A Generalized Representer Theorem

Bernhard Schölkopf [1] Ralf Herbrich [2] Alex J. Smola [3]
Robert Williamson [4]

NeuroCOLT2 Technical Report Series

NC2-TR-2000-81

May, 2000

Produced as part of the ESPRIT Working Group
in Neural and Computational Learning II,
NeuroCOLT2 27150

For more information see the NeuroCOLT website
http://www.neurocolt.com
or email neurocolt@neurocolt.com

[1] bsc@scientist.com, Microsoft Research Ltd., 1 Guildhall Street, Cambridge, UK
[2] ralfh@cs.tu-berlin.de, Department of Computer Science, Technische Universität, Franklinstrasse 28/29, 10587 Berlin, Germany
[3] Alex.Smola@anu.edu.au, Australian National University, Department of Engineering, Canberra ACT 0200, Australia
[4] Robert.Williamson@anu.edu.au, Australian National University, Department of Engineering, Canberra ACT 0200, Australia

Abstract

Wahba's classical representer theorem states that the solutions of certain risk minimization problems involving an empirical risk term and a quadratic regularizer can be written as expansions in terms of the training examples. We generalize the theorem to a larger class of regularizers and empirical risk terms, and give a self-contained proof utilizing the feature space associated with a support vector kernel. The result shows that a wide range of problems have optimal solutions that live in the finite dimensional span of the training examples mapped into feature space, thus enabling us to carry out kernel algorithms independent of the (potentially infinite) dimensionality of the feature space.

1 Introduction

Following the development of support vector (SV) machines [12], positive definite kernels have recently attracted considerable attention in the machine learning community. It turns out that a number of results that have now become popular were already known in the approximation theory community, as witnessed by the work of Wahba [13]. The present work brings together tools from both areas to formulate a generalized version of a classical theorem from the latter field, and to prove it using the geometrical view of kernel function classes as corresponding to vectors in linear feature spaces.

The article is organized as follows. In the present first section, we review some basic concepts. The two subsequent sections contain our main result, some examples and a short discussion.

1.1 Positive Definite Kernels

The question of under which conditions kernels correspond to dot products in linear spaces has been brought to the attention of the machine learning community by [1, 4, 12]. In functional analysis, the same problem has been studied under the heading of *Hilbert space representations* of kernels. A good monograph on the functional analytic theory of kernels is [3]. Most of the material in the present introductory section is taken from that work. Readers familiar with it can skip over the remainder of it.

Suppose we are given empirical data

$$(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathbb{R}. \tag{1}$$

Here, the *domain* $\mathcal{X}$ is some nonempty set that the patterns x_i are taken from; the target values y_i live in $\mathbb{R}$. Note that we have not made any assumptions concerning the domain $\mathcal{X}$ other than it being a set. In order to study the problem of learning, we need additional structure. In kernel methods, this is provided by a similarity measure

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \quad (x, x') \mapsto k(x, x'). \tag{2}$$

The function k is called a *kernel* [9]. The term stems from the first use of this type of function in the study of integral operators, where a function k giving rise to an operator T_k via

$$(T_k f)(x) = \int_{\mathcal{X}} k(x, x') f(x') \, dx' \tag{3}$$

is called the kernel of T_k.

Below, unless stated otherwise, indices i and j will be understood to run over the training set, i.e. $i, j = 1, \ldots, m$. Note that we will state most results for the more general case of complex-valued kernels; they specialize to the real-valued case in a straightforward manner.

Definition 1 (Gram matrix) *Given a kernel k and patterns $x_1, \ldots, x_m \in \mathcal{X}$, the $m \times m$ matrix*

$$K := (k(x_i, x_j))_{ij} \tag{4}$$

is called the Gram matrix *of k with respect to $x_1, \ldots, x_m$.*

Below, we use the notation $\bar{c}_j$ to denote the complex conjugate of c_j.

Definition 2 (Positive matrix) *An $m \times m$ matrix K over $\mathbb{C}$ satisfying*

$$\sum_{i,j} c_i \bar{c}_j K_{ij} \geq 0 \tag{5}$$

for all $c_1, \ldots, c_m \in \mathbb{C}$ is called positive.

