On a Kernel-based Method for Pattern Recognition, Regression, Approximation, and Operator Inversion

Alex J. Smola* and Bernhard Schölkopf†

June 24, 1997

Abstract

We present a Kernel–based framework for Pattern Recognition, Regression Estimation, Function Approximation and multiple Operator Inversion. Adopting a regularization–theoretic framework, the above are formulated as constrained optimization problems. Previous approaches such as ridge–regression, Support Vector methods and Regularization Networks are included as special cases. We will show connections between the cost function and some properties up to now believed to apply to Support Vector Machines only. For appropriately chosen cost functions, the optimal solution of all the problems described above can be found by solving a simple quadratic programming problem.

Acknowledgements:

We would like to thank Volker Blanz, Léon Bottou, Chris Burges, Patrick Haffner, Jörg Lemm, Klaus-Robert Müller, Noboru Murata, Sara Solla, Vladimir Vapnik and the referees for helpful comments and discussions. This work was supported by the Studienstiftung des deutschen Volkes. The authors are indebted to AT&T and Bell Laboratories for the possibility to profit from an excellent research environment during several research stays.

---

*GMD FIRST, Rudower Chaussee 5, 12489 Berlin, Germany, Tel. +49-030-63921871, Fax +49-030-63921805, E–mail smola@first.gmd.de

†Max Planck Institut für biologische Kybernetik, Spemannstr. 38, 72076 Tübingen, Germany, Tel. +49-7071-601609, Fax +49-7071-601616, E–mail bs@mpik-tueb.mpg.de

1 Introduction

Estimating dependences from empirical data can be viewed as *risk minimization* [43]: we are trying to estimate a function such that the risk, defined in terms of some a priori chosen cost function measuring the error of our estimate for (unseen) input–output *test* examples, becomes minimal. The fact that this has to be done based on a *limited* amount of *training* examples comprises the central problem of statistical learning theory. A number of approaches for estimating functions have been proposed in the past, ranging from simple methods like linear regression over ridge regression (see e.g. [3]) to advanced methods like Generalized Additive Models [16], Neural Networks and Support Vectors [4]. In combination with different types of cost functions, as for instance quadratic ones, robust ones in Huber's sense [18], or ϵ–insensitive ones [41], these yield a wide variety of different training procedures which at first sight seem incompatible with each other. The purpose of this paper, which was inspired by the treatments of [7] and [40], is to present a framework which contains the above models as special cases and provides a constructive algorithm for finding global solutions to these problems. The latter is of considerable practical relevance insofar as many common models, in particular neural networks, suffer from the possibility of getting trapped in local optima during training.

Our treatment starts by giving a definition of the risk functional general enough to deal with the case of solving multiple operator equations (Sec. 2). These provide a versatile tool for dealing with measurements obtained in different ways, as in the case of sensor fusion, or for solving boundary constrained problems. Moreover, we shall show that they are useful for describing symmetries inherent to the data, be it by the incorporation of virtual examples or by enforcing tangent constraints. To minimize risk, we adopt a regularization approach which consists in minimizing the sum of training error and a complexity term defined in terms of a regularization operator [38]. The minimization is carried out over classes of functions written as kernel expansions in terms of the training data (Sec. 3). Moreover, we describe several common choices of the regularization operator. Following that, section 4 contains a derivation of an algorithm for practically obtaining a solution of the problem of minimizing the regularized risk. For appropriate choices of cost functions, the algorithm reduces to quadratic programming. Section 5 generalizes a theorem by Morozov from quadratic cost functions to the case of convex ones, which will give the general form of the solution to the problems stated above. Finally, section 6 contains practical applications of multiple operators to the case of problems with prior knowledge. Appendix A and B contain proofs of the formulae of section 4 and 5, and Appendix C describes an algorithm for incorporating prior knowledge in the form of transformation invariances in Pattern Recognition problems.

2 Risk Minimization

In Regression estimation we try to estimate a functional dependency f between a set of sampling points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ taken from a space V, and target values $Y = \{y_1, \ldots, y_\ell\}$. Let us now consider a situation where we cannot observe X, but some other corresponding points $X_s = \{\mathbf{x}_{s1}, \ldots, \mathbf{x}_{s\ell_s}\}$, nor can we observe Y, but $Y_s = \{y_{s1}, \ldots, y_{s\ell_s}\}$. Let us call the pairs $(\mathbf{x}_{ss'}, y_{ss'})$ *measurements* of the dependency f. Suppose we know that the elements of X_s are generated from those of X by a (possibly nonlinear) transformation $\hat{T}$:

$$\mathbf{x}_{ss'} = \hat{T}\mathbf{x}_{s'}, \quad (s' = 1, \ldots, \ell). \tag{1}$$

The corresponding transformation $A_{\hat{T}}$ acting on f,

$$(A_{\hat{T}} f)(\mathbf{x}) := f(\hat{T}\mathbf{x}), \tag{2}$$

is then generally linear: for functions f, g and coefficients α, β we have

$$\begin{aligned} (A_{\hat{T}}(\alpha f + \beta g))(\mathbf{x}) &= (\alpha f + \beta g)(\hat{T}\mathbf{x}) \\ &= \alpha f(\hat{T}\mathbf{x}) + \beta g(\hat{T}\mathbf{x}) \\ &= \alpha (A_{\hat{T}} f)(\mathbf{x}) + \beta (A_{\hat{T}} g)(\mathbf{x}). \end{aligned} \tag{3}$$

Knowing $A_{\hat{T}}$, we can use the data to estimate the underlying functional dependency. For several reasons, this can be preferable to estimating the dependencies in the transformed data directly. For instance, there are cases where we specifically want to estimate the original function, as in the case of Magnetic Resonance Imaging [42]. Moreover, we may have multiple transformed data sets, but only estimate *one* underlying dependency. These data sets might differ in size; in addition, we might want to associate different costs with estimation errors for different types of measurements, e.g. if we believe them to differ in reliability. Finally, if we have knowledge of the transformations, we may as well utilize it to improve the estimation. Especially if the transformations are complicated, the original function might be easier to estimate. A striking example is the problem of backing up a truck with a trailer to a given position [14]. This problem is a complicated classification problem (steering wheel left or right) when expressed in cartesian coordinates; in polar coordinates, however, it becomes linearly separable.

Without restricting ourselves to the case of operators acting on the arguments of f only, but for general linear operators, we formalize the above as follows. We consider pairs of observations $(\mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}})$, with sampling points $\mathbf{x}_{\bar{\imath}}$ and corresponding target values $y_{\bar{\imath}}$. The first entry i of the multi–index $\bar{\imath} := (i, i')$ denotes the procedure by which we have obtained the target values; the second entry i' runs over the observations $1, \ldots, \ell_i$. In the following, it is understood that variables without bar correspond to the appropriate entries of the multi–indices. This will help us to avoid multiple summation symbols. We may group these pairs in q pairs of sets X_i and Y_i by defining

$$\begin{array}{rl} X_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{i\ell_i}\} & \text{with} \quad \mathbf{x}_{\bar{\imath}} \in V_i \\ Y_i = \{y_{i1}, \ldots, y_{i\ell_i}\} & \text{with} \quad y_{\bar{\imath}} \in \mathbb{R} \end{array} \tag{4}$$

with V_i being vector spaces.

Let us assume that these samples have been drawn independently from q corresponding probability distributions with densities $p_1(\mathbf{x}_1, y_1), \ldots, p_q(\mathbf{x}_q, y_q)$ for the sets X_i and Y_i respectively.

Let us further assume that there exists a Hilbert Space of real–valued functions on V, denoted by $\mathcal{H}(V)$, and a set of linear operators $\hat{A}_1, \ldots, \hat{A}_q$ on $\mathcal{H}(V)$ such that

$$\hat{A}_i : \mathcal{H}(V) \to \mathcal{H}(V_i) \tag{5}$$

for some Hilbert Space $\mathcal{H}(V_i)$ of real–valued functions on V_i. (In the case of Pattern Recognition, we consider functions with values in $\{\pm 1\}$ only.)

Our aim is to estimate a function $f \in \mathcal{H}(V)$ such that the *risk functional*

$$R[f] = \sum_{i=1}^{q} \int c_i \left((\hat{A}_i f)(\mathbf{x}_i), \mathbf{x}_i, y_i \right) p_i(\mathbf{x}_i, y_i) d\mathbf{x}_i dy_i \tag{6}$$

is minimized.[1] (In some cases, we restrict ourselves to subsets of $\mathcal{H}(V)$ in order to control the capacity of the admissible models.)

[1] A note on underlying functional dependences: for each V_i together with p_i one might define a function

$$\bar{y}_i(\mathbf{x}_i) := \int y_i P_i(y_i | \mathbf{x}_i) dy_i \tag{7}$$

and try to find a corresponding function f such that $\hat{A}_i f = \bar{y}_i$ holds. This intuition, however, is misleading, as $\bar{y}_i$ need not even lie in the range of $\hat{A}_i$, and $\hat{A}_i$ need not be invertible either. We will resort to find a pseudo-solution of the operator equation. For a detailed treatment see [26].

The functions c_i are cost functions determining the loss for deviations between the estimate generated by $\hat{A}_i f$ and the target value y_i at the position $\mathbf{x}_i$. We require these functions to be bounded from below and therefore, by adding a constant, we may as well require them to be nonnegative. The dependence of c_i on $\mathbf{x}_i$ can for instance accomodate the case of a measurement device whose precision depends on the location of the measurement.

