation technique of Xiang and Wahba (1996) is one such approach. Here, Wahba shows how the GACV method may be adapted to some support vector machines, thus giving one possible answer to some of the questions raised by practicioners (e.g. in chapter 14).

chapter 6

The following paper, Chris Burges's **Geometry and Invariance in Kernel Based Methods**, locks in on the kernel function and studies it in considerable detail. It consists of two parts. The first part explores the geometry of the manifold corresponding to the image of the input space under the kernel mapping. This leads to several interesting results: for example, the manifolds corresponding to homogeneous degree p polynomial kernels, acting on N dimensional data, are flat for $N = 2$ for all p, but have non-vanishing curvature for all $N > 2$ and $p > 1$. In the second part, the problem of adding local invariance to general kernel-based methods is discussed, using translation invariance in pixel images as a concrete example.

chapter 7

As described in the introduction, the VC dimension is the coarsest capacity concept of VC theory. Other concepts, such as the VC entropy or the Growth function, lead to tighter bounds on the generalization error, and could thus lead to more accurate tuning of learning machines. However, these concepts have so far either been believed to be hard to evaluate, or they have been neglected completely. The latter led to the belief that VC theory necessarily only gives bounds which disregard the actual structure of the problem at hand, and that the VC bounds (e.g. (1.4)) are so far from the actual behaviour of learning machines that they are useless. However, the utility of the bounds strongly depends on the capacity concepts used, and on the accuracy with which these concepts can be computed. In his contribution **On the Annealed VC Entropy for Margin Classifiers: A Statistical Mechanics Study**, Manfred Opper has accomplished the technically demanding job of computing a problem-dependent capacity measure for margin classifiers, for Gaussian input distribution. To this end, sophisticated methods from the statistical physics approach to learning theory had to be applied.

chapter 8

The dependence on the underlying distribution can lead to very precise capacity measures. This dependence is thus a strength of Opper's approach; however, his results strictly rely on the assumption of Gaussianity in the space where the hyperplane is constructed, i.e. in the feature space. Now suppose we do not want to make any such assumption, and still want to use a capacity concept which is more accurate than the VC dimension. Then we are essentially left with the Growth function or related concepts derived from taking suprema over covering numbers of function classes. This, again, had previously been considered too hard a quantity to compute. Bob Williamson et al., however, show that by considering the inverse of this quantity, the entropy numbers of suitably defined operators, one can tackle this problem. Crucial in an SV context, their work on **Entropy Numbers, Operators and Support Vector Kernels** takes into account the effect of the kernel, thus marrying the two main ingredients of SV machines — capacity control and the kernel mapping. Their hope is that this will indicate how to select the right kernel for a given problem.

chapter 9

2.2 Implementations

Despite the fact that the perceptron was invented in the sixties, interest in feed forward neural networks only took off in the eighties, due largely to a new training algorithm (actually invented by Werbos (1974), and rediscovered by several authors in the eighties, cf. Müller and Reinhardt (1990)). Backpropagation is conceptually simple and, perhaps more important, easy to implement. We believe that research into Support Vector Machines and related topics has been similarly hampered by the fact that training requires solving a quadratic programming problem, which, unless one happens to have a good QP package available, is a notoriously difficult business. In **Solving the Quadratic Programming Problem Arising in Support Vector Classification**, Linda Kaufman provides a review of the state of the art, and several valuable tips for the budding QP "programmer." A key point is to match the algorithm used to the problem at hand. For example, a largely separable, low noise problem will usually result in few support vectors: it then makes sense to use an algorithm that attempts to keep most of the Lagrange multipliers at zero. For very noisy problems, where most of the training data become support vectors, it makes more sense to consider approaches where most (or all) of the Lagrange multipliers are non-zero, such as interior point methods. One can also take advantage, in the high noise case, of the fact that many multipliers will be at their upper bound: this can be used, for example, to speed up core matrix multiplications.

chapter 10

Thorsten Joachims's contribution on **Making Large-Scale Support Vector Machine Learning Practical** is a hands-on approach to make chunk training work for large-scale real world problems. It builds on the idea of Osuna et al. (1997a) to split up the training set into small chunks and optimize only those variables while keeping the others fixed. The difficulty, however, consists in choosing a good

chapter 11

subset. Joachims gives a selection rule based on a linearization of the objective function similar to the work of Zoutendijk (1970). Yet even more can be gained by effectively restraining the subset of variables which are considered for chunking and taking into account the contribution of the other (then fixed) samples by modifying the overall linear and constant terms of the problem. This is both advantageous for data sets with low noise and data sets containing a large number of errors. An extensive experimental section, including text classification problems (in this domain, SVMs have been reported to exhibit excellent performance, cf. Joachims (1998); Dumais (1998)), backs the effect of these (and other) heuristics. Using tricks like kernel caching, Joachims reports examples where his method performs even faster than John Platt's latest algorithm. In **Fast Training of Support Vector Machines using Sequential Minimal Optimization**, SVMs may have found their backprop. Here is an algorithm that is easy to understand, easy to implement (the chapter even lists pseudocode), and trains an order of magnitude faster (in some cases) than a conjugate-gradient implementation of a QP optimizer. It is our hope that, due to its simplicity and speed, this algorithm will make SVMs accessible to a much larger group of users, and that the encouraging results will withstand the test of a wide range of data sets. Additionally, it seems likely that the algorithm can be modified to handle the case of regression estimation. Note also that it is expected that by using caching of kernel evaluations, it will be possible to further improve the speed of the algorithm.

