GMD Technical Report: 1064

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

Alex J. Smola* and Bernhard Schölkopf†

April 10, 1997

Abstract

We present a Kernel–based framework for Pattern Recognition, Regression Estimation, Function Approximation and multiple Operator Inversion. Previous approaches such as ridge-regression, Support Vector methods and regression by Smoothing Kernels 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. The optimal solution of all the problems described above can be found by solving a simple quadratic programming problem. The paper closes with a proof of the equivalence between Support Vector kernels and Greene's functions of regularization operators.

Acknowledgements:

We would like to thank Volker Blanz, Léon Bottou, Chris Burges, Patrick Haffner, Leo van Hemmen, Jörg Lemm, Klaus-Robert Müller, Heidrun Mündlein, Noboru Murata, Sara Solla, and Vladimir Vapnik 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. +49-030-63921900, Fax +49-030-63921805, E–mail smola@prosun.first.gmd.de

†MPI 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

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. [2]) to advanced methods like Generalized Additive Models [17], Neural Networks and Support Vectors [3]. In combination with different types of cost functions, as for instance quadratic ones, robust ones in Huber's sense [18], or ϵ–insensitive ones [36], these yield a wide variety of different training procedures which at first sight might seem incompatible with each other. The purpose of this paper, which was inspired by the treatments of [6] and [35], 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.

The setting of the problem which we present in this paper is based on a generalization of the well–known functionals comprising approximation quality plus complexity control to the case of solving multiple operator equations. As will be shown this is a viable approach both for pattern recognition and function estimation. The purpose of this paper is not to provide the most general setting possible, which would render parts of the proofs very tedious, but to provide insight in a new way of jointly solving multiple measurement problems by a kernel based approach.

Multiple operator equations 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.

We have organized the presentation as follows. In section 2, we state the mathematical setting of the problem and give expressions of both the risk and the empirical risk functional. Section 3 shows how a regularized solution of the empirical risk functional may be found and describes 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 posed previously. 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, section 6 contains practical applications of multiple operators to the case of problems with prior knowledge. Finally section 7 shows the connection between regularization operators and Hilbert-Schmidt kernels and proves the admissibility of a large class of radial basis functions as suitable kernels for Support Vector machines. Appendix A and B contain proofs of the formulas of section 4 and 5.

2 Risk Minimization

In Regression estimation we try to estimate a functional dependency f between a set of sampling points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ taken from a space V, and target values $Y = \{y_1, \ldots, y_l\}$. Let us now consider a situation where we cannot observe X, but some other corresponding points $X_s = \{\mathbf{x}_{s1}, \ldots, \mathbf{x}_{sl}\}$, nor can we observe Y, but $Y_s = \{y_{s1}, \ldots, y_{sl}\}$. Let us call the pairs $(\mathbf{x}_{si}, y_{si})$ *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 $\hat{T}$:

$$\mathbf{x}_{si} = \hat{T}\mathbf{x}_i, \quad (i = 1, \ldots, l). \tag{1}$$

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

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

If we know the transformation $A_{\hat{T}}$, 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 [37]. 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 (steering wheel left or right) problem 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 using the assumption of general linear operators, we formalize the above as follows. We consider pairs of observations $(\mathbf{x}_{ij}, y_{ij})$, with sampling points $\mathbf{x}_{ij}$ and corresponding target values y_{ij}. The index i denotes the procedure by which we have obtained the target values, and j runs over the observations $1..l_i$. We may group these pairs in q pairs of sets X_i and Y_i by defining

$$\begin{array}{rll} X_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{il_i}\} & \text{with} & \mathbf{x}_{ij} \in V_i \\ Y_i = \{y_{i1}, \ldots, y_{il_i}\} & \text{with} & y_{ij} \in \mathbb{R} \end{array} \tag{4}$$

with V_i being vector spaces.

Let us assume that these samples have been drawn independently from q corresponding probability density functions $p_1(\mathbf{x}_1, y_1), \ldots, p_q(\mathbf{x}_q, y_q)$ 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 $\hat{A}_1, \ldots, \hat{A}_q$ acting 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.

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

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

is minimized. (Sometimes we may want to restrict ourselves to subsets of $\mathcal{H}(V)$ in order to control the capacity of the admissible models.)

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.

Example 1 (Ordinary Least Mean Squares)
Setting

$$q = 1 \tag{7}$$
$$\hat{A} = \mathbf{1} \tag{8}$$
$$c(f(\mathbf{x}), \mathbf{x}, y) = (f(\mathbf{x}) - y)^2 \tag{9}$$

leads to

$$R[f] = \int (f(\mathbf{x}) - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy \tag{10}$$

This is the well–known definition of the least mean square error risk [10].

Example 2 (Common Definition of the Risk functional)
By specializing

$$q = 1 \tag{11}$$
$$\hat{A} = \mathbf{1} \tag{12}$$

we arrive at the definition of the risk functional of [35]:

$$R[f] = \int c(f, \mathbf{x}, y) p(\mathbf{x}, y) d\mathbf{x} dy \tag{13}$$

Obviously this includes Example 1 as a special case. Also note that eq. (13) 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 [35] (*c *is a functional of* f *and therefore it also may be a function of* $\hat{A}f$ *for some* $\hat{A}$*).*

Remark 3 (On Underlying Functional Dependences)
For each V_i *together with* p_i *we might define a function*

$$\bar{y}_i(\mathbf{x}_i) := \int y_i P_i(y_i|\mathbf{x}_i) dy_i \tag{14}$$

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 finding a pseudo-solution of the operator equation. For a detailed treatment see [24].*

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

$$f_{min} := \mathrm{argmin}_{\mathcal{H}(V)} R[f] \tag{15}$$

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:

$$R_{emp}[f] = \sum_{i=1}^{q} \frac{1}{l_i} \sum_{j=1}^{l_i} c_i \left((\hat{A}_i f)(\mathbf{x}_{ij}), \mathbf{x}_{ij}, y_{ij}) \right) \tag{16}$$

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 [10]. Therefore it is not advisable to minimize the empirical risk without any means of capacity control or regularization [35].

3 Regularization Operators and Additive Models

We will assume a regularization term in the spirit of [33] and [23], namely a positive semidefinite operator

$$\hat{P} : \mathcal{H}(V) \longmapsto \mathcal{D} \tag{17}$$

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

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2}||\hat{P}f||^2_{\mathcal{D}} \tag{18}$$

with a regularization parameter $\lambda \geq 0$. This additional term effectively reduces our model space and thereby controls the complexity of the solution. We do not require positive definiteness 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 (18), we assume f to be of the Additive Model type [17], 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 $\mathcal{H}(V) \otimes \mathcal{H}(V)$,

$$((\hat{A} \otimes \hat{B})k)(\mathbf{x}, \mathbf{x}') \tag{19}$$

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 (18) are expansions of the form

$$f(\mathbf{x}) = \sum_{i=1}^{q} \sum_{j=1}^{l_i} \alpha_{ij}((\hat{A}_i \otimes \mathbf{1})k)(\mathbf{x}_{ij}, \mathbf{x}) + b, \quad \text{with} \quad \alpha_{ij} \in \mathbb{R}. \tag{20}$$

This may seem to be a rather arbitrary assumption; however, kernel expansions of the type $\sum_{i=1}^{l} \alpha_i k(\mathbf{x}_i, \mathbf{x})$ are quite common in regression and pattern recognition models [17], and in the case of Support Vectors even follow naturally from optimality conditions [3], [37]. 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 $k_i(\mathbf{x}) := k(\mathbf{x}_i, \mathbf{x})$ 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 explicit expression for $||\hat{P}f||^2_{\mathcal{D}}$ in terms of the variables α_{ij}, we first note

$$(\hat{P}f)(\mathbf{x}) = \sum_{i=1}^{q} \sum_{j=1}^{l_i} \alpha_{ij}((\hat{A}_i \otimes \hat{P})k)(\mathbf{x}_{ij}, \mathbf{x}). \tag{21}$$