Definition 3 (Positive definite kernel) *Let $\mathcal{X}$ be a nonempty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ which for all $m \in \mathbb{N}, x_i \in \mathcal{X}$ gives rise to a positive Gram matrix is called a* positive definite (pd) kernel.

One might argue that the term *positive definite kernel* is slightly misleading. In matrix theory, the term *definite* is sometimes used to denote the case where equality in (5) only occurs if $c_1 = \ldots = c_m = 0$. Simply using the term *positive kernel*, on the other hand, could be confused with a kernel whose *values* are positive.

Real-valued kernels are contained in the above definition as a special case. However, it is not sufficient to require that (5) hold for real coefficients c_i. If we want to get away with real coefficients only, we additionally have to require that the kernel be symmetric. The complex case is slightly more elegant; in that case, (5) can be shown to imply symmetry, i.e. $k(x_i, x_j) = \overline{k(x_j, x_i)}$.

Positive definite kernels can be regarded as generalized dot products. Indeed, any dot product is a pd kernel; however, linearity does not carry over from dot products to general pd kernels. Another property of dot products, the Cauchy-Schwarz inequality, does have a natural generalization: if k is a positive definite kernel, and $x_1, x_2 \in \mathcal{X}$, then

$$|k(x_1, x_2)|^2 \leq k(x_1, x_1) \cdot k(x_2, x_2). \tag{6}$$

1.2 ... and Associated Feature Spaces

We define a map from $\mathcal{X}$ into the space of functions mapping $\mathcal{X}$ into $\mathbb{C}$, denoted as $\mathbb{C}^{\mathcal{X}}$, via [3]

$$\phi : \mathcal{X} \to \mathbb{C}^{\mathcal{X}}, \quad x \mapsto k(., x). \tag{7}$$

Here, $\phi(x) = k(., x)$ denotes the function that assigns the value $k(x', x)$ to $x' \in \mathcal{X}$. Applying ϕ to x amounts to representing it by its similarity to *all* other points in the input domain $\mathcal{X}$. This seems a very rich representation, but it will turn out that the kernel allows the computation of a dot product in that representation.

We shall now construct a dot product space containing the images of the input patterns under ϕ. To this end, we first need to endow it with the linear structure of a vector space. This is done by forming linear combinations of the form

$$f(.) = \sum_{i=1}^{m} \alpha_i k(., x_i), \quad g(.) = \sum_{j=1}^{m'} \beta_j k(., x'_j). \tag{8}$$

Here, $m, m' \in \mathbb{N}$, $\alpha_i, \beta_j \in \mathbb{C}$ and $x_i, x'_j \in \mathcal{X}$ are arbitrary. A dot product between f and g can be constructed as

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \bar{\alpha}_i \beta_j k(x_i, x'_j). \tag{9}$$

To see that this is well-defined, although it explicitly contains the expansion coefficients (which need not be unique), note that

$$\langle f, g \rangle = \sum_{j=1}^{m'} \beta_j \overline{f(x'_j)}, \tag{10}$$

using $k(x'_j, x_i) = \overline{k(x_i, x'_j)}$. The latter, however, does not depend on the particular expansion of f. Similarly, for g, note that

$$\langle f, g \rangle = \sum_{i} \bar{\alpha}_i g(x_i). \tag{11}$$

This also shows that $\langle ., . \rangle$ is antilinear in the first argument and linear in the second one. It is symmetric, since $\langle f, g \rangle = \overline{\langle g, f \rangle}$. Moreover, given functions $f_1, \ldots, f_n$, and coefficients $\gamma_1, \ldots, \gamma_n \in \mathbb{C}$, we have

$$\sum_{i,j=1}^{n} \bar{\gamma}_i \gamma_j \langle f_i, f_j \rangle = \left\langle \sum_{i=1}^{n} \gamma_i f_i, \sum_{j=1}^{n} \gamma_j f_j \right\rangle \geq 0, \tag{12}$$

hence $\langle ., . \rangle$ is actually a pd kernel on our function space.