chapter 12

2.3 Applications

Following these three chapters on SV implementations, we have three interesting papers on applications. Davide Mattera and Simon Haykin use **Support Vector Machines for Dynamic Reconstruction of a Chaotic System**. Also the chapter of Klaus Müller et al., **Using Support Vector Machines for Time Series Prediction**, employs SV regression for reconstruction of a chaotic system and time series prediction. Both works explore the use of different loss functions, most notably ones which are designed in the spirit of Huber's robust statistics (Huber, 1981). The chapters contain thorough experimental analyses on prediction problems, obtaining strong results. Müller et al. compare the SVM approach with that of RBF networks with adaptive centers and widths, a powerful technique for which they also give a useful, self-contained description. The results — a 29% improvement over the best known results on the Santa Fe time series set D benchmark — are clearly a strong endorsement of the approach. However, there are plenty of opportunities for further improvements. For example, how can one make the RBF centers adaptive, in the SVM approach? How can one best choose the SVM error penalty and regression tube sizes? Should one allow different error penalties for different input data (thus weighting different pieces of the time series differently), or allow varying tube sizes? These are all areas of active investigation. The chapters discuss these questions, and propose methods and heuristics for selecting the SV

chapters 13, 14

machine parameters C and ε. Nevertheless, a fair amount of theoretical work remains to be done on these issues.

chapter 15

The following chapter, dealing with **Pairwise Classification and Support Vector Machines**, describes application-oriented work on the optical character recognition (OCR) problem that already served as a testbed in the beginnings of SV research (Cortes and Vapnik, 1995; Schölkopf et al., 1995). In it, Ulrich Kreßel gives an experimental investigation of multi-class pattern recognition by means of pairwise classification. This way, he was able to set a new record on the standard OCR benchmark that he and his colleagues at Daimler-Benz Research had used for a number of years. The question whether this is due to the use of SVMs in general, or due to his specific pairwise approach, cannot be answered conclusively at present. In any case, the pairwise method seems an interesting direction for research, especially on large databases.

2.4 Extensions

chapter 16

For applications where the time it takes to compute a prediction is critical, one needs methods for simplifying the representation of the estimated function. In their contribution on **Reducing the Run-time Complexity in Support Vector Machines**, Edgar Osuna and Federico Girosi describe three such methods. They start by summarizing the so-called reduced set technique, which has been shown to yield substantial speed-ups in OCR applications (Burges, 1996; Burges and Schölkopf, 1997). Following this, they describe two novel methods. First, they propose the use of SV regression to approximate SV expansions using a smaller number of terms. Second, they reformulate the training problem such that it automatically produces expansions which are more sparse, while yielding the same solution. Their methods, which are experimentally validated, are of particular use in situations where many of the Lagrange multipliers hit the upper boundary, such as in noisy applications.

chapter 17

The chapter **Support Vector Regression with ANOVA Decomposition Kernels**, contributed by Mark Stitson et al., considers one particular kernel function, generating the ANOVA decompositions known from classical statistics in an efficient way. Encouraging experimental results on the Boston housing real-world regression problem are reported.

chapter 18

In **Support Vector Density Estimation**, Jason Weston et al. propose SV algorithms for SV density estimation. This is done by considering the problem of density estimation as a problem of solving a linear integral equation: the integral over the density is the distribution function, and the data can be used to give an empirical distribution function. The problem of density estimation differs from the regression problem in several ways, one of them being that the estimated density is required to be positive. These difficulties lead the authors to propose several modifications of the SV algorithm, using linear programming techniques, dictionaries of kernels, and different loss functions.

chapter 19

In **Combining Support Vector and Mathematical Programming Methods for Classification**, Kristin Bennett relates the two domains of Mathematical Programming and Support Vector Classification. By doing so, she is able to construct SV decision trees. Moreover, she analyses a regularization term which is new to the SV field, similar in construction to the one introduced in chapter 18 for density estimation. This is a potentially very fruitful approach as it may lead to linear programming algorithms that could be solved with lower computational cost. The method is extended to multicategory classification, and should be compared to the ones of Weston and Watkins (1998) and chapter 15. The final section presents an extension of the standard SV algorithm to the case of transduction (cf. chapter 3). It is shown empirically that overall optimization of the margin can lead to improved results.

chapter 20

The book concludes with a chapter which uses one of the main ideas of SV learning and transfers it to a rather different domain of learning. Using the kernel trick, Bernhard Schölkopf et al. generalized one of the most widely used algorithms for unsupervised data analysis, linear principal component analysis, to the nonlinear case, obtaining **Kernel Principal Component Analysis**. Experimental results show that this leads to powerful nonlinear feature extractors.

References

H. D. I. Abarbanel. *Analysis of Observed Chaotic Data.* Springer Verlag, New York, 1996.

M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821 – 837, 1964.

N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale–sensitive Dimensions, Uniform Convergence, and Learnability. *Journal of the ACM*, 44(4):615–631, 1997.

E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147:181–210, 1995.

E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 1998. To appear.

M. Anthony. Probabilistic analysis of learning in artificial neural networks: The PAC model and its variants. *Neural Computing Surveys*, 1:1–47, 1997. http://www.icsi.berkeley.edu/~jagota/NCS.

M. Anthony and N. Biggs. *Computational Learning Theory*, volume 30 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1992.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337 – 404, 1950.

R. Ash. *Information Theory.* Interscience Publishers, New York, 1965.

P. L. Bartlett. Pattern classification in neural networks. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998a.

P. L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998b.

P. L. Bartlett, P. Long, and R. C. Williamson. Fat–Shattering and the Learnability of Real–Valued Functions. *Journal of Computer and System Sciences*, 52(3):434–452, 1996.

Y. Bengio, Y. LeCun, and D. Henderson. Globally trained handwritten word recognizer using spatial representation, convolutional neural networks and hidden markov models. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 937–944, 1994.

K. P. Bennett. Decision tree construction via linear programming. In M. Evans, editor, *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference*, pages 97–101, Utica, Illinois, 1992.