Example 1 (Vapnik's Risk functional)
By specializing

$$q = 1, \quad \hat{A} = \mathbf{1} \tag{8}$$

we arrive at the definition of the risk functional of [40]:[2]

$$R[f] = \int c(f, \mathbf{x}, y) p(\mathbf{x}, y) d\mathbf{x} dy \tag{9}$$

Specializing to $c(f(\mathbf{x}), \mathbf{x}, y) = (f(\mathbf{x}) - y)^2$ leads to the definition of the least mean square error risk [11].

As the probability density functions p_i are unknown, we cannot evaluate (and minimize) $R[f]$ directly. Instead we only can try to approximate

$$f_{min} := \operatorname{argmin}_{\mathcal{H}(V)} R[f] \tag{10}$$

by some function $\hat{f}$, using the given datasets X_i and Y_i. In practice, this requires considering the *empirical risk functional*, which is obtained by replacing the integrals over the probability density functions p_i (cf. (6)) with summations over the empirical data:

$$R_{emp}[f] = \sum_{\bar{\imath}} \frac{1}{\ell_i} c_i \left((\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}), \mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}} \right) \tag{11}$$

Here the notation $\sum_{\bar{\imath}}$ is a shorthand for $\sum_{i=1}^{q} \sum_{i'=1}^{\ell_i}$ with $\bar{\imath} = (i, i')$. The problem that arises now is how to connect the values obtained from $R_{emp}[f]$ with $R[f]$: we can only compute the former, but we want to minimize the latter. A naive approach is to minimize R_{emp}, hoping to obtain a solution $\hat{f}$ that is close to minimizing R, too. The Ordinary Least Mean Squares method is an example for these approaches, exhibiting overfitting in the case of a high model capacity, and thus poor generalization [11]. Therefore it is not advisable to minimize the empirical risk without any means of capacity control or regularization [40].

3 Regularization Operators and Additive Models

We will assume a regularization term in the spirit of [38] and [23], namely a positive semidefinite operator

$$\hat{P} : \mathcal{H}(V) \longrightarrow \mathcal{D} \tag{12}$$

mapping into a dot product space $\mathcal{D}$ (whose closure $\bar{\mathcal{D}}$ is a Hilbert Space), defining a regularized risk functional

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2} \|\hat{P} f\|_{\mathcal{D}}^2 \tag{13}$$

with a regularization parameter $\lambda \geq 0$. This additional term effectively reduces our model space and thereby controls the complexity of the solution. Note that the topic

[2] Note that (9) already includes multiple operator equations for the special case where $V_i = V$ and $p_i = p$ for all i, even though this is not explicitly mentioned in [40]: c is a functional of f and therefore it also may be a sum of functionals $\hat{A}_i f$ for several $\hat{A}_i$.

of this paper is not finding the best regularization parameter, which would require model selection criteria as for instance VC-theory [43], Bayesian methods [22], the Minimum Description Length principle [30], AIC [2], NIC [25] — a discussion of these methods, however, would go beyond the scope of this work. Instead, we focus on how and under which conditions, *given* a value of λ, the function minimizing R_{reg} can be found efficiently.

We do not require positive definiteness of $\hat{P}$, as we may not want to attenuate contributions of functions stemming from a given class of models M (e.g. linear and constant ones): in this case, we will construct $\hat{P}$ such that $M \subseteq \text{Ker } \hat{P}$. Making more specific assumptions about the type of functions used for minimizing (13), we assume f to be of the Additive Model type [16], with a function expansion based on a symmetric kernel $k(\mathbf{x}, \mathbf{x}')$ ($\mathbf{x}, \mathbf{x}' \in V$) with the property that for all $\mathbf{x} \in V$, the function on V obtained by fixing one argument of k to $\mathbf{x}$ is an element of $\mathcal{H}(V)$. To formulate the expansion, we use the tensor product notation for operators on $\mathcal{H}(V) \otimes \mathcal{H}(V)$,

$$((\hat{A} \otimes \hat{B})k)(\mathbf{x}, \mathbf{x}') \tag{14}$$

Here $\hat{A}$ acts on k as a function of $\mathbf{x}$ only (with $\mathbf{x}'$ fixed), and $\hat{B}$ vice versa. The class of models that we will investigate as admissible solutions for minimizing (13) are expansions of the form

$$f(\mathbf{x}) = \sum_{\bar{\imath}} \alpha_{\bar{\imath}} ((\hat{A}_i \otimes \mathbf{1})k)(\mathbf{x}_{\bar{\imath}}, \mathbf{x}) + b, \quad \text{with} \quad \alpha_{\bar{\imath}} \in \mathbb{R}. \tag{15}$$

This may seem to be a rather arbitrary assumption; however, kernel expansions of the type $\sum_{\bar{\imath}} \alpha_{\bar{\imath}} k(\mathbf{x}_{\bar{\imath}}, \mathbf{x})$ are quite common in regression and pattern recognition models [16], and in the case of Support Vectors even follow naturally from optimality conditions with respect to a chosen regularization [4], [42]. Moreover an expansion with as many basis functions as data points is rich enough to interpolate all measurements exactly, except for some pathological cases, e.g. if the functions $k_{\bar{\imath}}(\mathbf{x}) := k(\mathbf{x}_{\bar{\imath}}, \mathbf{x})$ are linearly dependent, or if there are conflicting measurements at one point (different target values for the same $\mathbf{x}$). Finally, using additive models is a useful approach insofar as the computations of the coefficients may be carried out more easily.

To obtain an expression for $\|\hat{P}f\|^2_{\mathcal{D}}$ in terms of the coefficients $\alpha_{\bar{\imath}}$, we first note

$$(\hat{P}f)(\mathbf{x}) = \sum_{\bar{\imath}} \alpha_{\bar{\imath}} ((\hat{A}_i \otimes \hat{P})k)(\mathbf{x}_{\bar{\imath}}, \mathbf{x}). \tag{16}$$

For simplicity we have assumed the constant function to lie in the kernel of $\hat{P}$, i.e. $\hat{P}b = 0$. Exploiting the linearity of the dot product in $\mathcal{D}$, we can express $\|\hat{P}f\|^2_{\mathcal{D}}$ as

$$(\hat{P}f \cdot \hat{P}f) = \sum_{\bar{\imath}, \bar{\jmath}} \alpha_{\bar{\imath}} \alpha_{\bar{\jmath}} (((\hat{A}_i \otimes \hat{P})k)(\mathbf{x}_{\bar{\imath}}, \mathbf{x}) \cdot ((\hat{A}_j \otimes \hat{P})k)(\mathbf{x}_{\bar{\jmath}}, \mathbf{x})). \tag{17}$$

For a suitable choice of k and $\hat{P}$, the coefficients

$$D_{\bar{\imath}\bar{\jmath}} := (((\hat{A}_i \otimes \hat{P})k)(\mathbf{x}_{\bar{\imath}}, .) \cdot ((\hat{A}_j \otimes \hat{P})k)(\mathbf{x}_{\bar{\jmath}}, .)) \tag{18}$$

can be evaluated in closed form, allowing an efficient implementation (here, the dot in $k(\mathbf{x}_{\bar{\imath}}, .)$ means that k is considered as a function of its second argument, with $\mathbf{x}_{\bar{\imath}}$ fixed). Positivity of (17) implies positivity of the regularization matrix D (arranging $\bar{\imath}$ and $\bar{\jmath}$ in dictionary order). Conversely, any positive semidefinite matrix will act as a regularization matrix. As we minimize the regularized risk (13), the functions corresponding to the largest eigenvalue of $D_{\bar{\imath}\bar{\jmath}}$ will be attenuated most; functions with expansion coefficient vectors lying in the kernel of D, however, will not be dampened at all.

Example 2 (Sobolev Regularization)
Smoothness properties of functions f can be enforced effectively by minimizing the Sobolev norm of a given order. Our exposition in this point follows [15]:

The Sobolev Space $H^{s,p}(V)$ ($s \in \mathbb{N}$, $1 \leq p \leq \infty$) is defined as the space of those L_p functions on V whose derivatives up to the order s are L_p functions. It is a Banach space with the norm

$$\|f\|_{H^{s,p}(V)} = \sum_{|\gamma| \leq s} \|\hat{D}^{\gamma} f\|_{L_p} \tag{19}$$

where γ is a multi-index and $\hat{D}^{\gamma}$ is the derivative of order γ. A special case of the Sobolev embedding theorem [37] yields:

$$H^{s,p}(V) \subset C^k \quad \text{for} \quad k \in \mathbb{N} \quad \text{and} \quad s > k + \frac{d}{2} \tag{20}$$

Here d denotes the dimensionality of V and C^k is the space of functions with continuous derivatives up to order k. Moreover there exists a constant c such that

$$max_{|\gamma| \leq k} \; sup_{\mathbf{x} \in V} |\hat{D}^{\gamma} f(\mathbf{x})| \;\; \leq \;\; c\|f\|_{H^{s,p}(V)} \tag{21}$$

i.e. convergence in the Sobolev norm enforces uniform convergence in the derivatives up to order k.