For the last step in proving that it even is a dot product, we will use the following interesting property of ϕ, which follows directly from the definition: for all functions (8), we have

$$\langle k(., x), f \rangle = f(x), \tag{13}$$

i.e., k is the *representer of evaluation.* In particular,

$$\langle k(., x), k(., x') \rangle = k(x, x'), \tag{14}$$

hence (cf. (7)) $k(x, x') = \langle \phi(x), \phi(x') \rangle$. By virtue of these properties, pd kernels k are also called *reproducing kernels* [2, 3, 13]. By (13) and (6), we have

$$|f(x)|^2 = |\langle k(., x), f \rangle|^2 \leq k(x, x) \cdot \langle f, f \rangle. \tag{15}$$

Therefore, $\langle f, f \rangle = 0$ implies $f = 0$, which is the last property that was left to prove in order to establish that $\langle ., . \rangle$ is a dot product.

One can complete the space of functions (8) in the norm corresponding to the dot product, i.e. add the limit points of sequences that are convergent in that norm, and thus gets a Hilbert space H_k, usually called a *reproducing kernel Hilbert space (RKHS).* The case of real-valued kernels is included in the above; in that case, H_k can be chosen as a real Hilbert space.

2 The Representer Theorem

We now state our main result, showing that a large class of optimization problems with RKHS regularizers have solutions that can be expressed as kernel expansions in terms of the training data.

Theorem 4 (Nonparametric Representer Theorem) *Suppose we are given a nonempty set $\mathcal{X}$, a positive definite real-valued kernel k on $\mathcal{X} \times \mathcal{X}$, a training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathbb{R}$, a strictly monotonically increasing real-valued function g on $[0, \infty[$, an arbitrary cost function $c : (\mathcal{X} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{\infty\}$, and a class of functions*

$$\mathcal{F} = \left\{ f \in \mathbb{R}^{\mathcal{X}} \,\middle|\, f(.) = \sum_{i=1}^{\infty} \beta_i k(., z_i), \beta_i \in \mathbb{R}, z_i \in \mathcal{X}, \|f\| < \infty \right\}. \tag{16}$$

Here, $\|.\|$ is the norm in the RKHS H_k associated with k, i.e. for any $z_i \in \mathcal{X}, \beta_i \in \mathbb{R}$ $(i \in \mathbb{N})$,

$$\left\| \sum_{i=1}^{\infty} \beta_i k(., z_i) \right\|^2 = \sum_{i,j=1}^{\infty} \beta_i \beta_j k(z_i, z_j). \tag{17}$$

Then any $f \in \mathcal{F}$ minimizing the regularized risk functional

$$c\left((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))\right) + g\left(\|f\|\right) \tag{18}$$

admits a representation of the form

$$f(.) = \sum_{i=1}^{m} \alpha_i k(., x_i). \tag{19}$$

Let us give a few remarks before the proof. In its original form, with mean squared loss

$$c((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))) = \frac{1}{m} \sum_{i=1}^{m} (y_i - f(x_i))^2, \tag{20}$$

or hard constraints on the outputs, and $g(\|f\|) = \lambda \|f\|^2$ $(\lambda > 0)$, the theorem is due to [8]. Note that in our formulation, hard constraints on the solution are included by the possibility of c taking the value ∞. A generalization to non-quadratic cost functions was stated by [5], cf. the discussion in [14]. The present generalization to $g(\|f\|)$ is, to our knowledge, new.[1] For a machine learning point of view on the representer theorem, and a variational proof, cf. [6].

The significance of the theorem is that it shows that *a whole range of learning algorithms have optimal solutions that can be expressed as expansions in terms of the training examples.*

Note that monotonicity of g is necessary to ensure that the theorem holds. It does not ensure that the regularized risk functional (18) does not have multiple local minima. For this, we would need to require convexity of g and of the cost function c.

[1] thanks to Grace Wahba for her knowledgeable comments

If we discarded the strictness of the monotonicity of g, it would no longer follow that each minimizer (there might be multiple ones) of the regularized risk admits an expansion (19); however, it would still follow that there is always another solution minimizing (18) that *does* admit the expansion.

In the SV community, (19) is called the *SV expansion.* For suitable choices of cost functions, it has empirically been found that often many of the α_i equal 0.