K. P. Bennett and J. A. Blue. A support vector machine approach to decision trees. In *Proceedings of IJCNN'98*, pages 2396 – 2401, Anchorage, Alaska, 1997.

K. P. Bennett and E. J. Bredensteiner. A parametric optimization method for machine learning. *INFORMS Journal on Computing*, 9(3):311–318, 1997.

K. P. Bennett and E. J. Bredensteiner. Geometry in learning. In C. Gorini, E. Hart, W. Meyer, and T. Phillips, editors, *Geometry at Work*, Washington, D.C., 1998. Mathematical Association of America. Available http://www.math.rpi.edu/~bennek/geometry2.ps.

K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. Unpublished manuscript based on talk given at Machines That Learn Conference, Snowbird, 1998.

K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

K. P. Bennett and O. L. Mangasarian. Multicategory separation via linear programming. *Optimization Methods and Software*, 3:27–39, 1993.

K. P. Bennett and O. L. Mangasarian. Serial and parallel multicategory discrimination. *SIAM Journal on Optimization*, 4(4):722–734, 1994.

K. P. Bennett, D. H. Wu, and L. Auslender. On support vector decision trees for database marketing. R.P.I. Math Report No. 98-100, Rensselaer Polytechnic Institute, Troy, NY, 1998.

L. Bernhardt. Zur Klassifizierung vieler Musterklassen mit wenigen Merkmalen. In H. Kazmierczak, editor, *5. DAGM Symposium: Mustererkennung 1983*, pages 255 – 260, Berlin, 1983. VDE-Verlag.

D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1995.

M. Bierlaire, Ph. Toint, and D. Tuyttens. On iterative algorithms for linear least squares problems with bound constraints. *Linear Alebra Appl.*, pages 111–143, 1991.

C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, and T. Vetter. Comparison of view-based object recognition algorithms using realistic 3D models. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, editors, *Artificial Neural Networks — ICANN'96*, pages 251 – 256, Berlin, 1996. Springer Lecture Notes in Computer Science, Vol. 1112.

J. A. Blue. *A Hybrid of Tabu Search and Local Descent Algorithms with Applications in Artificial Intelligence*. PhD thesis, Rensselaer Polytechnic Institute, 1998.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

W. M. Boothby. *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, 2nd edition, 1986.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Müller, E. Säckinger, P. Simard, and V. Vapnik. Comparison of classifier methods: a case study in handwritten digit recognition. In *Proceedings of the 12th International Conference on Pattern Recognition and Neural Networks, Jerusalem*, pages 77 – 87. IEEE Computer Society Press, 1994.

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

P. S. Bradley, U. M. Fayyad, and O. L. Mangasarian. Data mining: Overview and optimization opportunities. Technical Report Mathematical Programming Technical Report 98-01, University of Wisconsin-Madison, 1998. Submitted for publication.

P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. Technical Report Mathematical Programming Technical Report 98-03, University of Wisconsin-Madison, 1998a. To appear in ICML-98.

P. S. Bradley and O. L. Mangasarian. Massive data discrimination via linear support vector machines. Technical Report Mathematical Programming Technical Report 98-05, University of Wisconsin-Madison, 1998b. Submitted for publication.

P. S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. Technical Report 95-21, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1995. To appear in *INFORMS Journal on Computing* 10, 1998.

E. J. Bredensteiner. *Optimization Methods in Data Mining and Machine Learning*. PhD thesis, Rensselaer Polytechnic Institute, 1997.

E. J. Bredensteiner and K. P. Bennett. Feature minimization within decision trees. *Computational Optimization and Applications*, 10:110–126, 1997.

E. J. Bredensteiner and K. P. Bennett. Multicategory classification by support vector machines. *Computational Optimization and Applications*, 1998. To appear.

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

L. Breiman. Bagging predictors. Technical Report 421, Department of Statistics, UC Berkeley, 1994. ftp://ftp.stat.berkeley.edu/pub/tech-reports/421.ps.Z.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth International, California, 1984.

R. Brown, P. Bryant, and H. D. I. Abarbanel. Computing the lyapunov spectrum of a dynamical system from observed time-series. *Phys. Rev. Lett.*, 43(6):2787–2806, 1991.

J. R. Bunch and L. Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Mathematics of Computation*, 31:163–179, 1977.

C. J. C. Burges. Simplified support vector decision rules. In L. Saitta, editor, *Proceedings, 13th Intl. Conf. on Machine Learning*, pages 71–77, San Mateo, CA, 1996. Morgan Kaufmann.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):1–47, 1998.

C. J. C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector learning machines. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 375–381, Cambridge, MA, 1997. MIT Press.

C. J. C. Burges and V. Vapnik. A new method for constructing artificial neural networks: Interim technical report, ONR contract N00014-94-c-0186. Technical report, AT&T Bell Laboratories, 1995.

B. Carl. Inequalities of Bernstein-Jackson-type and the degree of compactness of operators in Banach spaces. *Annales de l'Institut Fourier*, 35(3):79–118, 1985.

B. Carl and I. Stephani. *Entropy, compactness, and the approximation of operators.* Cambridge University Press, Cambridge, UK, 1990.

Y. Censor. Row-action methods for huge and sparse systems and their applications. *SIAM Review*, 23(4):444–467, 1981.

Y. Censor and A. Lent. An iterative row-action method for interval convex programming. *J. Optimization Theory and Applications*, 34(3):321–353, 1981.

S. Chen. *Basis Pursuit.* PhD thesis, Department of Statistics, Stanford University, 1995.

S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. Technical Report 479, Department of Statistics, Stanford University, 1995.

C. R. Chester. *Techniques in Partial Differential Equations.* McGraw Hill, 1971.

E. T. Copson. *Metric Spaces.* Cambridge University Press, 1968.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273 – 297, 1995.

R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience Publishers, Inc, New York, 1953.