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

$$(\hat{P}f \cdot \hat{P}f) = \sum_{i,m=1}^{q} \sum_{j=1}^{l_i} \sum_{n=1}^{l_m} \alpha_{ij}\alpha_{mn}(((\hat{A}_i \otimes \hat{P})k)(\mathbf{x}_{ij}, \mathbf{x}) \cdot ((\hat{A}_m \otimes \hat{P})k)(\mathbf{x}_{mn}, \mathbf{x})). \tag{22}$$

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

$$D_{(ij)(mn)} := (((\hat{A}_i \otimes \hat{P})k)(\mathbf{x}_{ij}, .) \cdot ((\hat{A}_m \otimes \hat{P})k)(\mathbf{x}_{mn}, .)) \tag{23}$$

can be evaluated in closed form, allowing an efficient implementation (here, the dot in $k(\mathbf{x}_{ij}, .)$ means that k is considered as a function of its second argument, with $\mathbf{x}_{ij}$ fixed). Arranging ij and kl in dictionary order, positivity of Eq. (22) amounts to positivity of the regularization matrix D. Conversely, any positive semidefinite matrix will act as a regularization matrix. As we minimize the regularized risk (18), the functions corresponding to the largest eigenvalue of $D_{(ij)(kl)}$ will be attenuated most; functions with expansion coefficient vectors lying in the kernel of D, however, will not be dampened at all.

Example 4 (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

$$||f||_{H^{s,p}(V)} = \sum_{|\gamma| \leq s} ||\hat{D}^\gamma f||_{L_p} \tag{24}$$

where γ is a multi-index and $\hat{D}^\gamma$ is the derivative of order γ. A special case of the Sobolev embedding theorem [32] yields:

$$H^{s,p}(V) \subset C^k \quad \textit{for} \quad k \in \mathbb{N} \quad \textit{and} \quad s > k + \frac{d}{2} \tag{25}$$

Here d denotes the dimensionality of V. Moreover there exists a constant c such that

$$max_{|\gamma| \leq k} \; sup_{\mathbf{x} \in V} |\hat{D}^\gamma f(\mathbf{x})| \;\; \leq \;\; c ||f||_{H^{s,p}(V)} \tag{26}$$

i.e. convergence in the Sobolev norm enforces uniform convergence in the first k derivatives. This renders this norm especially favourable for regularization of the function expansion.

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

$$D_{(ij)(mn)} = \sum_{|\gamma| \leq s} (((\hat{A}_i \otimes \hat{D}^\gamma) k)(\mathbf{x}_{ij}, \mathbf{x}) \cdot ((\hat{A}_m \otimes \hat{D}^\gamma) k)(\mathbf{x}_{mn}, \mathbf{x})). \tag{27}$$

Example 5 (Support Vector Regularization)
Consider the case of Example 2, i.e. the case of standard regression estimation or pattern recognition. Let us consider functions which can be written as separating hyperplanes in some Hilbert space H,

$$f(\mathbf{x}) = (\Psi \cdot \Phi(\mathbf{x})) + b \tag{28}$$

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

$$\Psi = \sum_{j=1}^{l} \alpha_j \Phi(\mathbf{x}_j). \tag{29}$$

The regularization operator $\hat{P}$ is chosen such that $\hat{P}f = \Psi$ (in view of the expansion (20), 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 (28) on H. Moreover, Φ is *chosen such that we can express the terms* $(\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}))$ *as some symmetric function* $k(\mathbf{x}_i, \mathbf{x})$ *which can be computed without carrying out* Φ*. Thus the solution (28) reads*

$$f(\mathbf{x}) = \sum_{j=1}^{l} \alpha_j k(\mathbf{x}_j, \mathbf{x}) + b, \tag{30}$$

and the regularization term becomes

$$||\Psi||_H^2 = \sum_{j,k=1}^{l} \alpha_j \alpha_k k(\mathbf{x}_j, \mathbf{x}_k). \tag{31}$$

This leads to the optimization problem of [3]. The mapping Φ *need not be known explicitly: for any continuous symmetric kernel* k *satisfying Mercer's condition [8]*

$$\int f(x)k(x,y)f(y)\; dx\; dy > 0 \quad \textit{if} \quad f \in L_2 \backslash \{0\}, \tag{32}$$

we can expand k *into a series*

$$k(x,y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y) \tag{33}$$

with positive coefficients λ_i*, and*

$$(\phi_i \cdot \phi_j)_{L_2} = \delta_{ij} \tag{34}$$

for $i, j \in \mathbb{N}$*. In this case, we can construct the desired map* Φ*. The feature space* F *is either* $\mathbb{R}^N$ *for some* $N \in \mathbb{N}$ *or the space of sequences* $\mathbb{R}^{\mathbb{N}}$*, depending on whether the above sum has infinitely many nonzero terms. Let* $(\mathbf{x}_i)_i$ *be an orthonormal basis of* F *with respect to a given dot product in* F*, then*

$$\Phi(\mathbf{x}) := \sum_{i=1}^{\infty} \sqrt{\lambda_i} \phi_i(\mathbf{x})\; \mathbf{x}_i \tag{35}$$

is a map into F *such that* k *acts as the given dot product, i.e.*

$$(\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})) = k(\mathbf{x}, \mathbf{y}). \tag{36}$$

In particular, this implies that the matrix $D_{jk} = k(\mathbf{x}_j, \mathbf{x}_k)$ *is positive semidefinite. 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 [11].*

It should be noted that the expansion of Ψ *in terms of the images of the data follows more naturally if viewed in the Support Vector context [36]; however the idea of selecting the flattest function in a high dimensional space is preserved in the present exposition.*

Example 6 (Ridge Regression)
If we define $\hat{P}$ *such that all functions used in the expansion of* f *are attenuated equally,* D *becomes the identity matrix,* $D_{(ij)(kl)} = \delta_{(ij)(kl)}$*. This leads to*

$$||\hat{P}f||_{\mathcal{D}}^2 = \sum_{i,m=1}^{q} \sum_{j=1}^{l_i} \sum_{n=1}^{l_m} \alpha_{ij} \alpha_{mn} D_{(ij)(mn)} = \sum_{i=1}^{q} \sum_{j=1}^{l_i} \alpha_{ij}^2 \tag{37}$$

and

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2} \sum_{i=1}^{q} \sum_{j=1}^{l_i} \alpha_{ij}^2, \tag{38}$$

which is exactly the definition of a ridge–regularized risk functional.

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

4 Risk Minimization by Quadratic Programming

To minimize the regularized risk (18), we only require the cost functions to be convex in the first argument and

$$c_i(y_i, \mathbf{x}_i, y_i) = 0 \text{ for all } \mathbf{x}_i \in V_i. \tag{39}$$

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

$$\begin{array}{rcll} c_{ij}(\eta_{ij}) & := & \frac{1}{l_i} c_i(y_{ij} + \epsilon_{ij} + \eta_{ij}, \mathbf{x}_{ij}, y_{ij}) & \text{with} \quad \eta_{ij} \geq 0 \\ c_{ij}^*(\eta_{ij}^*) & := & \frac{1}{l_i} c_i(y_{ij} - \epsilon_{ij}^* - \eta_{ij}^*, \mathbf{x}_{ij}, y_{ij}) & \text{with} \quad \eta_{ij}^* \geq 0 \end{array} \tag{40}$$

with $\mathbf{x}_{ij}$ and y_{ij} fixed and

$$\begin{array}{rcl} \eta_{ij} & := & \max\left((\hat{A}_i f)(\mathbf{x}_{ij}) - y_{ij} - \epsilon_{ij}, 0\right) \\ \eta_{ij}^* & := & \max\left(-(\hat{A}_i f)(\mathbf{x}_{ij}) + y_{ij} - \epsilon_{ij}^*, 0\right) \end{array} \tag{41}$$

The asterisk is used for distinguishing positive and negative slack variables and corresponding cost functions.