Proof. As we have assumed that k maps into $\mathbb{R}$, we will use (cf. (7))

$$\phi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}, \quad x \mapsto k(.,x). \tag{21}$$

Since k is a reproducing kernel, evaluation of the function $\phi(x)$ on the point x' yields

$$\phi(x)(x') = k(x',x) = \langle \phi(x'), \phi(x) \rangle \tag{22}$$

for all $x, x' \in \mathcal{X}$. Here, $\langle .,. \rangle$ denotes the dot product of H_k.

Given $x_1, \ldots, x_m$, any $f \in \mathcal{F}$ can be decomposed into a part that lives in the span of the $\phi(x_i)$ and a part which is orthogonal to it, i.e.

$$f = \sum_i \alpha_i \phi(x_i) + v \tag{23}$$

for some $\alpha \in \mathbb{R}^m$ and $v \in \mathcal{F}$ satisfying, for all j,

$$\langle v, \phi(x_j) \rangle = 0. \tag{24}$$

Using the latter and (22), application of f to an arbitrary training point x_j yields

$$\begin{aligned} f(x_j) &= \left\langle \sum_i \alpha_i \phi(x_i) + v, \phi(x_j) \right\rangle \\ &= \sum_i \alpha_i \langle \phi(x_i), \phi(x_j) \rangle, \end{aligned} \tag{25}$$

independent of v. Consequently, the first term of (18) is independent of v.

As for the second term, since v is orthogonal to $\sum_i \alpha_i \phi(x_i)$, and g is strictly monotonic, we get

$$\begin{aligned} g(\|f\|) &= g\left(\left\| \sum_i \alpha_i \phi(x_i) + v \right\| \right) \\ &= g\left(\sqrt{ \left\| \sum_i \alpha_i \phi(x_i) \right\|^2 + \|v\|^2 } \right) \geq g\left(\left\| \sum_i \alpha_i \phi(x_i) \right\| \right), \end{aligned} \tag{26}$$

with equality occuring if and only if $v = 0$.

Setting $v = 0$ thus does not affect the first term of (18), while strictly reducing the second term — hence, any minimizer must have $v = 0$. Consequently, any solution takes the form $f = \sum_i \alpha_i \phi(x_i)$, i.e., using (22),

$$f(.) = \sum_i \alpha_i k(.,x_i). \tag{27}$$

■

The extension to the case where we also include a parametric part is straightforward, cf. [8]. We thus state the corresponding result without proof:

Theorem 5 (Semiparametric Representer Theorem) *Suppose that in addition to the assumptions of the previous theorem we are given a set of M real-valued functions $\{\psi_p\}_{p=1}^M$ on $\mathcal{X}$, with the property that the $m \times M$ matrix $(\psi_p(x_i))_{ip}$ has rank M. Then any $\tilde{f} := f + h$, with $f \in \mathcal{F}$ and $h \in \operatorname{span}\{\psi_p\}$, minimizing the regularized risk*

$$c\left((x_1, y_1, \tilde{f}(x_1)), \ldots, (x_m, y_m, \tilde{f}(x_m))\right) + g\left(\|f\|\right) \tag{28}$$

admits a representation of the form

$$\tilde{f}(.) = \sum_{i=1}^{m} \alpha_i k(x_i, .) + \sum_{p=1}^{M} \beta_p \psi_p(.), \tag{29}$$

with $\beta_p \in \mathbb{R}$ for all $p = 1, \ldots, M$.

Remark 6 (Biased regularization) *A straightforward extension of the representer theorems can be obtained by including a term $-\langle f_0, f\rangle$ into (18) or (28), respectively, where $f_0 \in H_k$. In this case, if a solution to the minimization problem exists, it admits an expansion which differs from the above ones in that it additionally contains a multiple of f_0. To see this, decompose the vector v used in the proof of Theorem 4 into a part orthogonal to f_0 and the remainder.*

In the case where $g(\|f\|) = \frac{1}{2}\|f\|^2$, this can be seen to correspond to an effective overall regularizer of the form $\frac{1}{2}\|f - f_0\|^2$.

Some explicit applications of theorems 4 and 5 are given below.