T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Elect. Comp.*, 14:326–334, 1965.

D. Cox and F. O'Sullivan. Asymptotic analysis of penalized likelihood and related estimators. *Ann. Statist.*, 18:1676–1695, 1990.

CPLEX Optimization Incorporated, Incline Village, Nevada. *Using the CPLEX Callable Library*, 1994.

P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.*, 31:377–403, 1979.

N. Cristianini, J. Shawe-Taylor, and P. Sykacek. Bayesian classifiers are large margin hyperplanes in a hilbert space. In J. Shavlik, editor, *Machine Learning: Proceedings of the Fifteenth International Conference*, San Francisco, CA, 1998. Morgan Kaufmann.

K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks.* Wiley, New York, 1996.

K. Dodson and T. Poston. *Tensor Geometry.* Springer-Verlag, 2nd edition, 1991.

H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, Cambridge, MA, 1997. MIT Press.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

S. Dumais. Using SVMs for text categorization. *IEEE Intelligent Systems*, 13(4), 1998. In: M.A. Hearst, B. Schölkopf, S. Dumais, E. Osuna, and J. Platt: Trends and Controversies — Support Vector Machines.

N. Dunford and J. T. Schwartz. *Linear Operators Part II: Spectral Theory, Self Adjoint Operators in Hilbert Space.* Number VII in Pure and Applied Mathematics. John Wiley & Sons, New York, 1963.

J. P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Rev. Modern Phys.*, 57(3):617–656, 1985.

K. Efetov. *Supersymmetry in Disorder and Chaos.* Cambridge University Press, Cambridge, 1997.

A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82: 247–261, 1989.

L. Elden and L. Wittmeyer-Koch. *Numerical Analysis: An Introduction.* Academic Press, Cambrigde, 1990.

R. Fourer, D. Gay, and B. Kernighan. *AMPL A Modeling Language for Mathematical Programming.* Boyd and Frazer, Danvers, Massachusetts, 1993.

J. H. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics and Stanford Linear Accelerator Center, Stanford University, 1996.

E. Gardner. The space of interactions in neural networks. *Journal of Physics A*, 21:257–70, 1988.

P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. Technical Report NA-97-2, Dept. of Mathematics, U.C. San Diego, 1997.

P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, 1981.

D. Girard. Asymptotic optimality of the fast randomized versions of GCV and C_L in ridge regression and regularization. *Ann. Statist.*, 19:1950–1963, 1991.

D. Girard. Asymptotic comparison of (partial) cross-validation, GCV and randomized GCV in nonparametric regression. *Ann. Statist.*, 126:315–334, 1998.

F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998.

F. Girosi, M. Jones, and T. Poggio. Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines. A.I. Memo No. 1430, MIT, 1993.

F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

F. Glover. Improved linear programming models for discriminant analysis. *Decision Sciences*, 21:771–785, 1990.

W. Gochet, A. Stam, V. Srinivasan, and S. Chen. Multigroup discriminant analysis using linear programming. *Operations Research*, 45(2):213–559, 1997.

H. Goldstein. *Classical Mechanics*. Addison-Wesley, Reading, MA, 1986.

M. Golea, P. L. Bartlett, W. S. Lee, and L. Mason. Generalization in decision trees and DNF: Does size matter? In *Advances in Neural Information Processing Systems 10*, 1998.

G. Golub and U. von Matt. Generalized cross-validation for large-scale problems. *J. Comput. Graph. Statist.*, 6:1–34, 1997.

J. Gong, G. Wahba, D. Johnson, and J. Tribbia. Adaptive tuning of numerical weather prediction models: simultaneous estimation of weighting, smoothing and physical parameters. *Monthly Weather Review*, 125:210–231, 1998.

Y. Gordon, H. König, and C. Schütt. Geometric and probabilistic estimates for entropy and approximation numbers of operators. *Journal of Approximation Theory*, 49:219–239, 1987.

T. Graepel and K. Obermayer. Fuzzy topographic kernel clustering. In W. Brauer, editor, *Proceedings of the 5th GI Workshop Fuzzy Neuro Systems '98*, pages 90 – 97, 1998.

R. E. Greene. *Isometric Embeddings of Riemannian and Pseudo-Riemannian*

Manifolds. American Mathematical Society, 1970.

C. Gu and G. Wahba. Semiparametric analysis of variance with tensor product thin plate splines. *J. Royal Statistical Soc. Ser. B*, 55:353–368, 1993.

L. Gurvits. A note on a scale-sensitive dimension of linear bounded functionals in banach spaces. In *Proceedings of Algorithm Learning Theory, ALT-97*, 1997. Also: NECI Technical Report, 1997.

I. Guyon, B. Boser, and V. Vapnik. Automatic capacity tuning of very large VC-dimension classifiers. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 147–155. Morgan Kaufmann, San Mateo, CA, 1993.

M. Hamermesh. *Group theory and its applications to physical problems.* Addison Wesley, Reading, MA, 2 edition, 1962. Reprint by Dover, New York, NY.

D. Harrison and D. L. Rubinfeld. Hedonic prices and the demand for clean air. In *J. Environ. Economics & Management*, volume 5, pages 81–102, 1978. Original source of the Boston Housing data, actually from ftp://ftp.ics.uci.com/pub/machine-learning-databases/housing.

T. Hastie and W. Stuetzle. Principal curves. *JASA*, 84:502 – 516, 1989.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability.* Chapman & Hall, London, 1990.

S. Haykin. *Neural Networks : A Comprehensive Foundation.* Macmillan, New York, 1994.

S. Haykin, S. Puthusserypady, and P. Yee. Reconstruction of underlying dynamics of an observed chaotic process. Technical Report 353, Comm. Res. Lab., McMaster University, 1997.