For our purposes, we use $p = 2$, for which $H^{s,p}(V)$ becomes a Hilbert Space. In this case, the coefficients of D are

$$D_{\bar{\imath}\bar{\jmath}} = \sum_{|\gamma| \leq s} (((\hat{A}_i \otimes \hat{D}^{\gamma})k)(\mathbf{x}_{\bar{\imath}}, \mathbf{x}) \cdot ((\hat{A}_j \otimes \hat{D}^{\gamma})k)(\mathbf{x}_{\bar{\jmath}}, \mathbf{x})). \tag{22}$$

Example 3 (Support Vector Regularization)
Let us consider functions which can be written as linear functions in some Hilbert space H,

$$f(\mathbf{x}) = (\Psi \cdot \Phi(\mathbf{x})) + b \tag{23}$$

with $\Phi : V \to H$ and $\Psi \in H$. The weight vector Ψ is expressed as a linear combination of the images of $\mathbf{x}_{\bar{\imath}}$,

$$\Psi = \sum_{\bar{\imath}} \alpha_{\bar{\imath}} \Phi(\mathbf{x}_{\bar{\imath}}). \tag{24}$$

The regularization operator $\hat{P}$ is chosen such that $\hat{P}f = \Psi$ for all $\alpha_{\bar{\imath}}$ (in view of the expansion (15), this defines a linear operator). Hence using the term $\|\hat{P}f\|^2_{\mathcal{D}} = \|\Psi\|^2_H$ corresponds to looking for the flattest linear function (23) on H. Moreover, Φ is chosen such that we can express the terms $(\Phi(\mathbf{x}_{\bar{\imath}}) \cdot \Phi(\mathbf{x}))$ in closed form as some symmetric function $k(\mathbf{x}_{\bar{\imath}}, \mathbf{x})$, thus the solution (23) reads

$$f(\mathbf{x}) = \sum_{\bar{\imath}} \alpha_{\bar{\imath}} k(\mathbf{x}_{\bar{\imath}}, \mathbf{x}) + b, \tag{25}$$

and the regularization term becomes

$$\|\Psi\|^2_H = \sum_{\bar{\imath}\bar{\jmath}} \alpha_{\bar{\imath}} \alpha_{\bar{\jmath}} k(\mathbf{x}_{\bar{\imath}}, \mathbf{x}_{\bar{\jmath}}). \tag{26}$$

This leads to the optimization problem of [4]. The mapping Φ need not be known explicitly: for any continuous symmetric kernel k satisfying Mercer's condition [9]

$$\int f(x)k(x,y)f(y) \; dx \; dy > 0 \quad if \quad f \in L_2 \backslash \{0\}, \tag{27}$$

one can expand k into a uniformly convergent series $k(x,y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\phi_i(y)$ with positive coefficients λ_i for $i \in \mathbb{N}$. Using this, it is easy to see that $\Phi(x) := \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(x)\mathbf{e}_i$ ($\{\mathbf{e}_i\}$ denoting an orthonormal basis of ℓ_2) is a map satisfying $(\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}')$. In particular, this implies that the matrix $D_{\bar{\imath}\bar{\jmath}} = k(\mathbf{x}_{\bar{\imath}}, \mathbf{x}_{\bar{\jmath}})$ is positive.

Different choices of kernel functions allow the construction of polynomial classifiers [4] and radial basis function classifiers [33]. Although formulated originally for the case where f is a function of one variable, Mercer's theorem also holds if f is defined on a space of arbitrary dimensionality, provided that it is compact [12].[3]

In the next example, as well as in the remainder of the paper, we shall use vector notation; e.g., $\vec{\alpha}$ denotes the vector with entries $\alpha_{\bar{\imath}}$, with $\bar{\imath}$ arranged in dictionary order.

Example 4 (Ridge Regression)
If we define $\hat{P}$ such that all functions used in the expansion of f are attenuated equally and decouple, D becomes the identity matrix, $D_{\bar{\imath}\bar{\jmath}} = \delta_{\bar{\imath}\bar{\jmath}}$. This leads to

$$\|\hat{P}f\|_{\mathcal{D}}^2 = \sum_{\bar{\imath}\bar{\jmath}} \alpha_{\bar{\imath}}\alpha_{\bar{\jmath}}D_{\bar{\imath}\bar{\jmath}} = \|\vec{\alpha}\|^2 \tag{28}$$

and

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2}\|\vec{\alpha}\|^2 \tag{29}$$

which is exactly the definition of a ridge–regularized risk functional, known in the neural network community as weight *principle [3]. The concept of ridge regression appeared in [17] in the context of Linear Discriminant Analysis.*

[28] gives an overview over some more choices of regularization operators and corresponding kernel expansions.

4 Risk Minimization by Quadratic Programming

The goal of this section is to transform the problem of minimizing the regularized risk (13) into a quadratic programming problem which can be solved efficiently by existing techniques. In the following we only require the cost functions to be convex in the first argument and

$$c_i(y_i, \mathbf{x}_i, y_i) = 0 \quad \text{for all} \quad \mathbf{x}_i \in V_i. \tag{30}$$

More specifically, we require $c_i(., \mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}})$ to be zero exactly on the interval $[-\epsilon_{\bar{\imath}}^* + y_{\bar{\imath}}, \epsilon_{\bar{\imath}} + y_{\bar{\imath}}]$ with $0 \leq \epsilon_{\bar{\imath}}, \epsilon_{\bar{\imath}}^* \leq \infty$ and C^1 everywhere else. For brevity we will write

$$\begin{array}{lcll} c_{\bar{\imath}}(\eta_{\bar{\imath}}) & := & \frac{1}{\ell_i}c_i(y_{\bar{\imath}} + \epsilon_{\bar{\imath}} + \eta_{\bar{\imath}}, \mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}}) & \text{with} \quad \eta_{\bar{\imath}} \geq 0 \\ c_{\bar{\imath}}^*(\eta_{\bar{\imath}}^*) & := & \frac{1}{\ell_i}c_i(y_{\bar{\imath}} - \epsilon_{\bar{\imath}}^* - \eta_{\bar{\imath}}^*, \mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}}) & \text{with} \quad \eta_{\bar{\imath}}^* \geq 0 \end{array} \tag{31}$$

with $\mathbf{x}_{\bar{\imath}}$ and $y_{\bar{\imath}}$ fixed and

$$\begin{array}{lcl} \eta_{\bar{\imath}} & := & \max\left(\ (\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) - y_{\bar{\imath}} - \epsilon_{\bar{\imath}}, 0\right) \\ \eta_{\bar{\imath}}^* & := & \max\left(-(\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) + y_{\bar{\imath}} - \epsilon_{\bar{\imath}}^*, 0\right) \end{array} \tag{32}$$

The asterisk is used for distinguishing positive and negative slack variables and corresponding cost functions. The functions $c_{\bar{\imath}}$ and $c_{\bar{\imath}}^*$ describe the parts of the cost

[3] The expansion of Ψ in terms of the images of the data follows more naturally if viewed in the Support Vector context [41]; however the idea of selecting the flattest function in a high dimensional space is preserved in the present exposition.

functions c_i at the location $(\mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}})$ which differ from zero, split up into a separate treatment of $(\hat{A}_i f) - y_{\bar{\imath}} \geq \epsilon_{\bar{\imath}}$ and $(\hat{A}_i f) - y_{\bar{\imath}} \leq -\epsilon^*_{\bar{\imath}}$. This is done to avoid the (possible) discontinuity in the first derivative of c_i at the point where it starts differing from zero.

In pattern recognition problems, the intervals $[-\epsilon^*_{\bar{\imath}}, \epsilon_{\bar{\imath}}]$ are either $[-\infty, 0]$ or $[0, \infty]$. In this case, we can get rid of one of the two appearing slack variables, thereby getting a simpler form for the optimization problem. In the following, however, we shall deal with the more general case of regression estimation.

We may rewrite the minimization of R_{reg} as a constrained optimization problem, using $\eta_{\bar{\imath}}$ and $\eta^*_{\bar{\imath}}$, to render the subsequent calculus more amenable.

$$\begin{array}{ll} \text{minimize} & \frac{1}{\lambda} R_{reg} = \frac{1}{\lambda} \sum_{\bar{\imath}} (c_{\bar{\imath}}(\eta_{\bar{\imath}}) + c^*_{\bar{\imath}}(\eta^*_{\bar{\imath}})) + \frac{1}{2} \|\hat{P} f\|^2_{\mathcal{D}} \\ \text{subject to} & (\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) \leq y_{\bar{\imath}} + \epsilon_{\bar{\imath}} + \eta_{\bar{\imath}} \\ & (\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) \geq y_{\bar{\imath}} - \epsilon^*_{\bar{\imath}} - \eta^*_{\bar{\imath}} \\ & \eta_{\bar{\imath}}, \eta^*_{\bar{\imath}} \geq 0 \end{array} \tag{33}$$

The dual of this problem can be computed using standard Lagrange multiplier techniques. In the following, we shall make use of the results derived in Appendix A, and discuss some special cases obtained by choosing specific loss functions.