In pattern recognition problems, the intervals $[-\epsilon_{ij}^*, \epsilon_{ij}]$ 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 a set of slack variables, to render the subsequent calculus more amenable.

$$\begin{array}{ll} \text{minimize} & \frac{1}{\lambda} R_{reg} = \frac{1}{\lambda} \sum\limits_{i=1}^{q} \sum\limits_{j=1}^{l_i} (c_{ij}(\eta_{ij}) + c_{ij}^*(\eta_{ij}^*)) + \frac{1}{2} ||\hat{P} f||_{\mathcal{D}}^2 \\ \text{subject to} & (\hat{A}_i f)(\mathbf{x}_{ij}) \leq y_{ij} + \epsilon_{ij} + \eta_{ij} \\ & (\hat{A}_i f)(\mathbf{x}_{ij}) \geq y_{ij} - \epsilon_{ij}^* - \eta_{ij}^* \\ & \eta_{ij}, \eta_{ij}^* \geq 0 \end{array} \tag{42}$$

The problem can be solved 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 7 (Quadratic Cost Function)
We use Eq. (111) (derived in Appendix A, Example 24) in the special case

$$p = 2, \epsilon = 0 \tag{43}$$

to get the following unconstrained optimization problem:

$$\begin{array}{ll} \textit{maximize} & \sum\limits_{i=1}^{q} \sum\limits_{j=1}^{l_i} (\beta_{ij}^* - \beta_{ij}) y_{ij} - \frac{\lambda}{2} (\beta_{ij}^2 + {\beta_{ij}^*}^2) - \\ & \frac{1}{2} \sum\limits_{i=1}^{q} \sum\limits_{j=1}^{l_i} \sum\limits_{m=1}^{q} \sum\limits_{n=1}^{l_m} (\beta_{ij} - \beta_{ij}^*)(\beta_{mn} - \beta_{mn}^*)(K D^{-1} K)_{(ij)(mn)} \\ \textit{subject to} & \sum\limits_{i=1}^{q} (\hat{A}_i 1) \sum\limits_{j=1}^{l_i} (\beta_{ij}^* - \beta_{ij}) = 0 \\ & \beta_{ij}, \beta_{ij}^* \in \mathbb{R}_0^+ \end{array} \tag{44}$$

Transformation back to α_{ij} *is done by*

$$\vec{\alpha} = D^{-1} K (\vec{\beta} - \vec{\beta}^*) \tag{45}$$

Here, the symmetric matrix K is defined as

$$K_{(mn)(ij)} := ((\hat{A}_m \otimes \hat{A}_i)k)(\mathbf{x}_{mn}, \mathbf{x}_{ij}), \tag{46}$$

$(\hat{A}_i 1)$ is the operator $\hat{A}_i$ acting on the constant function with value 1, and $\vec{\alpha}$ denotes the vector of all α_{ij} with the index (ij) sorted in dictionary order (the same applies to the indices of K and D).

Of course there would have been a simpler solution to this problem (by combining β_{ij} and β^*_{ij} 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 8 (ϵ-insensitive Cost Function)
Here we use Eq. (115) (Appendix A, Example 25) for $\sigma = 0$. This leads to

$$\begin{array}{ll} \textit{maximize} & \sum_{i=1}^{q} \sum_{j=1}^{l_i} (\beta^*_{ij}(y_{ij} - \epsilon_{ij}) - \beta_{ij}(y_{ij} + \epsilon_{ij})) - \\ & \frac{1}{2} \sum_{i=1}^{q} \sum_{j=1}^{l_i} \sum_{m=1}^{q} \sum_{n=1}^{l_m} (\beta_{ij} - \beta^*_{ij})(\beta_{mn} - \beta^*_{mn})(KD^{-1}K)_{(ij)(mn)} \\ \textit{subject to} & \sum_{i=1}^{q} (\hat{A}_i 1) \sum_{j=1}^{l_i} (\beta^*_{ij} - \beta_{ij}) = 0 \\ & \beta_{ij}, \beta^*_{ij} \in [0, \frac{1}{\lambda}] \end{array} \tag{47}$$

with the same back substitution rules (45) as in Example 7. For the special case of Support Vector regularization this leads to exactly the same equations as in Support Vector Pattern Recognition or Regression Estimation [36]. 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 (30) and (31) with the definitions of D and K.

Example 9 (Laplace's Cost Function)
This is a special case of Example 8 for $\epsilon = 0$, with the same maximization problem as described above.

Example 10 (Huber's robust Cost Function)
Setting

$$p = 2, \epsilon = 0 \tag{48}$$

in Example 25 leads to the following optimization problem

$$\begin{array}{ll} \textit{maximize} & \sum_{i=1}^{q} \sum_{j=1}^{l_i} (\beta^*_{ij} - \beta_{ij}) y_{ij} - \frac{\lambda}{2}(\sigma_{ij}\beta^2_{ij} + \sigma^*_{ij}{\beta_{ij}}^2) - \\ & \frac{1}{2} \sum_{i=1}^{q} \sum_{j=1}^{l_i} \sum_{m=1}^{q} \sum_{n=1}^{l_m} (\beta_{ij} - \beta^*_{ij})(\beta_{mn} - \beta^*_{mn})(KD^{-1}K)_{(ij)(mn)} \\ \textit{subject to} & \sum_{i=1}^{q} (\hat{A}_i 1) \sum_{j=1}^{l_i} (\beta^*_{ij} - \beta_{ij}) = 0 \\ & \beta_{ij}, \beta^*_{ij} \in [0, \frac{1}{\lambda}] \end{array} \tag{49}$$

with the same backsubstitution rules (45) as in Example 7.

Remark 11 (Combinations of Cost Functions)
The cost functions described in the Examples 7, 8, 9, 10, 24, and 25 may be linearly combined into more complicated ones. However, this results in using additional Lagrange multipliers for practical purposes as each of the cost functions mentioned

before 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).

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 [25] and [6]. Like in section 4 we will require the cost functions c_i to be convex and C^1 in the first argument with the extra requirement

$$c_i(y_i, \mathbf{x}_i, y_i) = 0 \quad \text{for all} \quad \mathbf{x}_i \in V_i \quad \text{and} \quad y_i \in \mathbb{R}. \tag{50}$$

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

$$\begin{aligned} \hat{P} : &\quad \mathcal{H}(V) \to \mathcal{D} \\ \hat{P}^* : &\quad \bar{\mathcal{D}} \to \hat{P}^* \bar{\mathcal{D}} \subseteq \mathcal{H}(V). \end{aligned} \tag{51}$$

Theorem 12 (Optimality Condition)
Under the assumptions stated above a necessary and sufficient condition for

$$f = f_{opt} := argmin_{f \in \mathcal{H}(V)} R_{reg}[f] \tag{52}$$

is that the following equation holds true on $\mathcal{H}'$:

$$\hat{P}^* \hat{P} f = -\frac{1}{\lambda} \sum_{i=1}^{q} \frac{1}{l_i} \sum_{j=1}^{l_i} \partial_1 c_i \left((\hat{A}_i f)(\mathbf{x}_{ij}), \mathbf{x}_{ij}, y_{ij} \right) \hat{A}_i^* \delta_{\mathbf{x}_{ij}} \tag{53}$$