C. Hildreth. A quadratic programming procedure. *Naval Research Logistics Quarterly*, 4:79–85, 1957.

T. K. Ho and E. Kleinberg. Building projectable classifiers for arbitrary complexity. In *Proceedings of the 12th International Conference on Pattern Recognition, Vienna*, pages 880–885, 1996.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441 and 498–520, 1933.

P. J. Huber. Robust statistics: a review. *Ann. Statist.*, 43:1041, 1972.

P. J. Huber. *Robust Statistics.* John Wiley and Sons, New York, 1981.

A. M. Hughes. *The Complete Database Marketer.* Irwin Prof. Publishing, Chicago, 1996.

M. Hutchinson. A stochastic estimator for the trace of the influence matrix for Laplacian smoothing splines. *Commun. Statist.-Simula.*, 18:1059–1076, 1989.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. Technical Report 23, LS VIII, University of Dortmund, 1997.

T. Joachims. Text categorization with support vector machines. In *European Conference on Machine Learning (ECML)*, 1998.

I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, New York, 1986.

J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4):549–562, 1995.

K. Karhunen. Zur Spektraltheorie stochastischer Prozesse. *Ann. Acad. Sci. Fenn.*, 34, 1946.

W. Karush. Minima of functions of several variables with inequalities as side constraints. Master's thesis, Dept. of Mathematics, Univ. of Chicago, 1939.

M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2):115–141, 1994.

M. Kennel, R. Brown, and H. D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A.*, 45:3403–3411, 1992.

G. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation of stochastic processes and smoothing by splines. *Ann. Math. Statist.*, 41:495–502, 1970.

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

M. Kirby and L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

J. Kohlmorgen, K.-R. Müller, and K. Pawelzik. Analysis of drifting dynamics with neural network hidden markov models. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, Cambridge, MA, 1998. MIT Press.

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59 – 69, 1982.

A. N. Kolmogorov and S. V. Fomin. *Introductory Real Analysis.* Prentice-Hall, Inc., 1970.

H. König. *Eigenvalue Distribution of Compact Operators.* Birkhäuser, Basel, 1986.

U. Kreßel. The impact of the learning–set size in handwritten–digit recognition. In T. Kohonen et al., editor, *Artificial Neural Networks — ICANN'91*, pages 1685 – 1689, Amsterdam, 1991. North–Holland.

U. Kreßel. Polynomial classifiers and support vector machines. In W. Gerstner et al., editor, *Artificial Neural Networks — ICANN'97*, pages 397 – 402, Berlin, 1997. Springer Lecture Notes in Computer Science, Vol. 1327.

U. Kreßel and J. Schürmann. Pattern classification techniques based on function approximation. In H. Bunke and P.S.P. Wang, editors, *Handbook on Optical Character Recognition and Document Analysis*, pages 49 – 78. World Scientific Publishing Company, Singapore, 1997.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proc.* 2^{nd} *Berkeley Symposium on Mathematical Statistics and Probabilistics*, pages 481–492, Berkeley, 1951. University of California Press.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. J. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541 – 551, 1989.

Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Müller, E. Säckinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In F. Fogelman-Soulié and P. Gallinari, editors, *Proceedings ICANN'95 — International Conference on Artificial Neural Networks*, volume II, pages 53 – 60, Nanterre, France, 1995. EC2. The MNIST benchmark data is available from http://www.research.att.com/∼yann/ocr/mnist/.

W. S. Lee, P. L. Bartlett, and R. C. Williamson. The importance of convexity in learning with squared loss. *IEEE Transactions on Information Theory*, 1998. to appear.

K. C. Li. Asymptotic optimality of C_L and generalized cross validation in ridge regression with application to spline smoothing. *Ann. Statist.*, 14:1101–1112, 1986.

W. Liebert, K. Pawelzik, and H. G. Schuster. Optimal embeddings of chaotic attractors from topological considerations. *Europhys. Lett.*, 14:521 – 526, 1991.

B. Lillekjendlie, D. Kugiumtzis, and N. Christophersen. Chaotic time series: Part ii. system identification and prediction. *Modeling, Identification and Control*, 15 (4):225–243, 1994.

N. Littlestone and M. Warmuth. Relating data compression and learnability. Technical report, University of California Santa Cruz, 1986.

E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.

G. Loy and P. L. Bartlett. Generalization and the size of the weights: an experimental study. In *Proceedings of the Eighth Australian Conference on Neural Networks*, pages 60–64, 1997.

D. J. C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4:720–736, 1992a.

D. J. C. MacKay. A practical Bayesian framework for backprop networks. *Neural Computation*, 4:448–472, 1992b.

M. C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science*, 197:287–289, 1977.

S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

O. L. Mangasarian. Multi-surface method of pattern separation. *IEEE Transactions on Information Theory*, IT-14:801–807, 1968.

O. L. Mangasarian. Misclassification minimization. *J. Global Optimization*, 5:309–323, 1994.

O. L. Mangasarian. Mathematical programming in data mining. *Data Mining and Knowledge Discovery*, 42(1):183–201, 1997.

O. L. Mangasarian and R. Meyer. Nonlinear perturbations of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, 1979.

O. L. Mangasarian, R. Setiono, and W. H. Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. In *Proceedings of the Workshop on Large-Scale Numerical Optimization, 1989*, pages 22–31, Philadelphia, Pennsylvania, 1990. SIAM.

O. L. Mangasiarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13:444–452, 1965.

J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, A 209:415–446, 1909.

C. J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1998. [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear prediction of chaotic time series using a support vector machine. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, pages 511 – 520, New York, 1997. IEEE.

B. Müller and J. Reinhardt. *Neural Networks: An Introduction.* Springer Verlag, 1990.

K.-R. Müller, J. Kohlmorgen, and K. Pawelzik. Analysis of switching dynamics with competing neural networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E78–A(10):1306–1315, 1995.