Example 5 (Quadratic Cost Function)
We use Eq. (71) (Appendix A, Example 12) in the special case $p = 2, \epsilon = 0$ to get the following unconstrained optimization problem:

$$\begin{array}{ll} \textit{maximize} & (\vec{\beta}^* - \vec{\beta})^\top \vec{y} - \frac{\lambda}{2}(\|\vec{\beta}\|^2 + \|\vec{\beta}^*\|^2) - \frac{1}{2}(\vec{\beta}^* - \vec{\beta})^\top K D^{-1} K (\vec{\beta}^* - \vec{\beta}) \\ \textit{subject to} & \sum_{\bar{\imath}} (\hat{A}_i 1)(\beta^*_{\bar{\imath}} - \beta_{\bar{\imath}}) = 0, \quad \beta_{\bar{\imath}}, \beta^*_{\bar{\imath}} \in \mathbb{R}^+_0 \end{array} \tag{34}$$

Transformation back to $\alpha_{\bar{\imath}}$ is done by

$$\vec{\alpha} = D^{-1} K (\vec{\beta} - \vec{\beta}^*) \tag{35}$$

Here, the symmetric matrix K is defined as

$$K_{\bar{\imath}\bar{\jmath}} := ((\hat{A}_i \otimes \hat{A}_j) k)(\mathbf{x}_{\bar{\imath}}, \mathbf{x}_{\bar{\jmath}}), \tag{36}$$

$(\hat{A}_i 1)$ is the operator $\hat{A}_i$ acting on the constant function with value 1. Of course there would have been a simpler solution to this problem (by combining $\beta_{\bar{\imath}}$ and $\beta^*_{\bar{\imath}}$ into one variable resulting in an unconstrained optimization problem) but in combination with other cost functions we may exploit the full flexibility of our approach.

Example 6 (ϵ-insensitive Cost Function)
Here we use Eq. (75) (Appendix A, Example 13) for $\sigma = 0$. This leads to

$$\begin{array}{ll} \textit{maximize} & (\vec{\beta}^* - \vec{\beta})^\top \vec{y} - (\vec{\beta}^* + \vec{\beta})^\top \vec{\epsilon} - \frac{1}{2}(\vec{\beta}^* - \vec{\beta})^\top K D^{-1} K (\vec{\beta}^* - \vec{\beta}) \\ \textit{subject to} & \sum_{\bar{\imath}} (\hat{A}_i 1)(\beta^*_{\bar{\imath}} - \beta_{\bar{\imath}}) = 0, \quad \beta_{\bar{\imath}}, \beta^*_{\bar{\imath}} \in [0, \frac{1}{\lambda}] \end{array} \tag{37}$$

with the same back substitution rules (35) as in Example 5. For the special case of Support Vector regularization, this leads to exactly the same equations as in Support Vector Pattern Recognition or Regression Estimation [41]. In that case, one can show that $D = K$, and therefore the terms $D^{-1}K$ cancel out, with only the Support Vector equations remaining. This follows directly from (25) and (26) with the definitions of D and K.

Note that the Laplacian cost function is included as a special case for $\epsilon = 0$.

Example 7 (Huber's robust Cost Function)
Setting

$$p = 2, \epsilon = 0 \tag{38}$$

in Example 13 leads to the following optimization problem

$$\begin{aligned} \textit{maximize} \quad & (\vec{\beta}^* - \vec{\beta})^\top \vec{y} - \frac{\lambda}{2} \sum_{\bar{\imath}} (\sigma_{\bar{\imath}} \beta_{\bar{\imath}}^2 + \sigma_{\bar{\imath}}^* {\beta^2}^*_{\bar{\imath}}) - \frac{1}{2} (\vec{\beta}^* - \vec{\beta})^\top K D^{-1} K (\vec{\beta}^* - \vec{\beta}) \\ \textit{subject to} \quad & \sum_{\bar{\imath}} (\hat{A}_i 1)(\beta_{\bar{\imath}}^* - \beta_{\bar{\imath}}) = 0, \quad \beta_{\bar{\imath}}, \beta_{\bar{\imath}}^* \in [0, \frac{1}{\lambda}] \end{aligned} \tag{39}$$

with the same backsubstitution rules (35) as in Example 5.

The cost functions described in the Examples 5, 6, 7, 12, and 13 may be linearly combined into more complicated ones. In practice, this results in using additional Lagrange multipliers, as each of the cost functions has to be dealt with using one multiplier. Still, by doing so computational complexity is not greatly increased as only the linear part of the optimization problem is increased, whereas the quadratic part remains unaltered (except for a diagonal term for cost functions of the Huber type). [24] reports excellent performance of the Support Vector regression algorithm for both ϵ–insensitive and Huber cost function matching the correct type of the noise in an application to time series prediction.

5 A Generalization of a Theorem by Morozov

We will follow and extend the proof of a theorem originally stated by [23] as described in [28] and [7]. As in section 4, we require the cost functions c_i to be convex and C^1 in the first argument with the extra requirement

$$c_i(y_i, \mathbf{x}_i, y_i) = 0 \quad \text{for all} \quad \mathbf{x}_i \in V_i \quad \text{and} \quad y_i \in \mathbb{R}. \tag{40}$$

We use the notation $\bar{\mathcal{D}}$ for the closure of $\mathcal{D}$, and $\hat{P}^*$ to refer to the adjoint[4] of $\hat{P}$,

$$\begin{aligned} \hat{P}: \quad & \mathcal{H}(V) \to \mathcal{D} \\ \hat{P}^*: \quad & \bar{\mathcal{D}} \to \hat{P}^* \bar{\mathcal{D}} \subseteq \mathcal{H}(V). \end{aligned} \tag{41}$$

Theorem 1 (Optimality Condition)
Under the assumptions stated above, a necessary and sufficient condition for

$$f = f_{opt} := \mathit{argmin}_{f \in \mathcal{H}(V)} R_{reg}[f] \tag{42}$$

is that the following equation holds true:

$$\hat{P}^* \hat{P} f = -\frac{1}{\lambda} \sum_{\bar{\imath}} \frac{1}{\ell_i} \partial_1 c_i \left((\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}), \mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}} \right) \hat{A}_i^* \delta_{\mathbf{x}_{\bar{\imath}}} \tag{43}$$

[4] The adjoint of an operator $\hat{O} : \mathcal{H}_O \to \mathcal{D}_O$ mapping from a Hilbert Space $\mathcal{H}_O$ to a dot product space $\mathcal{D}_O$ is the operator $\hat{O}^*$ such that for all $f \in \mathcal{H}_O$ and $g \in \bar{\mathcal{D}}_O$,

$$\left(g \cdot \hat{O} f \right)_{\mathcal{H}_O} = \left(\hat{O}^* g \cdot f \right)_{\bar{\mathcal{D}}_O}.$$

Here, ∂_1 denotes the partial derivative of c_i by its first argument, and $\delta_{\mathbf{x}_{\bar{\imath}}}$ is the Dirac distribution, centered on $\mathbf{x}_{\bar{\imath}}$. For a proof of the theorem see appendix B.

In order to illustrate the theorem, let us first consider the special case of $q = 1$ and $\hat{A} = \mathbf{1}$, i.e. the well known setting of regression and pattern recognition. The Green's function $G(\mathbf{x}, \mathbf{x}_j)$ corresponding to the operator $\hat{P}^*\hat{P}$ satisfies

$$(\hat{P}^*\hat{P}G)(\mathbf{x}, \mathbf{x}_j) = \delta_{\mathbf{x}_j}(\mathbf{x}), \tag{44}$$

as previously described in [28]. In this case we derive from (43) the following system of equations which has to be solved in a self consistent manner.

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{\ell} \gamma_i G(\mathbf{x}, \mathbf{x}_i) + b & (45) \\ \text{with } \gamma_i &= -\frac{1}{\gamma}\partial_1 c\left(f(\mathbf{x}_i), \mathbf{x}_i, y_i\right) & (46) \end{aligned}$$

Here the expansion of f in terms of kernel functions follows naturally with γ_i corresponding to Lagrange multipliers. It can be shown that G is symmetric in its arguments, and translation invariant for suitable regularization operators $\hat{P}$. Eq. (46) determines the size of γ_i according to how much f deviates from the original measurements y_i.

For the general case, (44) becomes a little bit more complicated, namely we will have q functions $G_i(\mathbf{x}, \mathbf{x}_{\bar{\imath}})$ such that

$$(\hat{P}^*\hat{P}G_i)(\mathbf{x}, \mathbf{x}_{\bar{\imath}}) = (\hat{A}_i^*\delta_{\mathbf{x}_{\bar{\imath}}})(\mathbf{x}) \tag{47}$$

holds. In [28] the Green's formalism is used for finding suitable kernel expansions corresponding to the chosen regularization operators for the case of regression and pattern recognition. This also may be applied to the case of estimating functional dependencies from indirect measurements. Moreover, (43) also may be useful for approximately solving some classes of partial differential equations by rewriting them as optimization problems.

6 Applications of Multiple Operator Equations

In the following, we discuss some examples of incorporating domain knowledge by using multiple operator equations as contained in (6).