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

In order to illustrate the theorem above 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. According to the theory of Greene's functions there is a function $G(\mathbf{x}, \mathbf{x}_j)$ corresponding to the operator $\hat{P}^* \hat{P}$ such that

$$(\hat{P}^* \hat{P} G)(\mathbf{x}, \mathbf{x}_j) = \delta_{\mathbf{x}_j}(\mathbf{x}) \tag{54}$$

as previously described in [25]. In this case we derive from (53) the following system of equations which has to be solved in a self consistent manner.

$$\begin{aligned} f(\mathbf{x}) &= \sum_{j=1}^{l} \gamma_j G(\mathbf{x}, \mathbf{x}_j) + b & (55) \\ \text{with} \quad \gamma_j &= -\partial_1 c(f(\mathbf{x}_j), \mathbf{x}_j, y_j) & (56) \end{aligned}$$

Here the expansion of f in terms of kernel functions follows naturally with γ_j corresponding to Lagrange multipliers. For a more detailed treatment see section 7.

For the general case (54) becomes a little bit more complicated, namely we will have q functions $G_i(\mathbf{x}, \mathbf{x}_{ij})$ such that

$$(\hat{P}^* \hat{P} G)(\mathbf{x}, \mathbf{x}_{ij}) = (\hat{A}_i^* \delta)(\mathbf{x}_i - \mathbf{x}_{ij}) \tag{57}$$

holds. In [25] 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 might be applied to the case of estimating functional dependencies from indirect measurements. Moreover eq. (53) 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 shall discuss some examples of incorporating domain knowledge by using multiple operator equations as contained in Eq. (6).

Example 13 (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 an extra set of sampling points* $X_s = \{\mathbf{x}_{s1}, \ldots, \mathbf{x}_{sl_s}\} \subset X$ *with corresponding target values* $Y_s = \{y_{s1}, \ldots, y_{sl_s}\} \subset Y$*, an operator* $\hat{A}_s = \mathbf{1}$*, and a cost function (defined on* X_s*)*

$$c_s(f(\mathbf{x}_{si}), \mathbf{x}_{si}, y_{si}) = \begin{cases} 0 & \text{if } -\epsilon_i \leq f(\mathbf{x}_{si}) - y_{si} \leq \epsilon_i^* \\ \infty & \text{otherwise} \end{cases} \tag{58}$$

defined in terms of $\epsilon_1, \ldots, \epsilon_{l_s}$ *and* $\epsilon_1^*, \ldots, \epsilon_{l_s}^*$.

These additional hard constraints result in optimization problems similar to those obtained in the ϵ–insensitive approach of Support Vector regression [37]. See Example 26 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

$$\hat{A}_s = (\frac{\partial}{\partial \mathbf{x}})^p \tag{59}$$

instead of the $\hat{A}_s = \mathbf{1}$ used above. This requires differentiability of the function expansion of f. If we want to use more general expansions (20), we have to resort to finite difference operators.

Example 14 (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 [29]. We then define corresponding linear operators* $\hat{A}_i$ *acting on* $\mathcal{H}(V)$ *as in Eq. (2).*

As the empirical risk functional (16) contains a sum over original and transformed ("virtual") patterns, this corresponds to training on an artificially enlarged data set. Unlike previous approaches as the one of [27], 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. (103)). Still there are more general types of symmetries, especially non–deterministic ones. For an extended discussion of this topic see [21].

Example 15 (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* $f(\mathbf{x}_2)$*,*

$$(\hat{A}_s f)(\mathbf{x}_1 \oplus \mathbf{x}_2) := f(\mathbf{x}_1) - f(\mathbf{x}_2). \tag{60}$$

The technique of Example 13 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. 59), we can modify the above methods using derivatives of f. This will lead to tangent regularizers as the ones proposed by [30], as we shall presently show.

Example 16 (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

$$(g \cdot f)(\mathbf{x}) := f(g\mathbf{x}) \text{ for } g \in G, f \in \mathcal{H}(V). \tag{61}$$

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

$$(\hat{T}_i f)(\mathbf{x}) = 0 \quad \text{for } i = 1, \ldots, r. \tag{62}$$

To motivate this, note that

$$\begin{aligned} \frac{\partial}{\partial \alpha_i}\Big|_{\boldsymbol{\alpha}=0} f\Big(\exp\Big(\sum_j \alpha_j \hat{T}_j\Big)\mathbf{x}\Big) &= \Big(\Big(\frac{\partial}{\partial \alpha_i}\Big|_{\boldsymbol{\alpha}=0} \exp\Big(\sum_j \alpha_j \hat{T}_j\Big)\Big) f\Big)(\mathbf{x}) \\ &= (\hat{T}_i f)(\mathbf{x}), \end{aligned} \tag{63}$$

using Eq. (61), the chain rule, and the identity $\exp(0) = \mathbf{1}$.

Examples of operators $\hat{T}_i$ that can be used are derivative operators, which are the generators of translations. Operator equations of the type (62) allow us to use virtual examples which incorporate knowledge about derivatives of f. In the sense of [30], 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 14) as it might appear superficially.

As in Example 13, prior knowledge could also be given in terms of allowed ranges or cost functions [20] for not exact symmetries rather than enforced equalities as (62). Moreover, we can apply the approach of Example 16 as well to higher–order derivatives, generalizing what we said above about additional constraints on the estimated function (Example 13).

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 On Hilbert Schmidt Kernels and Regularization Operators

In this section, we will for simplicity only discuss the special setting of example 2 — a generalization to the case of multiple operators is rather straightforward. In example 5 (Support Vector Machines), we postulated an operator $\hat{P}$ such that

$$||\hat{P}f||^2_{\mathcal{D}} = \sum_{j,k=1}^{l} \alpha_j \alpha_k k(\mathbf{x}_j, \mathbf{x}_k). \tag{64}$$

with

$$f(\mathbf{x}) = \sum_{j=1}^{l} \alpha_j k(\mathbf{x}_j, \mathbf{x}) + b. \tag{65}$$

The advantage of this is that a (possibly) sparse decomposition of f in terms of β is preserved in α as both sets of variables coincide ($D = K$, hence $D^{-1}K = \mathbf{1}$, see (23) and (46) for definitions). This gives rise to the following two problems:

- For a given regularization operator $\hat{P}$, find a function $k(\mathbf{x}_i, \mathbf{x}_j)$ such that $D = K$ for all possible $\mathbf{x}_i, \mathbf{x}_j$ in the domain of interest.
- Which regularization operator $\hat{P}$ corresponds to a special choice of kernel functions $k(\mathbf{x}_i, \mathbf{x}_j)$?

These problems are of considerable interest in the context of support vector methods, since so far no theory has been developed regarding the optimal choice of kernel functions for a given problem. In an attempt to deal with these problems, we shall make use of the results of [25] in the context of Hilbert Schmidt kernels.