K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks — ICANN'97*, pages 999 – 1004, Berlin, 1997. Springer Lecture Notes in Computer Science, Vol. 1327.

P.M. Murphy and D.W. Aha. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, California, 1992.

B. A. Murtagh and M. A. Saunders. MINOS 5.4 user's guide. Technical Report SOL 83.20, Stanford University, 1993.

K. G. Murthy. *Linear Programming.* John Wiley & Sons, New York, New York, 1983.

S. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

I. Nagayama and N. Akamatsu. Approximation of chaotic behavior by using neural network. *IEICE Trans. Inf. & Syst.*, E77-D(4), 1994.

J. Nash. The embedding problem for riemannian manifolds. *Annals of Mathematics*, 63:20 – 63, 1956.

R. Neal. Priors for infinite networks. Technical Report CRG-TR-94-1, Dept. of Computer Science, University of Toronto, 1994.

R. Neal. *Bayesian Learning in Neural Networks.* Springer Verlag, 1996.

N. J. Nilsson. *Learning machines: Foundations of Trainable Pattern Classifying Systems.* McGraw-Hill, 1965.

E. Oja. A simplified neuron model as a principal component analyzer. *J. Math. Biology*, 15:267 – 273, 1982.

P. J. Olver. *Applications of Lie Groups to Differential Equations.* Springer-Verlag, 1986.

M. Opper. Learning in neural networks: Solvable dynamics. *Europhysics Letters*, 8 (4):389–392, 1989.

M. Opper and W. Kinzel. Physics of generalization. In E. Domany J.L. van Hemmen and K. Schulten, editors, *Physics of Neural Networks III.* Springer Verlag, New York, 1996.

M. Opper, P. Kuhlmann, and A. Mietzner. Convexity, internal representations and the statistical mechanics of neural networks. *Europhysics Letters*, 37(1):31–36, 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, 1997.

E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, pages 276 – 285, New York, 1997a. IEEE.

E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. AI Memo 1602, Massachusetts Institute of Technology, 1997b.

E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proc. Computer Vision and Pattern Recognition '97*, pages 130–136, 1997c.

N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Phys. Rev. Lett.*, 45:712–716, 1980.

E. Parzen. An approach to time series analysis. *Ann. Math. Statist.*, 32:951–989, 1962.

E. Parzen. Statistical inference on time series by rkhs methods. In R. Pyke, editor, *Proceedings 12th Biennial Seminar*, Montreal, 1970. Canadian Mathematical Congress. 1-37.

K. Pawelzik, J. Kohlmorgen, and K.-R. Müller. Annealed competition of experts for a segmentation and classification of switching dynamics. *Neural Computation*, 8 (2):342–358, 1996a.

K. Pawelzik, K.-R. Müller, and J. Kohlmorgen. Prediction of mixtures. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, editors, *Artificial Neural Networks — ICANN'96*, pages 127–133, Berlin, 1996b. Springer Lecture Notes in Computer Science, Vol. 1112.

K. Pearson. On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2 (sixth series):559–572, 1901.

J. C. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.

J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, 1998.

T. Poggio. On optimal nonlinear associative recall. *Biological Cybernetics*, 19: 201–209, 1975.

T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), 1990a.

T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990b.

M. Pontil and A. Verri. Properties of support vector machines. *Neural Computation*, 10:955 – 974, 1997.

M. J. D. Powell. Radial basis functions for multivariable interpolation: A review. In *Algorithms for Approximation, J.C. Mason and M.G. Cox (Eds.)*, pages 143–167. Oxford Clarendon Press, 1987.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press, Cambridge, 1992.

J. C. Principe and J. M. Kuo. Dynamic modeling of chaotic time series with neural networks. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Precessing Systems 7*, San Mateo, CA, 1995. Morgan Kaufmann Publishers.

R.T. Prosser. The ε–Entropy and ε–Capacity of Certain Time–Varying Channels. *Journal of Mathematical Analysis and Applications*, 16:553–573, 1966.

J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

C. Rasmussen. *Evaluation of Gaussian Processes and Other Methods for Non-Linear Regression*. PhD thesis, Department of Computer Science, University of Toronto, 1996. ftp://ftp.cs.toronto.edu/pub/carl/thesis.ps.gz.

H. J. Ritter, T. M. Martinetz, and K. J. Schulten. *Neuronale Netze: Eine Einführung in die Neuroinformatik selbstorganisierender Abbildungen*. Addison-Wesley, Munich, Germany, 1990.

A. Roy, S. Govil, and R. Miranda. An algorithm to generate radial basis function (RBF)-like nets for classification problems. *Neural Networks*, 8(2):179–202, 1995.

A. Roy, L. S. Kim, and S. Mukhopadhyay. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. *Neural Networks*, 6:535–545, 1993.

A. Roy and S. Mukhopadhyay. Iterative generation of higher-order nets in polynomial time using linear programming. *IEEE Transactions on Neural Networks*, 8 (2):402–412, 1997.

M. A. Saunders S. S. Chen, D. L. Donoho. Atomic decomposition by basis pursuit. Technical Report Dept. of Statistics Technical Report, Stanford University, 1996.

S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England, 1988.

T. D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward network. *Neural Networks*, 2:459–473, 1989.

T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *J. Stat. Phys.*, 65:579–616, 1991.

C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A. Smola. Support vector machine - reference manual. Technical Report CSD-TR-98-03, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998. TR available as http://www.dcs.rhbnc.ac.uk/research/compint/areas/comp_learn/sv/pub/report98-03.ps; SVM available at http://svm.dcs.rhbnc.ac.uk/.

R. J. Schalkoff. *Digital Image Processing and Computer Vision*. John Wiley and Sons, Inc., 1989.

R. Schapire, Y. Freund, P. Bartlett, and W. Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 1998. (To appear. An earlier version appeared in: D.H. Fisher, Jr. (ed.), Proceedings ICML97, Morgan Kaufmann.).