Example 8 (Additional Constraints on the Estimated Function) *Suppose we have additional knowledge on the function values at some points, for instance saying that $-\epsilon \leq f(0) \leq \epsilon^*$ for some $\epsilon, \epsilon^* > 0$. This can be incorporated by adding the points as an extra set $X_s = \{\mathbf{x}_{s1}, \ldots, \mathbf{x}_{s\ell_s}\} \subset X$ with corresponding target values $Y_s = \{y_{s1}, \ldots, y_{s\ell_s}\} \subset Y$, an operator $\hat{A}_s = \mathbf{1}$, and a cost function (defined on X_s)*

$$c_s(f(\mathbf{x}_{\bar{s}}), \mathbf{x}_{\bar{s}}, y_{\bar{s}}) = \begin{cases} 0 & \text{if } -\epsilon_{\bar{s}} \leq f(\mathbf{x}_{\bar{s}}) - y_{\bar{s}} \leq \epsilon^*_{\bar{s}} \\ \infty & \text{otherwise} \end{cases} \tag{48}$$

defined in terms of $\epsilon_{s1}, \ldots, \epsilon_{s\ell_s}$ and $\epsilon^*_{s1}, \ldots, \epsilon^*_{s\ell_s}$.

These additional hard constraints result in optimization problems similar to those obtained in the ϵ–insensitive approach of Support Vector regression [42]. See Example 14 for details.

Monotonicity and convexity of a function f, along with other constraints on derivatives of f, can be enforced similarly. In that case, we use

$$\hat{A}_s = \left(\frac{\partial}{\partial \mathbf{x}}\right)^p \tag{49}$$

instead of the $\hat{A}_s = \mathbf{1}$ used above. This requires differentiability of the function expansion of f. If we want to use general expansions (15), we have to resort to finite difference operators.

Example 9 (Virtual Examples)
Suppose we have additional knowledge telling us that the function to be estimated should be invariant with respect to certain transformations $\hat{T}_i$ of the input. For instance, in optical character recognition, these transformations might be translations, small rotations, or changes in line thickness [34]. We then define corresponding linear operators $\hat{A}_i$ acting on $\mathcal{H}(V)$ as in Eq. (2).

As the empirical risk functional (11) then contains a sum over original and transformed ("virtual") patterns, this corresponds to training on an artificially enlarged data set. Unlike previous approaches such as the one of [32], we may assign different weight to the enforcement of the different invariances by choosing different cost functions c_i. If the $\hat{T}_i$ comprise translations of different amounts, we may for instance use smaller cost functions for bigger translations. Thus, deviations of the estimated function on these examples will be penalized less severely, which is reflected by smaller Lagrange multipliers (cf. Eq. (62)). Still there are more general types of symmetries, especially non–deterministic ones which also could be taken care of by modified cost functions. For an extended discussion of this topic see [21]. In Appendix C, we give a more detailed description of how to implement a virtual examples algorithm.

Much work on symmetries and invariances (e.g. [44]) is mainly concerned with global symmetries (independent of the training data) that have a linear representation in the domain of the input patterns. This concept however can be rather restrictive. Even for the case of Handwritten Digit Recognition, the above requirements can be fulfilled for translation symmetries only. Rotations for instance cannot be faithfully represented in this context. Moreover would a full rotation invariance not be desirable (thereby transforming a **6** into a **9**) — only local invariances should be admitted. Some symmetries only exist for a class of patterns (mirror symmetries are a reasonable concept for the digits **8** and **0** only) and some only can be defined on the patterns themselves, e.g. stroke changes, and do not make any sense on a random collection of pixels at all. This requires a model capable of dealing with nonlinear, local, pattern dependent and possibly only approximate symmetries, all of which can be achieved by the concept of virtual examples.

Example 10 (Hints)
We can also utilize prior knowledge where target values or ranges for the function are not explicitly available. For instance, we might know that f takes the same value at two different points $\mathbf{x}_1$ and $\mathbf{x}_2$ [1]. E.g., we could use unlabelled data together with known invariance transformations to generate such pairs of points. To incorporate this type of invariance of the target function, we use a linear operator acting on the direct sum of two copies of input space, computing the difference between $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$,

$$(\hat{A}_s f)(\mathbf{x}_1 \oplus \mathbf{x}_2) := f(\mathbf{x}_1) - f(\mathbf{x}_2). \tag{50}$$

The technique of Example 8 then allows us to constrain $(\hat{A}_s f)$ to be small, on a set of sampling points generated as direct sums of the given pairs of points.

As before (Eq. 49), we can modify the above methods using derivatives of f. This will lead to tangent regularizers as the ones proposed by [35], as we shall presently show.

Example 11 (Tangent Regularizers)
Let us assume that G is a Lie group of invariance transformations. Similar to (2), we can define an action of G on a Hilbert space $\mathcal{H}(V)$ of functions on V, by

$$(g \cdot f)(\mathbf{x}) := f(g\mathbf{x}) \text{ for } g \in G, f \in \mathcal{H}(V). \tag{51}$$

The generators in this representation, call them $\hat{S}_i$, $i = 1, \ldots, r$, generate the group in a neighbourhood of the identity via the exponential map $\exp(\sum_i \alpha_i \hat{S}_i)$. As first–order (tangential) invariance is a local property at the identity, we may enforce it by requiring

$$(\hat{S}_i f)(\mathbf{x}) = 0 \quad \text{for } i = 1, \ldots, r. \tag{52}$$

To motivate this, note that

$$\begin{aligned} \frac{\partial}{\partial \alpha_i}\Big|_{\boldsymbol{\alpha}=0} f\Big(\exp\Big(\sum_j \alpha_j \hat{S}_j\Big)\mathbf{x}\Big) &= \Big(\Big(\frac{\partial}{\partial \alpha_i}\Big|_{\boldsymbol{\alpha}=0} \exp\Big(\sum_j \alpha_j \hat{S}_j\Big)\Big) f\Big)(\mathbf{x}) \\ &= (\hat{S}_i f)(\mathbf{x}), \end{aligned} \tag{53}$$

using Eq. (51), the chain rule, and the identity $\exp(0) = \mathbf{1}$.

Examples of operators $\hat{S}_i$ that can be used are derivative operators, which are the generators of translations. Operator equations of the type (52) allow us to use virtual examples which incorporate knowledge about derivatives of f. In the sense of [35], this corresponds to having a regularizer enforcing invariance. Interestingly, our analysis suggests that this case is not as different from a direct virtual examples approach (Example 9) as it might appear superficially.

As in Example 8, prior knowledge could also be given in terms of allowed ranges or cost functions [20] for approximate symmetries, rather than enforced equalities as (52). Moreover, we can apply the approach of Example 11 as well to higher–order derivatives, generalizing what we said above about additional constraints on the estimated function (Example 8).

We conclude this section with an example of a possible application where the latter could be useful. In 3–D surface mesh construction (e.g. [19]), one tries to represent a surface by a mesh of few points, subject to the following constraints. First, the surface points should be represented accurately — this can be viewed as a standard regression problem. Second, the normal vectors should be represented correctly, to make sure that the surface will look realistically when rendered. Third, if there are specular reflections, say, geometrical optics comes into play, and thus surface curvature (i.e. higher–order derivatives) should be represented accurately.

7 Discussion

We have shown that we can employ fairly general types of regularization and cost functions, and still arrive at a Support–Vector type quadratic optimization problem. An important feature of Support Vector machines, however, sparsity of the decompositions of f, is due to a special type of cost functions used. The decisive part is the nonvanishing interval $[y_{\bar{\imath}} - \epsilon^*_{\bar{\imath}}, y_{\bar{\imath}} + \epsilon_{\bar{\imath}}]$ inside of which the cost for approximation, regression, or pattern recognition is zero. Therefore there exists a range of values $(\hat{A}_i f)(\mathbf{x}_{\bar{\imath}})$ in which (32) holds with $\eta_{\bar{\imath}}, \eta^*_{\bar{\imath}} = 0$ for some $\bar{\imath}$. By virtue of the Karush-Kuhn-Tucker conditions, stating that the product of constraints and Lagrange multipliers have to vanish at the point of optimality, (33) implies

$$\beta_{\bar{\imath}}(y_{\bar{\imath}} + \epsilon_{\bar{\imath}} + \eta_{\bar{\imath}} - (\hat{A}_i f)(\mathbf{x}_{\bar{\imath}})) = 0 \quad (54)$$
$$\beta^*_{\bar{\imath}}((\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) - y_{\bar{\imath}} + \epsilon^*_{\bar{\imath}} + \eta_{\bar{\imath}}) = 0. \quad (55)$$

Therefore, the $\beta_{\bar{\imath}}$ and $\beta^*_{\bar{\imath}}$ have to vanish for the constraints of (33) that become strict inequalities. This causes sparsity in the solution of $\beta_{\bar{\imath}}$ and $\beta^*_{\bar{\imath}}$.

As shown in the Examples 3 and 6, the special choice of a Support Vector regularization combined with the ϵ-insensitive cost function brings us to the case of Support Vector Pattern Recognition and Regression Estimation. The advantage of this setting is that in the low noise case, it generates sparse decompositions of $f(\mathbf{x})$ in terms of the training data, i.e. in terms of Support Vectors. This advantage however vanishes for noisy data as the number of Support Vectors increases with the noise (see [36] for details).

Unfortunately, independent of the noise level, the choice of a different regularization prevents such an efficient calculation scheme due to equation (35), as $D^{-1}K$ generally may not be assumed to be diagonal. Consequently, the expansion of f is only sparse in terms of β but not in α. Yet this is sufficient for some *encoding* purposes as f is defined uniquely by the matrix $D^{-1}K$ and the set of $\beta_{\bar{\imath}}$. Hence storing $\alpha_{\bar{\imath}}$ is not required.