Due to (64), (65) and $\hat{P}b = 0$, it is equivalent to require element–wise coincidence of the two matrices D and K for all possible $\mathbf{x}_i, \mathbf{x}_j$ in the domain of interest:

$$\Big((\hat{P}k)(\mathbf{x}_i, .) \cdot (\hat{P}k)(\mathbf{x}_j, .)\Big) = k(\mathbf{x}_i, \mathbf{x}_j). \tag{66}$$

Recalling that Green's functions $G_{\mathbf{x}_i}(\mathbf{x})$ of $\hat{P}^*\hat{P}$ satisfy

$$(\hat{P}^*\hat{P}G_{\mathbf{x}_i})(\mathbf{x}) = \delta_{\mathbf{x}_i}(\mathbf{x}) \tag{67}$$

leads to

Theorem 17 (Green's functions and Hilbert Schmidt Kernels)
Given a regularization operator $\hat{P}$, Green's function G of $\hat{P}^*\hat{P}$ is a Hilbert Schmidt kernel such that $D = K$. Moreover Support Vector Machines using $G(\mathbf{x}_i, \mathbf{x}_j)$ as a kernel function correspond to minimizing the regularized risk functional with $\hat{P}$ as regularization operator.

Proof:
Substituting (67) into $G_{\mathbf{x}_j}(\mathbf{x}_i) = \big(G_{\mathbf{x}_j}(\mathbf{x}) \cdot \delta_{\mathbf{x}_i}(\mathbf{x})\big)$ yields

$$G_{\mathbf{x}_j}(\mathbf{x}_i) = \Big((\hat{P}G_{\mathbf{x}_i})(\mathbf{x}) \cdot (\hat{P}G_{\mathbf{x}_j})(\mathbf{x})\Big) = G_{\mathbf{x}_i}(\mathbf{x}_j), \tag{68}$$

thus G is a symmetric function in $\mathbf{x}_i$ and $\mathbf{x}_j$ ($G(\mathbf{x}_i, \mathbf{x}_j) = G(\mathbf{x}_j, \mathbf{x}_i)$) satisfying (66), and therefore $D = K$. Furthermore it is an admissible positive kernel as it can be written as a dot product in another space, namely

$$G(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) \tag{69}$$

with

$$\Phi : \mathbf{x}_i \longmapsto (\hat{P}G_{\mathbf{x}_i})(\mathbf{x}). \tag{70}$$

Now suppose we have a Support Vector Machine using G as kernel function. This leads to an identical optimization problem and an identical expansion in f for any possible choice of the training data. Therefore these two methods can be considered equivalent which proves the theorem.

This connection allows us to determine the optimal support vector kernel given a corresponding optimal regularization operator for a specific problem.

Let us now more specifically consider regularization operators $\hat{P}$ that may be written as multiplications in Fourier space

$$||\hat{P}f||^2 = \int_\Omega \frac{\overline{\tilde{f}(\boldsymbol{\omega})}\tilde{f}(\boldsymbol{\omega})}{P(\boldsymbol{\omega})} d\boldsymbol{\omega} \tag{71}$$

with $\tilde{f}(\boldsymbol{\omega})$ denoting the Fourier transform of $f(\mathbf{x})$, $P(\boldsymbol{\omega}) \geq 0$ and $P(\boldsymbol{\omega})$ converges uniformly to 0 for $|\boldsymbol{\omega}| \to \infty$ and $\Omega \subseteq \text{supp}[P(\boldsymbol{\omega})]$. Here (71) weakly defines $\hat{P}^*\hat{P}$ as by using the polarization identity we can derive $(\hat{P}f \cdot \hat{P}g)$ from it. This is sufficient for our purpose as we are only interested in the effect of $\hat{P}^*\hat{P}$ (usually, there exist different operators $\hat{P}$ which yield the same $\hat{P}^*\hat{P}$). Moreover it follows from (71) together with (68) that $\tilde{G}(\boldsymbol{\omega}) = P(\boldsymbol{\omega})$. By using the Fourier Plancherel identity one can show that Mercer's condition

$$\int\int d\mathbf{x}_i d\mathbf{x}_j \; k(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_i) g(\mathbf{x}_j) \geq 0 \tag{72}$$

for translation invariant kernels $k(\mathbf{x}_i - \mathbf{x}_j)$ transforms to

$$\int d\boldsymbol{\omega} \; \tilde{k}(\boldsymbol{\omega}) \overline{\tilde{g}(\boldsymbol{\omega})} \tilde{g}(\boldsymbol{\omega}) \geq 0 \tag{73}$$

This provides us with an efficient tool of associating regularization operators with kernel functions and analyzing the types of capacity control for some commonly used Support Vector kernels.

Example 18 (Kernels of Gaussians)
Following the exposition of [38] as described in [25] one can see that for

$$||\hat{P}f||^2 = \int d\mathbf{x} \sum_m \frac{\sigma^{2m}}{m! 2^m} (\hat{P}^m f(\mathbf{x}))^2 \tag{74}$$

with $\hat{P}^{2m} = \Delta^m$ *and* $\hat{P}^{2m+1} = \nabla \Delta^m$, Δ *being the Laplacian and* ∇ *the Gradient operator we get Gaussians kernels*

$$k(\mathbf{x}) = e^{-\frac{||\mathbf{x}||^2}{2\sigma^2}}. \tag{75}$$

Training a Support Vector Machine with Gaussian RBF kernels [28] corresponds to minimizing the specific cost function with a regularization operator of type (74). This also explains the good performance of Support Vector Machines in this case, as it is by no means obvious that choosing a flat function in high dimensional space will correspond to a simple function in low dimensional space, as we shall see in example 20.

Example 19 ($B_{(2n+1)}$-splines)
In [37] it has been proposed to use B_l*-splines as building blocks for kernels, i.e.*

$$k(\mathbf{x}) = \prod_{i=1}^{n} B_l(\mathbf{x}_i) \tag{76}$$

with $\mathbf{x} \in \mathbb{R}^n$. *Without loss of generality we will assume* $n = 1$. *Recalling the definition of* $B_l = \otimes^{k+1} 1_{[-0.5,0.5]}$ *(*$\otimes$ *denotes the convolution and* 1_X *the indicator function on* X*) we can construct the Fourier representation of the corresponding regularization operator:*

$$\tilde{k}(\boldsymbol{\omega}) = sinc^{(l+1)}(\frac{\boldsymbol{\omega}_i}{2}) \tag{77}$$

As showed above $P(\boldsymbol{\omega}) = \tilde{k}(\boldsymbol{\omega})$. Here we can see that only B-splines of odd order are admissible as the ones of even order have negative parts in the Fourier spectrum. The zeros in $\tilde{k}$ stem from the fact that B_l has only compact support $[-(k+1)/2, (k+1)/2]$. By using this kernel we trade reduced computational complexity in calculating f (we only have to take points with $||\mathbf{x}_i - \mathbf{x}_j|| \le c$ from some limited neighborhood determined by c into account) for a possibly worse performance of the regularization operator as it is not effective for frequencies $\boldsymbol{\omega}_p$ with $\tilde{k}(\boldsymbol{\omega}_p) = 0$.

Example 20 (Dirichlet kernels)
In [37], a class of kernels generating Fourier–like expansions was introduced. As in the previous example we assume $\mathbf{x} \in \mathbb{R}$ to avoid tedious notation.

$$k(x) = \frac{\sin(2N+1)x/2}{\sin x/2} \tag{78}$$

Clearly this corresponds to

$$\tilde{k}(\omega) = \frac{1}{2} \sum_{i=-N}^{N} \delta(\omega - i) \tag{79}$$

A regularization operator with these properties however may not be desirable as it only damps a finite number of frequencies and leaves all other frequencies unchanged. For the case of the training data not being spaced on a lattice with a spatial frequency corresponding to (78), this introduces frequency components that are not attenuated by the corresponding regularization operator, resulting in possibly poor performance. Even for the case of properly spaced training samples the effect of the regularization consists in attenuating all frequencies equally which is not very desirable either. Therefore in general the use of kernels of this type does not seem advisable.