M. Schmidt and H. Gish. Speaker identification via support vector classifiers. In *Proc. ICASSP '96*, pages 105–108, Atlanta, GA, 1996.

I. Schoenberg. Positive definite functions on spheres. *Duke Math. J.*, 9:96–108, 1942.

B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings, First International Conference on Knowledge Discovery & Data Mining*. AAAI Press, Menlo Park, CA, 1995.

B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, Munich, 1997.

B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Support vector regression with automatic accuracy control. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks*, Perspectives in Neural Computing, Berlin, 1998a. Springer Verlag. In press.

B. Schölkopf, C. Burges, and V. Vapnik. Incorporating invariances in support vector learning machines. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, editors, *Artificial Neural Networks — ICANN'96*, pages 47 – 52, Berlin, 1996a. Springer Lecture Notes in Computer Science, Vol. 1112.

B. Schölkopf, P. Knirsch, A. Smola, and C. Burges. Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces. In *20. DAGM Symposium Mustererkennung*, Lecture Notes in Computer Science, Berlin, 1998b. Springer. To appear.

B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K.-R. Müller. Kernel PCA pattern reconstruction *via* approximate pre-images. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks*, Perspectives in Neural Computing, Berlin, 1998c. Springer Verlag. In press.

B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, Cambridge, MA, 1998d. MIT Press.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report 44, Max-Planck-Institut für biologische Kybernetik, 1996b.

B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks — ICANN'97*, pages 583 – 588, Berlin, 1997a. Springer Lecture Notes in Computer Science, Vol. 1327.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299 – 1319, 1998e.

B. Schölkopf, A. Smola, K.-R. Müller, C. Burges, and V. Vapnik. Support vector methods in learning and feature extraction. In T. Downs, M. Frean, and M. Gallagher, editors, *Proceedings of the Ninth Australian Conference on Neural Networks*, pages 72 – 78, Brisbane, Australia, 1998f. University of Queensland.

B. Schölkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Trans. Sign. Processing*, 45:2758 – 2765, 1997b.

J. C. Schouten, F. Takens, and C. M. van den Bleek. Estimation of the dimension of a noisy attractor. *Physical Review E*, 50(3):1851–1860, 1994.

J. Schürmann. *Pattern Classification: a unified view of statistical and neural approaches.* Wiley, New York, 1996.

D.W Scott. *Multivariate Density Estimation.* Wiley-Interscience, New York, 1992.

H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056–6091, 1992.

J. Shawe-Taylor, P. Bartlett, R. Williamson, and M. Anthony. A framework for structural risk minimization. In *COLT*, 1996.

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.

J. Shawe-Taylor and N. Cristianini. Data-dependent structural risk minimisation for perceptron decision trees. In *Advances in Neural Information Processing Systems 10*, 1998.

P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 50–58, San Mateo, CA, 1993. Morgan Kaufmann.

A. Skorokhod and M. Yadrenko. On absolute continuity of measures corresponding to homogeneous Gaussian fields. *Theory of Probability and its Applications*, XVIII:27–40, 1973.

F. W. Smith. Pattern classifier design by linear programming. *IEEE Transactions on Computers*, C-17:367–372, 1968.

A. Smola and B. Schölkopf. From regularization operators to support vector kernels. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, Cambridge, MA, 1998a. MIT Press.

A. Smola and B. Schölkopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 1998b. In press.

A. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11:637–649, 1998a.

A. Smola, B. Schölkopf, and K.-R. Müller. Convex cost functions for support vector regression. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks*, Perspectives in Neural Computing, Berlin, 1998b. Springer Verlag. In press.

A. Smola, B. Schölkopf, and K.-R. Müller. General cost functions for support vector regression. In T. Downs, M. Frean, and M. Gallagher, editors, *Proc. of the Ninth Australian Conf. on Neural Networks*, pages 79 – 83, Brisbane, Australia, 1998c. University of Queensland.

J. Stewart. Positive definite funcions and generalizations, an historical survey. *Rocky Mountain Journal of Mathematics*, 6(3):409–434, 1978.

M. Stitson, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins, and J. Weston. Support vector regression with ANOVA decomposition kernels. Technical Report CSD-TR-97-22, Royal Holloway, University of London, 1997.

F. Takens. Detecting strange attractors in fluid turbulence. In D. Rand and L.S. Young, editors, *Dynamical Systems and Turbulence*, pages 366–381. Springer-Verlag, Berlin, 1981.

M. Talagrand. The Glivenko–Cantelli problem, ten years later. *Journal of Theoretical Probability*, 9(2):371–384, 1996.

W. Thomas. Database marketing: Dual approach outdoes response modeling. *Database Marketing News*, page 26, 1996.

R. Vanderbei. Loqo: An interior point code for quadratic programming. Technical Report SOR 94-15, Princeton University, 1994.

R. J. Vanderbei. LOQO user's manual – version 3.10. Technical Report SOR-97-08, Princeton University, Statistics and Operations Research, 1997. Code available at http://www.princeton.edu/~rvdb/.

V. Vapnik. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982).

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

V. Vapnik. Structure of statistical learning theory. In A. Gammerman, editor, *Computational and Probabalistic Reasoning*, chapter 1. Wiley, Chichester, 1996.

V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998. forthcoming.

V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.

V. Vapnik and A. Chervonenkis. Uniform convergence of frequencies of occurence of events to their probabilities. *Dokl. Akad. Nauk SSSR*, 181:915 – 918, 1968.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2): 264–280, 1971.

V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow, 1974. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie-Verlag, Berlin, 1979).

V. Vapnik and A. Chervonenkis. Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and its Applications*, 26(3):532–553, 1981.

V. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 281–287, Cambridge, MA, 1997. MIT Press.

V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method.

Automation and Remote Control, 24, 1963.

V. Vapnik, E. Levin, and Y. Le Cun. Measuring the VC-dimension of a learning machine. *Neural Computation*, 6(5):851–876, 1994.

N. Ya. Vilenkin. *Special Functions and the Theory of Group Representations*, volume 22 of *Translations of Mathematical Monographs.* American Mathematical Society Press, Providence, NY, 1968.

M. Villalobos and G. Wahba. Inequality constrained multivariate smoothing splines with application to the estimation of posterior probabilities. *J. Am. Statist. Assoc.*, 82:239–248, 1987.

G. Wahba. Convergence rates of certain approximate solutions to Fredholm integral equations of the first kind. *Journal of Approximation Theory*, 7:167 – 185, 1973.

G. Wahba. Improper priors, spline smoothing and the problem of guarding against model errors in regression. *J. Roy. Stat. Soc. Ser. B*, 40:364–372, 1978.

G. Wahba. Spline interpolation and smoothing on the sphere. *SIAM J. Sci. Stat. Comput.*, 2:5–16, 1981.

G. Wahba. Constrained regularization for ill posed linear operator equations, with applications in meteorology and medicine. In S. Gupta and J. Berger, editors, *Statistical Decision Theory and Related Topics, III, Vol.2*, pages 383–418. Academic Press, 1982a.

G. Wahba. Erratum: Spline interpolation and smoothing on the sphere. *SIAM J. Sci. Stat. Comput.*, 3:385–386, 1982b.

G. Wahba. A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Ann. Statist.*, 13:1378–1402, 1985a.

G. Wahba. Multivariate thin plate spline smoothing with positivity and other linear inequality constraints. In E. Wegman and D. dePriest, editors, *Statistical Image Processing and Graphics*, pages 275–290. Marcel Dekker, 1985b.

G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, Philadelphia, 1990.

G. Wahba. Multivariate function and operator estimation, based on smoothing splines and reproducing kernels. In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting, SFI Studies in the Sciences of Complexity, Proc. Vol XII*, pages 95–112. Addison-Wesley, 1992.

G. Wahba, D. Johnson, F. Gao, and J. Gong. Adaptive tuning of numerical weather prediction models: randomized GCV in three and four dimensional data assimilation. *Mon. Wea. Rev.*, 123:3358–3369, 1995a.

G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein. Structured machine learning for 'soft' classification with smoothing spline ANOVA and stacked tuning, testing and evaluation. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 415–422. Morgan Kauffman, 1994.

G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein. Smoothing spline ANOVA for exponential families, with application to the Wisconsin Epidemiological Study of Diabetic Retinopathy. *Ann. Statist.*, 23:1865–1895, 1995b.

T. L. H. Watkin, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Reviews of Modern Physics*, 65:499–556, 1993.

A. S. Weigend and N. A. Gershenfeld (Eds.). *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley, 1994. Santa Fe Institute Studies in the Sciences of Complexity.

P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* PhD thesis, Harvard, 1974.

J. Werner. *Optimization - Theory and Applications.* Vieweg, 1984.

J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.

H. Widom. Asymptotic behaviour of eigenvalues of certain integral operators. *Archive for Rational Mechanics and Analysis*, 17:215–229, 1964.

R. A. Wilkinson, J. Geist, S. Janet, P. J. Grother, C. J. C. Burges, R. Creecy, B. Hammond, J. J. Hull, N. J. Larsen, T. P. Vogl, and C. L. Wilson. The first census optical character recognition system conference. Technical Report NISTIR 4912, National Institute of Standards and Technology (NIST), Gaithersburg, 1992.

C. K. I. Williams. Computation with infinite networks. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, Cambridge, MA, 1997. MIT Press.

C. K. I. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning and Inference in Graphical Models.* Kluwer, 1998. To appear. Also: Technical Report NCRG/97/012, Aston University.

R. C. Williamson, A. J. Smola, and B. Schölkopf. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. Technical Report NC-TR-98-019, Royal Holloway College, University of London, UK, 1998a.

R. C. Williamson, A. J. Smola, and B. Schölkopf. A Maximum Margin Miscellany. Typescript, 1998b.

W. H. Wolberg and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences,U.S.A.*, 87:9193–9196, 1990.

A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano. Determining lyapunov exponents from a time series. *Physica D*, 16:285–317, 1985.

D. Xiang. *Model Fitting and Testing for Non-Gaussian Data with a Large Data Set.* PhD thesis, Technical Report 957, University of Wisconsin-Madison, Madison

WI, 1996.

D. Xiang and G. Wahba. A generalized approximate cross validation for smoothing splines with non-Gaussian data. *Statistica Sinica*, 6:675–692, 1996.

D. Xiang and G. Wahba. Approximate smoothing spline methods for large data sets in the binary case. Technical Report 982, Department of Statistics, University of Wisconsin, Madison WI, 1997. To appear in the Proceedings of the 1997 ASA Joint Statistical Meetings, Biometrics Section, pp 94-98 (1998).

P. V. Yee. *Regularized Radial Basis Function Netowrks: Theory and Applications to Probability Estimation, Classification, and Time Series Prediction.* PhD thesis, Dept. of ECE, McMaster University, Hamilton, Canada, 1998.

E. C. Zachmanoglou and Dale W. Thoe. *Introduction to Partial Differential Equations with Applications.* Dover, Mineola, N.Y., 1986.

X. Zhang and J. Hutchinson. Simple architectures on fast machines: practical issues in nonlinear time series prediction. In A. S. Weigend and N. A. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past.* Santa Fe Institute, Addison-Wesley, 1994.

G. Zoutendijk. *Methods of Feasible Directions: a Study in Linear and Non-line ar Programming.* Elsevier, 1970.