The computational cost of *evaluating* $f(\mathbf{x}_0)$ also can be reduced. For the case of a kernel $k(\mathbf{x}, \mathbf{x}')$ satisfying Mercer's condition (27), the reduced set method [6] can be applied to the initial solution. In that case, the final computational cost is comparable to the one of Support Vector machines, with the advantage of regularization in input space (which is the space we really are interested in) instead of high dimensional space.

The computational cost is approximately cubic in the number of nonzero Lagrange multipliers $\beta_{\bar{\imath}}$, as we have to solve a quadratic programming problem whose quadratic part is as large as the number of basis functions of the functional expansion of f. Optimization methods like the Bunch-Kaufman decomposition ([5], [10]) have the property of incuring computational cost only in the number of nonzero coefficients, whereas for cases with a large percentage of nonvanishing Lagrange multipliers, interior point methods (e.g. [39]) might be computationally more efficient.

We deliberately omitted the case of having fewer basis functions than constraints, as (depending on the cost function) optimization problems of this kind may become infeasible, at least for the case of hard constraints. However, it is not very difficult to see how a generalization to an arbitrary number of basis functions could be achieved: denote n the number of functions of which f is a linear combination, $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, and m the number of constraints or cost functions on f. Then D will be an $n \cdot n$ matrix and K an $n \cdot m$ matrix, i.e. we have n variables α_i and m Lagrange multipliers β_i. The calculations will lead to a similar class of quadratic optimization problems as described in (33) and (56), with the difference that the quadratic part of the problem will be at most of rank n, whereas the quadratic matrix will be of size $m \cdot m$. A possible way of dealing with this degeneracy is to use a singular value decomposition [29] and solve the optimization equations in the reduced space.

To summarize, we have embedded the Support Vector method into a wider regularization–theoretic framework, which allow us to view a variety of learning approaches, including but not limited to Least Mean Squares, Ridge Regression, and Support Vector machines as special cases of risk minimization using suitable loss fuctions. We have shown that general Arsenin–Tikhonov regularizers may be used while still preserving important advantages of Support Vector machines. Specifically, for particular choices of loss functions, the solution to the above problems (which can often be obtained only through nonlinear optimization, e.g. in regression estimation by neural networks) was reduced to a simple quadratic programming problem. Unlike many nonlinear optimization problems, the latter can be solved efficiently without the danger of getting trapped in local minima. Finally, we have shown that the formalism is powerful enough to deal with indirect measurements stemming from different sources.

A Optimization Problems for Risk Minimization

From equations (17) and (33) we arrive at the following statement of the optimization problem: Solve

$$\begin{array}{ll} \text{minimize} & \frac{1}{\lambda}\sum_{\bar{\imath}}(c_{\bar{\imath}}(\eta_{\bar{\imath}}) + c^*_{\bar{\imath}}(\eta^*_{\bar{\imath}})) + \frac{1}{2}\vec{\alpha}^\top D\vec{\alpha} \\ \text{subject to} & (\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) \le y_{\bar{\imath}} + \epsilon_{\bar{\imath}} + \eta_{\bar{\imath}} \\ & (\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) \ge y_{\bar{\imath}} - \epsilon^*_{\bar{\imath}} - \eta^*_{\bar{\imath}} \\ & \eta_{\bar{\imath}}, \eta^*_{\bar{\imath}} \ge 0 \text{ for all } \bar{\imath} \end{array} \tag{56}$$

To this end, we introduce a Lagrangian

$$\begin{array}{lll} L & = & \frac{1}{\lambda}\sum_{\bar{\imath}}(c_{\bar{\imath}}(\eta_{\bar{\imath}}) + c^*_{\bar{\imath}}(\eta^*_{\bar{\imath}})) + \frac{1}{2}\sum_{\bar{\imath}\bar{\jmath}}\alpha_{\bar{\imath}}\alpha_{\bar{\jmath}}D_{\bar{\imath}\bar{\jmath}} - \sum_{\bar{\imath}}(\eta_{\bar{\imath}}\xi_{\bar{\imath}} + \eta^*_{\bar{\imath}}\xi^*_{\bar{\imath}}) - \\ & & \sum_{\bar{\imath}}\beta_{\bar{\imath}}(y_{\bar{\imath}} + \epsilon_{\bar{\imath}} + \eta_{\bar{\imath}} - (\hat{A}_i f)(\mathbf{x}_{\bar{\imath}})) - \sum_{\bar{\imath}}\beta^*_{\bar{\imath}}((\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) - y_{\bar{\imath}} + \epsilon^*_{\bar{\imath}} + \eta^*_{\bar{\imath}}) \\ \text{with} & & \beta_{\bar{\imath}}, \beta^*_{\bar{\imath}}, \xi_{\bar{\imath}}, \xi^*_{\bar{\imath}} \ge 0. \end{array} \tag{57}$$

In (57), the regularization term is expressed in terms of the function expansion coefficients $\alpha_{\bar{\imath}}$. We next do the same for the terms stemming from the constraints on $(\hat{A}_i f)(\mathbf{x}_{\bar{\imath}})$, and compute $\hat{A}_i f$ by substituting the expansion (15) to get

$$(\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}) = \sum_{\bar{\jmath}}\alpha_{\bar{\jmath}}((\hat{A}_j \otimes \hat{A}_i)k)(\mathbf{x}_{\bar{\jmath}}, \mathbf{x}_{\bar{\imath}}) + \hat{A}_i b = \sum_{\bar{\jmath}}\alpha_{\bar{\jmath}}K_{\bar{\jmath}\bar{\imath}} + \hat{A}_i b. \tag{58}$$

See (36) for definition of K. Now we can compute the derivatives with respect to the primal variables $\alpha_{\bar{\imath}}, b, \eta_{\bar{\imath}}$. These have to vanish for optimality.

$$\frac{\partial}{\partial\alpha_{\bar{\jmath}}}L = \sum_{\bar{\imath}}(D_{\bar{\jmath}\bar{\imath}}\alpha_{\bar{\imath}} - K_{\bar{\jmath}\bar{\imath}}(\beta_{\bar{\imath}} - \beta^*_{\bar{\imath}})) = 0 \tag{59}$$

Solving (59) for $\vec{\alpha}$ yields

$$\vec{\alpha} = D^{-1}K(\vec{\beta} - \vec{\beta}^*), \tag{60}$$

where D^{-1} is the pseudoinverse in case D does not have full rank. We proceed to the next Lagrange condition, reading

$$\frac{1}{b}\cdot\frac{\partial}{\partial b}L = \sum_{\bar{\imath}}(\hat{A}_i 1)(\beta^*_{\bar{\imath}} - \beta_{\bar{\imath}}) = 0, \tag{61}$$

using $\hat{A}_i b = b\hat{A}_i 1$. Summands for which $(\hat{A}_i 1) = 0$ vanish, thereby removing the constraint imposed by (61) on the corresponding variables. Partial differentiation with respect to $\eta_{\bar{\imath}}$ and $\eta^*_{\bar{\imath}}$ yields

$$\frac{1}{\lambda}\frac{d}{d\eta_{\bar{\imath}}}c_{\bar{\imath}}(\eta_{\bar{\imath}}) = \beta_{\bar{\imath}} + \xi_{\bar{\imath}} \text{ and } \frac{1}{\lambda}\frac{d}{d\eta^*_{\bar{\imath}}}c^*_{\bar{\imath}}(\eta^*_{\bar{\imath}}) = \beta^*_{\bar{\imath}} + \xi^*_{\bar{\imath}} \tag{62}$$

Now we may substitute (60), (61), and (62) back into (57), taking into account the substitution (58), and eliminate $\alpha_{\bar{\imath}}$ and $\xi_{\bar{\imath}}$, obtaining

$$
\begin{aligned}
L &= \tfrac{1}{\lambda}\sum_{\bar{\imath}}(c_{\bar{\imath}}(\eta_{\bar{\imath}}) - \eta_{\bar{\imath}}\tfrac{d}{d\eta_{\bar{\imath}}}c_{\bar{\imath}}(\eta_{\bar{\imath}}) + c^*_{\bar{\imath}}(\eta^*_{\bar{\imath}}) - \eta^*_{\bar{\imath}}\tfrac{d}{d\eta^*_{\bar{\imath}}}c^*_{\bar{\imath}}(\eta^*_{\bar{\imath}})) + \\
&\quad (\vec{\beta}^* - \vec{\beta})^\top \vec{y} - (\vec{\beta}^* + \vec{\beta})^\top \vec{\epsilon} - \tfrac{1}{2}(\vec{\beta}^* - \vec{\beta})^\top K D^{-1} K (\vec{\beta}^* - \vec{\beta})
\end{aligned}
\tag{63}
$$