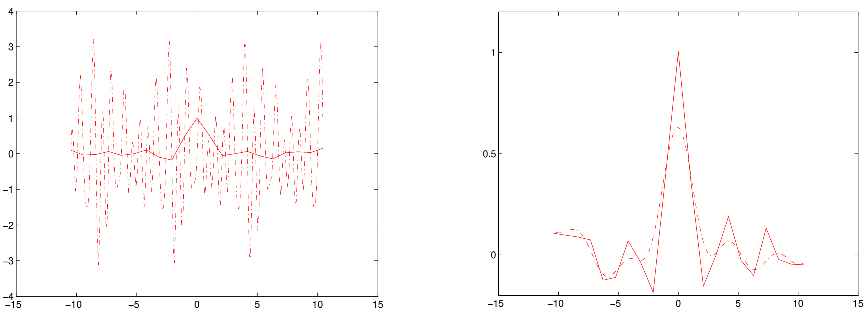

Figure 1: Interpolation with a Dirichlet Kernel (left) of order 10. One can clearly observe the overfitting (dashed line: interpolation, solid line: original data). Interpolation with a Gaussian Kernel (right) of width $\sigma^2 = 0.2$.

In [16] a class of functions is introduced which is suitable for radial basis expansions in regularization networks, namely the class of conditionally positive functions of any order. For convenience we will recall their definition (following [25]):

Definition 21 (Conditionally positive functions)
A continuous function $f(t)$, defined on $[0, \infty)$, is said to be conditionally (strictly) positive definite of order k on $\mathbb{R}^n$ if for any distinct points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^n$ and scalars $c_1, \ldots, c_n$ the quadratic form $\sum_{i,j=1}^{n} c_i c_j f(||\mathbf{x}_i - \mathbf{x}_j||)$ is (positive) nonnegative provided that $\sum_{i=1}^{n} c_i p(\mathbf{x}_i) = 0$ for all polynomials p on $\mathbb{R}^n$ of degree lower than k.

Theorem 22 (Completely monotonic functions and admissible kernels)
Every conditionally positive definite function h generates an admissible Hilbert Schmidt Kernel for Support Vector expansions.

$$k(\mathbf{x}_i, \mathbf{x}_j) = h(||\mathbf{x}_i - \mathbf{x}_j||^2) \tag{80}$$

Proof:
In [12] and [22] it was shown that conditionally positive definite functions h generate a semi–norm $||.||_h$ *by*

$$||f||_h := \int d\mathbf{x}_i d\mathbf{x}_j h(||\mathbf{x}_i - \mathbf{x}_j||_2^2) f(\mathbf{x}_i) f(\mathbf{x}_j). \tag{81}$$

This however is equivalent to fulfilling (32) which proves the theorem.

One should note that the null space of $||.||_h$ consists of the polynomials of degree lower than k for a conditionally positive definite function of order k. Consequently a polynomial of this order has to be added into the expansion of f — cf. (20). For $k > 1$, this leads to additional constraints in the quadratic programming problem. This can be seen by straightforward but tedious calculations which we do not present here. Moreover, kernels of this type require an additional number of parameters to be fixed for the polynomials (which may be a substantial problem for high dimensional data). Therefore in practise only functions of order 0 or 1 are of any practical relevance.

For convenience we will recall some of the functions described in [25] that can be used for Support Vector Machines. For a more detailed treatment see the original papers.

$$h(r) = e^{-\left(\frac{r}{c}\right)^2} \quad \text{(Gaussian)} \tag{82}$$

$$h(r) = \frac{1}{(c^2 + r^2)^\alpha} \quad (\alpha > 0) \tag{83}$$

$$h(r) = (c^2 + r^2)^\beta \quad (0 < \beta < 1) \tag{84}$$

$$h(r) = \sqrt{r^2 + c^2} \quad \text{(multiquadric)} \tag{85}$$

$$h(r) = r \quad \text{(linear)} \tag{86}$$

Another method of constructing regularization operators from given kernels exploits Mercer's theorem in a more straightforward manner. Using the expansion of k into eigenfunctions

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_l \Lambda_l \Phi_l(\mathbf{x}_i) \Phi_l(\mathbf{x}_j) \tag{87}$$

we arrive at:

Theorem 23 (Constructing $\hat{P}$ from k)
If for a regularization operator $\hat{P}$ *an expansion of* $\hat{P}^*\hat{P}$ *into a discrete eigensystem* (λ_l, ϕ_l) *exists, the eigensystem* (Λ_l, Φ_l) *of the corresponding Hilbert Schmidt kernel* $k(\mathbf{x}_i, \mathbf{x}_j)$ *has to satisfy*

$$\phi_l(\mathbf{x}) = \Phi_l(\mathbf{x}) \tag{88}$$

$$\lambda_l = \frac{1}{\Lambda_l} \tag{89}$$

Proof:
(λ_l, ϕ_l) being a basis, we can expand k in terms of this system

$$k(\mathbf{x}_i, \mathbf{x}) = \sum_l a_l(\mathbf{x}_i) \phi_l(\mathbf{x}) \tag{90}$$

using some coefficients $a_l \in \mathbb{R}$ depending on $\mathbf{x}_i$ only. Now we can substitute this expansion into (66) to get

$$\begin{aligned}\Big(k(\mathbf{x}_i, .) \cdot (\hat{P}^* \hat{P} k)(\mathbf{x}_j, .)\Big) &= \sum_l \lambda_l a_l(\mathbf{x}_i) a_l(\mathbf{x}_j) \qquad (91)\\ = \sum_l a_l(\mathbf{x}_i) \phi_l(\mathbf{x}_j) &= \sum_l \phi_l(\mathbf{x}_i) a_l(\mathbf{x}_j)\end{aligned}$$

This can only be true if and only if $\lambda_l a_l(\mathbf{x}) = \phi_l(\mathbf{x})$ (assuming appropriate indexing) which leads to

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_l \frac{1}{\lambda_l} \phi_l(\mathbf{x}_i) \phi_l(\mathbf{x}_j) \qquad (92)$$

and proves the theorem as (92) is an expansion in the sense of Mercer's theorem. Obviously this result also can be used for finding a regularization operator (or more precisely speaking the operator $\hat{P}^* \hat{P}$) associated with a given kernel.

Finally, connecting regularization operators and Support Vector machines leads to a Bayesian interpretation of the latter. In this case the choice of a special kernel can be regarded as a prior on the hypothesis space with

$$P[f] \propto \exp(-\lambda ||\hat{P} f||^2). \qquad (93)$$

8 Discussion

An important feature of Support Vector machines, sparsity of the decompositions of f, is due to a special type of cost functions used. The decisive part is the nonvanishing interval $[y_{ij} - \epsilon_{ij}^*, y_{ij} + \epsilon_{ij}]$ inside of which the cost for approximation, regression, or pattern recognition is zero. Therefore there exists a range of values $(\hat{A}_i f)(\mathbf{x}_{ij})$ in which (41) holds with $\eta_{ij}, \eta_{ij}^* = 0$ for some pairs (ij). 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, (42) implies

$$\begin{aligned}\beta_{ij}(y_{ij} + \epsilon_{ij} + \eta_{ij} - (\hat{A}_i f)(\mathbf{x}_{ij})) &= 0 \qquad (94)\\ \beta_{ij}^*((\hat{A}_i f)(\mathbf{x}_{ij}) - y_{ij} + \epsilon_{ij}^* + \eta_{ij}) &= 0. \qquad (95)\end{aligned}$$

Therefore, the β_{ij} and β_{ij}^* have to vanish for the constraints of (42) that are met. This causes sparsity in the expansion of β_{ij} and β_{ij}^*.

As shown in the Examples 5 and 8, 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 [31] for details).