Example 7 (SV Regression) *For SV regression with the ε–insensitive loss [12] we have*

$$c\left((x_i, y_i, f(x_i))_{i=1,\ldots,m}\right) = \frac{1}{\lambda}\sum_i \max\left(0, |f(x_i) - y_i| - \varepsilon\right) \tag{30}$$

and $g(\|f\|) = \|f\|^2$, where $\lambda > 0$ and $\varepsilon \geq 0$ are fixed parameters which determine the trade-off between regularization and fit to the training set. In addition, a single ($M = 1$) constant function $\psi_1(x) = b$ ($b \in \mathbb{R}$) is used as an offset that is not regularized by the algorithm [14].

In [11], a semiparametric extension was proposed which shows how to deal with the case $M > 1$ algorithmically. Theorem 5 applies in that case, too.

Example 8 (SV Classification) *Here, the targets satisfy $y_i \in \{\pm 1\}$, and we use*

$$c\left((x_i, y_i, f(x_i))_{i=1,\ldots,m}\right) = \frac{1}{\lambda}\sum_i \max\left(0, 1 - y_i f(x_i)\right), \tag{31}$$

the regularizer $g(\|f\|) = \|f\|^2$*, and* $\psi_1(x) = b$*. For* $\lambda \to 0$*, we recover the hard margin SVM, i.e. the minimizer must correctly classify each training point* (x_i, y_i)*. Note that after training, the actual classifier will be* $\operatorname{sgn}(f(.) + b)$*.*

Example 9 (Bayesian MAP Estimates) *The correspondence to Bayesian methods is established by identifying (18) with the negative log posterior [7, 6]. In this case,* $\exp(-c((x_i, y_i, f(x_i))_{i=1,\dots,m}))$ *is the likelihood of the data, while* $\exp(-g(\|f\|))$ *is the prior over the set of functions. The well-known Gaussian process prior (e.g. [13, 15]), with covariance function k, is obtained by using* $g(\|f\|) = \lambda\|f\|^2$ *(here,* $\lambda > 0$*, and, as above,* $\|.\|$ *is the norm of the RKHS associated with k). A Laplacian prior would be obtained by using* $g(\|f\|) = \lambda\|f\|$*. In all cases, the minimizer of (18) corresponds to a function with maximal a posteriori probability (MAP).*

Example 10 (Kernel PCA) *PCA in a kernel feature space can be shown to correspond to the case of*

$$c((x_i, y_i, f(x_i))_{i=1,\dots,m}) = \begin{cases} 0 & \text{if } \frac{1}{m}\sum_i \left(f(x_i) - \frac{1}{m}\sum_j f(x_j)\right)^2 = 1 \\ \infty & \text{otherwise} \end{cases} \qquad (32)$$

with g an arbitrary strictly monotonically increasing function [10]. The constraint ensures that we are only considering linear feature extraction functionals that produce outputs of unit empirical variance. Note that in this case of unsupervised learning, there are no labels y_i *to consider.*

3 Conclusion

We have shown that for a large class of algorithms minimizing a sum of an empirical risk term and a regularization term in a reproducing kernel Hilbert space, the optimal solutions can be written as kernel expansions in terms of training examples. This has been known for specific algorithms; e.g. for the SV algorithm, where it is a direct consequence of the structure of the optimization problem. The representer theorem puts these individual findings into a wider perspective [14], and it is our hope that the reader will find our present generalization useful by either gaining some insight, or by taking it as a practical guideline for designing innovative kernel algorithms: as long as the objective function can be cast into the form considered in the generalized representer theorem, one can recklessly carry out algorithms in infinite dimensional spaces, since the solution will always live in a specific subspace whose dimensionality equals at most the number of training examples.

Acknowledgements. Thanks to Grace Wahba, Jonathan Baxter and Peter Bartlett for useful comments. This work was supported by the Australian Research Council. AS was supported by DFG grant SM 62/1-1. Parts of this work were done while BS and RH were visiting the ANU.

References

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

[2] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[3] C. Berg, J.P.R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag, New York, 1984.

[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, July 1992. ACM Press.

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

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

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

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

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

[10] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 327–352. MIT Press, Cambridge, MA, 1999.

[11] A. Smola, T. Frieß, and B. Schölkopf. Semiparametric support vector and linear programming machines. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 585 – 591, Cambridge, MA, 1999. MIT Press.

[12] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.

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

[14] G. Wahba. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 69–88, Cambridge, MA, 1999. MIT Press.

[15] C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning and Inference in Graphical Models*. Kluwer, 1998.