The next step is to fill in the explicit form of the cost functions $c_{\bar{\imath}}$, which will enable us to eliminate $\eta_{\bar{\imath}}$, with programming problems in the $\beta_{\bar{\imath}}$ remaining. However (as one can see) each of the $c_{\bar{\imath}}$ and $c^*_{\bar{\imath}}$ may have its own special functional form. Therefore we will carry out the further calculations with

$$
T(\eta) := \frac{1}{\lambda}(c(\eta) - \eta\frac{d}{d\eta}c(\eta)) \tag{64}
$$

and

$$
\frac{1}{\lambda}\frac{d}{d\eta}c(\eta) = \beta + \xi, \tag{65}
$$

where $(\bar{\imath})$ and possible asterisks have been omitted for clarity. This leads to

$$
L = \sum_{\bar{\imath}} T_{\bar{\imath}}(\eta_{\bar{\imath}}) + T^*_{\bar{\imath}}(\eta^*_{\bar{\imath}}) + (\vec{\beta}^* - \vec{\beta})^\top \vec{y} - (\vec{\beta}^* + \vec{\beta})^\top \vec{\epsilon} - \frac{1}{2}(\vec{\beta}^* - \vec{\beta})^\top K D^{-1} K (\vec{\beta}^* - \vec{\beta}) \tag{66}
$$

Example 12 (Polynomial Loss Functions)
Let us assume the general case of functions with ϵ-insensitive loss zone (which may vanish, if $\epsilon = 0$) and polynomial loss of degree $p > 1$. In [8] this type of cost functions was used for pattern recognition. This contains all L_p loss functions as special cases ($\epsilon = 0$), except for the case of $p = 1$, which will be treated in Example 13. We use

$$
c(\eta) = \frac{1}{p}\eta^p. \tag{67}
$$

From Eqs. (64), (65), and (67) it follows that

$$
\begin{aligned}
\frac{1}{\lambda}\eta^{p-1} &= \beta + \xi && (68)\\
T(\eta) &= \frac{1}{\lambda}\left(\frac{1}{p}\eta^p - \eta\eta^{p-1}\right) = -\left(1 - \frac{1}{p}\right)\lambda^{\frac{1}{p-1}}(\beta + \xi)^{\frac{p}{p-1}} && (69)
\end{aligned}
$$

As we want to find the maximum of L in terms of the dual variables we get $\xi = 0$ as T is the only term where ξ appears and T becomes maximal for that value. This yields

$$
T(\beta) = -\left(1 - \frac{1}{p}\right)\lambda^{\frac{1}{p-1}}\beta^{\frac{p}{p-1}} \quad \text{with} \quad \beta \in \mathbb{R}_0^+. \tag{70}
$$

Moreover we have the following relation between β and η:

$$
\eta = (\lambda\beta)^{\frac{1}{p-1}} \tag{71}
$$

Example 13 (Piecewise Polynomial and Linear Loss Functions)
Here we discuss cost functions with polynomial growth for $[0, \sigma]$ with $\sigma \geq 0$ and linear growth for $[\sigma, \infty)$ such that $c(\eta)$ is C^1 and convex. A consequence of the linear growth for large η is that the range of the Lagrange multipliers becomes bounded, namely by the derivative of $c(\eta)$. Therefore we will have to solve box constrained

optimization problems.

$$c(\eta) = \begin{cases} \sigma^{1-p}\frac{1}{p}\,\eta^p & \textit{for } \eta < \sigma \\ \eta + \left(\frac{1}{p}-1\right)\sigma & \textit{for } \eta \geq \sigma \end{cases} \tag{72}$$

$$T(\eta) = \frac{1}{\lambda}\begin{cases} -\sigma^{1-p}\,\left(1-\frac{1}{p}\right)\eta^p & \textit{for } \eta < \sigma \\ -\sigma\,\left(1-\frac{1}{p}\right) & \textit{for } \eta \geq \sigma \end{cases} \tag{73}$$

$$\beta + \xi = \frac{1}{\lambda}\begin{cases} \sigma^{1-p}\,\eta^{p-1} & \textit{for } \eta < \sigma \\ 1 & \textit{for } \eta \geq \sigma \end{cases} \tag{74}$$

By the same reasoning as above we find that the optimal solution is obtained for $\xi = 0$. Furthermore we can see through the convexity of $c(\eta)$ that $\eta < \sigma$ iff $\beta < \frac{1}{\lambda}$. Hence we may easily substitute β for $\frac{1}{\lambda}$ in the case of $\eta > \sigma$. $\beta \in [0, \frac{1}{\lambda}]$ is always true as $\xi \geq 0$. Combining these findings leads to a simplification of (73).

$$T(\beta) = -\lambda^{\frac{1}{p-1}}\left(1-\frac{1}{p}\right)\sigma\beta^{\frac{p}{p-1}} \quad \textit{for} \quad \sigma \in \mathbb{R}_0^+ \tag{75}$$

Analogously to Example 12 we can determine the error for $\beta \in [0, \frac{1}{\lambda})$ by

$$\eta = \sigma(\lambda\beta)^{\frac{1}{p-1}} \tag{76}$$

Example 14 (Hard ϵ-constraints)
The simplest case to consider, however, are hard constraints, i.e. the requirement that the approximation of the data is performed with at most ϵ deviation. In this case defining a cost function does not make much sense in the Lagrange framework and we may skip all terms containing $\eta_{ij}^{(*)}$. This leads to a simplified optimization problem:

$$\begin{aligned} &\textit{maximize} \quad (\vec\beta^* - \vec\beta)^\top \vec y - (\vec\beta^* + \vec\beta)^\top \vec\epsilon - \frac{1}{2}(\vec\beta^* - \vec\beta)^\top K D^{-1} K (\vec\beta^* - \vec\beta) \\ &\textit{subject to} \quad \sum_{\bar\imath}(\hat A_i 1)(\beta_{\bar\imath}^* - \beta_{\bar\imath}) = 0, \quad \beta_{\bar\imath}, \beta_{\bar\imath}^* \in \mathbb{R}_0^+ \end{aligned} \tag{77}$$

Another way to to see this is to use the result of Example 6 and take the limit $\lambda \to 0$. Loosely speaking, the interval $[0, \frac{1}{\lambda}]$ then converges to $\mathbb{R}_0^+$.

B Proof of Theorem 1

We modify the proof given in [7] to deal with the more general case stated in Theorem 1. As R_{reg} is convex for all $\lambda \geq 0$, minimization of R_{reg} is equivalent to fulfilling the Euler–Lagrange equations. Thus a necessary and sufficient condition for $f \in \mathcal{H}(V)$ to minimize R_{reg} on $\mathcal{H}(V)$ is that the Gateaux functional derivative [13] $\frac{\delta}{\delta f} R_{reg}[f, \psi]$ vanish for all $\psi \in \mathcal{H}(V)$. We get

$$\begin{aligned} \frac{\delta}{\delta f} R_{reg}[f,\psi] &= \lim_{k\to 0} \frac{R_{reg}[f + k\psi] - R_{reg}[f]}{k} \\ &= \lim_{k\to 0} \frac{1}{k}\left[\sum_{\bar\imath} \frac{1}{\ell_i} c_i((\hat A_i(f + k\psi))(\mathbf{x}_{\bar\imath}), \mathbf{x}_{\bar\imath}, y_{\bar\imath}) - \right. \\ &\quad \left. \sum_{\bar\imath} \frac{1}{\ell_i} c_i((\hat A_i f)(\mathbf{x}_{\bar\imath}), \mathbf{x}_{\bar\imath}, y_{\bar\imath}) + \frac{\lambda}{2}\left(\|\hat P(f + k\psi)\|_{\mathcal{D}}^2 - \|\hat P f\|_{\mathcal{D}}^2\right)\right] \end{aligned} \tag{78}$$

Expanding (78) in terms of k and taking the limit $k \to 0$ yields

$$\frac{\delta}{\delta f} R_{reg}[f, \psi] = \sum_{\bar{\imath}} \frac{1}{\ell_i} \partial_1 c_i((\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}), \mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}})(\hat{A}_i \psi)(\mathbf{x}_{\bar{\imath}}) + \lambda(\hat{P} f \cdot \hat{P} \psi)_{\mathcal{D}}. \tag{79}$$

Equation (79) has to vanish for $f = f_{opt}$. As $\bar{\mathcal{D}}$ is a Hilbert space, we can define the adjoint $\hat{P}^*$ and get

$$(\hat{P} f \cdot \hat{P} \psi)_{\mathcal{D}} = (\hat{P}^* \hat{P} f \cdot \psi)_{\mathcal{H}(V)}. \tag{80}$$

Similarly, we rewrite the first term of (79) to get

$$\sum_{\bar{\imath}} \frac{1}{\ell_i} \partial_1 c_i((\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}), \mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}}) \left(\delta_{\mathbf{x}_{\bar{\imath}}} \cdot \hat{A}_i \psi\right)_{\mathcal{H}(V)} + \lambda \left(\hat{P}^* \hat{P} f \cdot \psi\right)_{\mathcal{H}(V)} \tag{81}$$

Using $(\delta_{\mathbf{x}_{\bar{\imath}}} \cdot \hat{A}_i \psi)_{\mathcal{H}(V)} = (\hat{A}_i^* \delta_{\mathbf{x}_{\bar{\imath}}} \cdot \psi)_{\mathcal{H}(V)}$, the whole expression (81) can be written as a dot product with ψ. As ψ was arbitrary, this proves the theorem.[5]

C An Algorithm for the Virtual Examples Case

We will discuss an application of this algorithm to the problem of optical character recognition. For the sake of simplicity we shall assume the case of a dichotomy problem, e.g. having to distinguish between the digits **0** and **1**, combined with a regularization operator of the Support Vector Type, i.e. $D = K$.