Unfortunately, independent of the noise level, the choice of a different regularization prevents such an efficient calculation scheme due to equation (45), 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 *encoding* purposes as f is defined uniquely by the set of β_{ij}. Moreover as shown in section 7 for the case that the explicit type of the expansion does not matter one may construct a kernel k corresponding to a given regularization operator $\hat{P}$ in order to achieve sparse decompositions.

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 (32) the reduced set method [5] 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 β_{ij}, as we have to solve a quadratic programming problem whose quadratic part is essentially as large as the number of basis functions of the functional expansion of f. Optimization methods like the Bunch-Kaufman decomposition ([4], [9]) 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 [34] 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 (42) and (96), 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 [26] and solve the optimization equations in the reduced space.

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 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 (22) and (42) we arrive at the following statement of the optimization problem: Solve

$$\begin{array}{ll} \text{minimize} & \frac{1}{\lambda} \sum\limits_{i=1}^{q} \sum\limits_{j=1}^{l_i} (c_{ij}(\eta_{ij}) + c_{ij}^*(\eta_{ij}^*)) + \frac{1}{2} \sum\limits_{i=1}^{q} \sum\limits_{j=1}^{l_i} \sum\limits_{m=1}^{q} \sum\limits_{n=1}^{l_m} \alpha_{ij}\alpha_{mn} D_{(ij)(mn)} \\ \text{subject to} & (\hat{A}_i f)(\mathbf{x}_{ij}) \leq y_{ij} + \epsilon_{ij} + \eta_{ij} \\ & (\hat{A}_i f)(\mathbf{x}_{ij}) \geq y_{ij} - \epsilon_{ij}^* - \eta_{ij}^* \\ & \eta_{ij}, \eta_{ij}^* \geq 0 \quad \text{for all} \quad 1 \leq i \leq q, 1 \leq j \leq l_i \end{array} \tag{96}$$

To this end, we introduce a Lagrangian

$$\begin{aligned}
L \quad = \quad & \tfrac{1}{\lambda}\sum_{i=1}^{q}\sum_{j=1}^{l_i}(c_{ij}(\eta_{ij})+c_{ij}^*(\eta_{ij}^*))+ \\
& \tfrac{1}{2}\sum_{i=1}^{q}\sum_{j=1}^{l_i}\sum_{m=1}^{q}\sum_{n=1}^{l_m}\alpha_{ij}\alpha_{mn}D_{(mn)(ij)}- \\
& \sum_{i=1}^{q}\sum_{j=1}^{l_i}\beta_{ij}(y_{ij}+\epsilon_{ij}+\eta_{ij}-(\hat{A}_i f)(\mathbf{x}_{ij}))- \\
& \sum_{i=1}^{q}\sum_{j=1}^{l_i}\beta_{ij}^*((\hat{A}_i f)(\mathbf{x}_{ij})-y_{ij}+\epsilon_{ij}^*+\eta_{ij}^*)- \\
& \sum_{i=1}^{q}\sum_{j=1}^{l_i}(\eta_{ij}\xi_{ij}+\eta_{ij}^*\xi_{ij}^*) \\
\text{with} \quad & \beta_{ij},\beta_{ij}^*,\xi_{ij},\xi_{ij}^* \geq 0.
\end{aligned} \tag{97}$$

In (97), the regularization term is expressed in terms of the function expansion coefficients α_{ij}. We next do the same for the terms stemming from the constraints on $(\hat{A}_i f)(\mathbf{x}_{ij})$, and compute $\hat{A}_i f$ by substituting the expansion (20) to get

$$\begin{aligned}
(\hat{A}_i f)(\mathbf{x}_{ij}) &= \sum_{m=1}^{q}\sum_{n=1}^{l_m}\alpha_{mn}((\hat{A}_m\otimes\hat{A}_i)k)(\mathbf{x}_{mn},\mathbf{x}_{ij})+\hat{A}_i b \\
&= \sum_{m=1}^{q}\sum_{n=1}^{l_m}\alpha_{mn}K_{(mn)(ij)}+\hat{A}_i b.
\end{aligned} \tag{98}$$

Now we can compute the derivatives with respect to the primal variables $\alpha_{ij}, b, \eta_{ij}$. These have to vanish for optimality.

$$\begin{aligned}
\tfrac{\partial}{\partial\alpha_{mn}}L &= \sum_{i=1}^{q}\sum_{j=1}^{l_i}(\alpha_{ij}D_{(mn)(ij)}-(\beta_{ij}-\beta_{ij}^*)K_{(mn)(ij)}) \\
&= 0
\end{aligned} \tag{99}$$

Solving (99) for $\vec{\alpha}$ and rewriting the equations in vector notation yields

$$\vec{\alpha} = D^{-1}K(\vec{\beta}-\vec{\beta}^*), \tag{100}$$

where D^{-1} is the pseudoinverse for the case that D does not have full rank.

We proceed to the next Lagrange condition, reading

$$\begin{aligned}
\tfrac{1}{b}\cdot\tfrac{\partial}{\partial b}L &= \sum_{i=1}^{q}(\hat{A}_i 1)\sum_{j=1}^{l_i}(\beta_{ij}^*-\beta_{ij}) \\
&= 0,
\end{aligned} \tag{101}$$

using $\hat{A}_i b = b\hat{A}_i 1$. Some summands may vanish for $(\hat{A}_i 1)=0$. Partial differentiation with respect to η_{ij} and η_{ij}^* yields

$$\frac{1}{\lambda}\frac{d}{d\eta_{ij}}c_{ij}(\eta_{ij}) = \beta_{ij}+\xi_{ij} \tag{102}$$

$$\frac{1}{\lambda}\frac{d}{d\eta_{ij}^*}c_{ij}^*(\eta_{ij}^*) = \beta_{ij}^*+\xi_{ij}^* \tag{103}$$

Now we may substitute (100), (101), (103), and (103) back into (97), taking into account the substitution (98), and eliminate α_{ij} and ξ_{ij}, obtaining

$$
\begin{aligned}
L \;=\; & \tfrac{1}{\lambda}\sum_{i=1}^{q}\sum_{j=1}^{l_i}(c_{ij}(\eta_{ij}) - \eta_{ij}\tfrac{d}{d\eta_{ij}}c_{ij}(\eta_{ij}) + c^*_{ij}(\eta^*_{ij}) - \eta^*_{ij}\tfrac{d}{d\eta^*_{ij}}c^*_{ij}(\eta^*_{ij})) - \\
& \sum_{i=1}^{q}\sum_{j=1}^{l_i}(\beta_{ij}(\epsilon_{ij}+y_{ij}) + \beta^*_{ij}(\epsilon^*_{ij}-y_{ij})) - \\
& \tfrac{1}{2}\sum_{i=1}^{q}\sum_{j=1}^{l_i}\sum_{m=1}^{q}\sum_{n=1}^{l_m}(\beta_{ij}-\beta^*_{ij})(\beta_{mn}-\beta^*_{mn})(KD^{-1}K)_{(ij)(mn)}
\end{aligned}
\tag{104}
$$

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

$$T(\eta) := \frac{1}{\lambda}(c(\eta) - \eta\frac{d}{d\eta}c(\eta)) \tag{105}$$

and

$$\frac{1}{\lambda}\frac{d}{d\eta}c(\eta) = \beta + \xi, \tag{106}$$

where the index (ij) and possible asterisks have been omitted for clarity.