Let us start with an initial set of training data $X_0 = \{\mathbf{x}_{01}, \ldots, \mathbf{x}_{0\ell_0}\}$ together with class labels $Y_0 = \{y_{01}, \ldots, y_{0\ell_0}, \mid y_{0i} \in \{-1, 1\}\}$. Additionally we know that the decision function should be invariant under *small* translations, rotations, changes of line thickness, radial scaling, and slanting or deslanting operations.[6]

Assume transformations $\hat{T}_s$ associated with the aforementioned symmetries, together with confidence levels $C_s \leq 1$ regarding whether $\hat{T}_s \mathbf{x}_{0i}$ will still belong to class $\mathbf{y}_{0i}$. As in Example 9, we use $X_s := X_0$, $(\hat{A}_s f)(\mathbf{x}) := f(\hat{T}_s \mathbf{x})$ and $\hat{T}_0 := \mathbf{1}$. As we are dealing with the case of pattern recognition, i.e. we are only interested in $\operatorname{sgn}(f(\mathbf{x}))$, not in $f(\mathbf{x})$ itself, it is beneficial to use a corresponding cost function, namely the soft margin loss as described in [8].

$$c_0(f(\mathbf{x}), \mathbf{x}, y) = \begin{cases} 0 & \text{for } f(\mathbf{x})y \geq 1 \\ 1 - f(\mathbf{x})y & \text{otherwise} \end{cases} \tag{82}$$

For the transformed datasets X_s we define cost functions $c_s := C_s c_0$ (i.e. we are going to penalize errors on X_s less than on X_0). As the cost functions are 0 for an interval unbounded in one direction (either $(-\infty, 0]$ or $[0, \infty)$, depending on the

[5] Note that this can be generalized to the case of convex functions which need not be C^1. We next briefly sketch the modifications in the proof. Partial derivatives of c_i now become subdifferentials, with the consequence that the equations only have to hold for some variables

$$\alpha_{\bar{\imath}} \in \partial_1 c_i((\hat{A}_i f)(\mathbf{x}_{\bar{\imath}}), \mathbf{x}_{\bar{\imath}}, y_{\bar{\imath}}).$$

In this case, ∂_1 denotes the subdifferential of a function, which consists of an interval rather than just a single number. For the proof, we convolve the non-C^1 cost functions with a positive C^1 smoothing kernel which preserves convexity (thereby rendering them C^1)), and take the limit to smoothing kernels with infinitely small support. Convergence of the smoothed cost functions to the non-smooth originals is exploited.

[6] Unfortunately no general rule can be given on the number or the extent of these transformations, as they depend heavily on the data at hand. A database containing only a very few (but very typical) instances of a class may benefit from a large number of additional virtual examples. A large database instead possibly already may contain realizations of the invariances in an explicit manner.

class labels), half of the Lagrange multipliers vanish. Therefore our setting can be simplified by using $\gamma_{\bar{\imath}} := \alpha_{\bar{\imath}} y_{\bar{\imath}}$ instead of $\alpha_{\bar{\imath}}$, i.e.

$$f(\mathbf{x}) = \sum_{\bar{\imath}} y_{\bar{\imath}} \gamma_{\bar{\imath}} (\hat{A}_i k)(\mathbf{x}_{\bar{\imath}}, \mathbf{x}) + b. \tag{83}$$

This allows us to get rid of the asterisks in the optimization problem, reading

$$\text{maximize} \quad \sum_{\bar{\imath}} \gamma_{\bar{\imath}} - \frac{1}{2} \vec{\gamma}^\top \bar{K} \vec{\gamma} \quad \text{subject to} \quad \sum_{\bar{\imath}} y_{\bar{\imath}} \gamma_{\bar{\imath}} = 0, \quad \gamma_{\bar{\imath}} \in \left[0, \frac{C_i}{\lambda}\right], \tag{84}$$

with

$$\bar{K}_{\bar{\imath}\bar{\jmath}} := k(\hat{T}_i \mathbf{x}_{\bar{\imath}}, \hat{T}_j \mathbf{x}_{\bar{\jmath}}) y_{\bar{\imath}} y_{\bar{\jmath}}. \tag{85}$$

The fact that less confidence has been put on the tranformed samples $\hat{T}_s \mathbf{x}_{0i}$ leads to a decrease in the upper boundary $\frac{C_s}{\lambda}$ for the corresponding Lagrange multipliers. In this point our algorithm differs from the Virtual Support Vector algorithm as proposed in [32]. Morover, their algorithm proceeds in two stages by first finding the Support Vectors and then training on a database generated only from the Support Vectors and their transforms.

If one was to tackle the quadratic programming problem with all variables at a time, the proposed algorithm would incur an substantial increase of computational complexity. However, only a small fraction of Lagrange multipliers corresponding to data relevant for the classification problem will differ from zero (e.g. [31]). Therefore it is advantageous to minimize the target function only on subsets of the $\bar{\alpha}_i$, keeping the other variables fixed (cf. [27]), possibly starting with the original dataset X_0.

References

[1] Y. S. Abu-Mostafa. Hints. *Neural Computation*, 7(4):639–671, 1995.

[2] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automat. Control*, 19(6):716–723, 1974.

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

[4] 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.

[5] J. R. Bunch and L. Kaufman. A computational method for the indefinite quadratic programming problem. *Linear Algebra and Its Applications*, pages 341–370, 1980.

[6] C. J. C. Burges. Simplified support vector decision rules. In *Proc. 13th International Conference on Machine Learning*, pages 71–77, San Mateo, CA, 1996. Morgan Kaufmann.

[7] S. Canu. Régularisation et l'apprentissage. work in progress, available from http://www.hds.utc.fr/˜scanu/regul.ps, 1996.

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

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

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

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

[12] 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.

[13] R. Gateaux. Sur les fonctionelles continues et les fonctionelles analytiques. *Bull. Soc. Math. France*, 50:1–21, 1922.

[14] S. Geva, J. Sitte, and G. Willshire. A one neuron truck backer–upper. In *International Joint Conference on Neural Networks*, pages 850–856, Baltimore, ML, 1992. IEEE.

[15] F. Girosi and G. Anzellotti. Convergence rates of approximation by translates. Technical Report AIM-1288, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, 1992.

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

[17] A. E. Hoerl and R. W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

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

[19] R. Klein, G. Liebich, and W. Straßer. Mesh reduction with error control. In R. Yagel, editor, *Visualization 96*, pages 311–318. ACM, 1996.

[20] T. K. Leen. From data distribution to regularization in invariant learning. *Neural Computation*, 7(5):974–981, 1995.

[21] J. C. Lemm. Prior information and generalized questions. Technical Report AIM-1598, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, 1996.

[22] David J. C. MacKay. *Bayesian Modelling and Neural Networks*. PhD thesis, Computation and Neural Systems, California Institute of Technology, Pasadena, CA, 1991.

[23] V. A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer Verlag, New York, NY, 1984.

[24] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. 1997. submitted to the International Conference on Neural Networks 97.

[25] N. Murata, S. Yoshizawa, and S. Amari. Network information criterion—determining the number of hidden units for artificial neural network models. *IEEE Transactions on Neural Networks*, 5:865–872, 1994.

[26] M. Z. Nashed and G. Wahba. Generalized inverses in reproducing kernel spaces: An approach to regularization of linear operator equations. *SIAM J. Math. Anal.*, 5(6):974–987, 1974.

[27] E. Osuna, R. Freund, and F. Girosi. Improved training algorithm for support vector machines. In *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, 1997. To appear in NNSP' 97.

[28] T. Poggio and F. Girosi. A theory of networks for approximation and learning. Technical Report AIM-1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, 1989.

[29] 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. ISBN 0-521-43108-5.

[30] J. Rissanen. Minimum-description-length principle. *Ann. Statist.*, 6:461–464, 1985.

[31] 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*, pages 252–257, Menlo Park, CA, 1995. AAAI Press.

[32] 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, 1996. Springer Lecture Notes in Computer Science, Vol. 1112.

[33] 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 Transactions on Signal Processing*, 1997. in press.

[34] P. Simard, Y. Le Cun, 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. Proceedings of the 1992 Conference*, pages 50–58, San Mateo, CA, 1993. Morgan Kaufmann.

[35] P. Simard, B. Victorri, Y. Le Cun, and J. Denker. Tangent prop — a formalism for specifying selected invariances in an adaptive network. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 895–903, San Mateo, CA, 1992. Morgan Kaufmann.

[36] A. J. Smola. Regression estimation with support vector learning machines. Master's thesis, Technische Universität München, Fakultät für Physik, 1996.

[37] E. M. Stein. *Singular Integrals and Differentiability Properties of Functions.* Princeton University Press, Princeton, NJ, 1970.

[38] A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill–Posed Problems.* Winston, Washington, DC, 1977.

[39] R. J. Vanderbei. LOQO: An interior point code for quadratic programming. Technical report, Program in Statistics & Operations Research, Princeton University, Princeton, NJ, 1994.

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

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

[42] V. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In M. C. Mozer, M. L. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, Cambridge, MA, 1997. MIT Press. in press.

[43] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, Berlin, 1982.

[44] J. Wood and J. Shawe-Taylor. A unifying framework for invariant pattern recognition. *Pattern Recognition Letters*, 17:1415–1422, 1996.