Example 24 (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 [7] 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 25. We use

$$c(\eta) = \frac{1}{p}\eta^p. \tag{107}$$

From Eqs. (105), (106), and (107) it follows that

$$\frac{1}{\lambda}\eta^{p-1} = \beta + \xi \tag{108}$$

$$
\begin{aligned}
T(\eta) &= \frac{1}{\lambda}\left(\frac{1}{p}\eta^p - \eta\eta^{p-1}\right) \\
&= -\left(1-\frac{1}{p}\right)\lambda^{\frac{1}{p-1}}(\beta+\xi)^{\frac{p}{p-1}}
\end{aligned}
\tag{109}
$$

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

$$T(\beta) = -\left(1-\frac{1}{p}\right)\lambda^{\frac{1}{p-1}}\beta^{\frac{p}{p-1}} \quad \textit{with} \quad \beta \in \mathbb{R}_0^+. \tag{110}$$

Moreover we have the following relation between β and η:

$$\eta = (\lambda\beta)^{\frac{1}{p-1}} \tag{111}$$

Example 25 (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.

$$c(\eta) = \begin{cases} \sigma^{1-p}\frac{1}{p}\,\eta^p & \text{for } \eta < \sigma \\ \eta + \left(\frac{1}{p}-1\right)\sigma & \text{for } \eta \geq \sigma \end{cases} \tag{112}$$

$$T(\eta) = \frac{1}{\lambda}\begin{cases} -\sigma^{1-p}\left(1-\frac{1}{p}\right)\eta^p & \text{for } \eta < \sigma \\ -\sigma\left(1-\frac{1}{p}\right) & \text{for } \eta \geq \sigma \end{cases} \tag{113}$$

$$\beta + \xi = \frac{1}{\lambda}\begin{cases} \sigma^{1-p}\,\eta^{p-1} & \text{for } \eta < \sigma \\ 1 & \text{for } \eta \geq \sigma \end{cases} \tag{114}$$

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 (113).

$$T(\beta) = -\lambda^{\frac{1}{p-1}}\left(1-\frac{1}{p}\right)\sigma\beta^{\frac{p}{p-1}} \quad \text{for} \quad \sigma \in \mathbb{R}_0^+ \tag{115}$$

Analogously to Example 24 we can determine the error for $\beta \in [0, \frac{1}{\lambda})$ by

$$\eta = \sigma(\lambda\beta)^{\frac{1}{p-1}} \tag{116}$$

Example 26 (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:

$$\begin{array}{ll} \textit{maximize} & \sum_{i=1}^{q}\sum_{j=1}^{l_i}(\beta_{ij}^*(y_{ij}-\epsilon_{ij}) - \beta_{ij}(y_{ij}+\epsilon_{ij})) - \\ & \frac{1}{2}\sum_{i=1}^{q}\sum_{j=1}^{l_i}\sum_{m=1}^{q}\sum_{n=1}^{l_m}(\beta_{ij}-\beta_{ij}^*)(\beta_{mn}-\beta_{mn}^*)(KD^{-1}K)_{(ij)(mn)} \\ \textit{subject to} & \sum_{i=1}^{q}(\hat{A}_i 1)\sum_{j=1}^{l_i}(\beta_{ij}^*-\beta_{ij}) = 0 \end{array} \tag{117}$$

Another way to to see this is to use the result of Example 8 and take take the limit $\lambda \to 0$. Loosely speaking, the interval $[0, \frac{1}{\lambda}]$ then converges to $\mathbb{R}_0^+$.

B Proof of Theorem 12

We modify the proof given in [6] to deal with the more general case stated in Theorem 12. 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

$$\begin{aligned} & \frac{\delta}{\delta f}R_{reg}[f,\psi] \\ = \; & \lim_{k\to 0}\frac{R_{reg}[f+k\psi] - R_{reg}[f]}{k} \end{aligned} \tag{118}$$

$$= \lim_{k\to 0} \frac{1}{k} \left[\sum_{i=1}^{q} \frac{1}{l_i} \sum_{j=1}^{l_i} c_i((\hat{A}_i(f+k\psi))(\mathbf{x}_{ij}), \mathbf{x}_{ij}, y_{ij}) \right.$$

$$\left. - \sum_{i=1}^{q} \frac{1}{l_i} \sum_{j=1}^{l_i} c_i((\hat{A}_i f)(\mathbf{x}_{ij}), \mathbf{x}_{ij}, y_{ij}) + \frac{\lambda}{2} \left(||\hat{P}(f+k\psi)||^2_{\mathcal{D}} - ||\hat{P}f||^2_{\mathcal{D}} \right) \right]$$

Expanding (118) in terms of k and taking the limit $k \to 0$ yields

$$\frac{\delta}{\delta f} R_{reg}[f,\psi] = \sum_{i=1}^{q} \frac{1}{l_i} \sum_{j=1}^{l_i} \partial_1 c_i((\hat{A}_i f)(\mathbf{x}_{ij}), \mathbf{x}_{ij}, y_{ij})(\hat{A}_i\psi)(\mathbf{x}_{ij}) + \lambda(\hat{P}f \cdot \hat{P}\psi)_{\mathcal{D}}. \quad (119)$$

Equation (119) has to vanish for $f = f_{opt}$. As $\bar{\mathcal{D}}$ is a Hilbert space, we can define the adjoint $\hat{P}^*$ and get

$$(\hat{P}f \cdot \hat{P}\psi)_{\mathcal{D}} = (\hat{P}^*\hat{P}f \cdot \psi)_{\mathcal{H}(V)}. \quad (120)$$

Similarly, we rewrite the first term of (119) to get

$$\sum_{i=1}^{q} \frac{1}{l_i} \sum_{j=1}^{l_i} \partial_1 c_i((\hat{A}_i f)(\mathbf{x}_{ij}), \mathbf{x}_{ij}, y_{ij}) \left(\delta_{\mathbf{x}_{ij}} \cdot \hat{A}_i\psi \right)_{\mathcal{H}(V)} + \lambda \left(\hat{P}^*\hat{P}f \cdot \psi \right)_{\mathcal{H}(V)} \quad (121)$$

Using $\left(\delta_{\mathbf{x}_{ij}} \cdot \hat{A}_i\psi \right)_{\mathcal{H}(V)} = \left(\hat{A}_i^* \delta_{\mathbf{x}_{ij}} \cdot \psi \right)_{\mathcal{H}(V)}$, the whole expression (121) can be written as a dot product with ψ. As ψ was arbitrary, this proves the theorem.

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

$$\alpha_{ij} \in \partial_1 c_i((\hat{A}_i f)(\mathbf{x}_{ij}), \mathbf{x}_{ij}, y_{ij}). \quad (122)$$

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.

References

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

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

[3] 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.

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

[5] C. J. C. Burges. Simplified support vector decision rules. In *13th International Conference on Machine Learning*, pages 71–77, 1996.

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

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

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

[9] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Linear support vector regression machines. In *Advances in Neural Information Processing Systems 9*, 1996. forthcoming.

[10] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, New York, 1974.

[11] 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.

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

[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*, 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] 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, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

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

[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. forthcoming.

[22] W.R. Madych and S.A. Nelson. Multivariate interpolation and conditionally positive definite functions. II. *Mathematics of Computation*, 54(189):211–230, 1990.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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. Technical Report 1599, Massachusetts Institute of Techology, 1996.

[29] 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.

[30] 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*, San Mateo, CA, 1992. Morgan Kaufmann.

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

[32] E. M. Stein. *Singular Integrals and Differentiability Properties of Functions.* Princeton University Press, Princeton, N.J., 1970.

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

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

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

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

[37] V. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In *Advances in Neural Information Processing Systems 9*, San Mateo, CA, 1996. (in press).

[38] A. Yuille and N. Grzywacz. The motion coherence theory. In *Proceedings of the International Conference on Computer Vision*, pages 344–354, Washington, D.C., 1988. IEEE Computer Society Press.