ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 185-196

Reproducing kernels and regularization methods in machine learning

Massimiliano Pontil[*]

Abstract. After a brief introduction to learning theory, we review the elements of reproducing kernel Hilbert spaces and discuss learning algorithms which work thereby. In particular, we focus on regularization-based algorithms, among which important examples are regularization networks and support vector machines.

1. Introduction

Over the past ten years learning theory has undergone a significant progress in the development of learning algorithms and in their theoretical foundation. The theory builds on concepts which combine ideas from probability and statistics, and functional analysis. The formers are the natural tools to study the performance of a learning algorithm. This has been formalized by the work of Vapnik and Chervonenkis which we briefly touch in Section 2. The latter provides us with families of function spaces where a learning algorithm comes to play. In this paper we focus on a general class of function spaces, called reproducing kernel Hilbert spaces (RKHS) [3]. Section 3 presents a self-contained introduction to this subject.

Among the recent learning methods, an increasing number makes use of RKHS. The best case is provided by the widespread support vector machine (SVM) [24], a state-of-the art technique in Machine Learning. SVM as well as the older regularization networks [20] are reviewed in Section 4 within the framework of regularization-based algorithms.

2. The learning problem

The central theme of learning theory is to compute a function on the base of a finite sample. The typical case studied is learning a real valued function (the related binary classification problem is often treated as a special case). There is a large literature on the subject; useful reviews are [25, 9, 24, 10, 8], and references therein. In the following we briefly explain the problem.

We consider two sets of random variables $x \in X$, and $y \in Y \subseteq \mathbb{R}$ which are related by a probabilistic relationship. The relationship is probabilistic because, in general, an element of X does not determine uniquely an element of Y, but rather a probability measure on Y. This can be formalized by assuming that an unknown probability measure $\rho(x,y)$ is defined over the set $X \times Y$. We are provided with *examples* of this probabilistic relationship, that is with a training set D_m of m pairs (x_i, y_i) sampled in $X \times Y$ according to $\rho(x,y)$. The goal is to estimate a function $f : X \to Y$ able to predict a value y from any possible value of $x \in X$.

The standard way to solve the learning problem consists in defining an error functional, which measures the average amount of a function, and then looking

[*] The author is with Department of Computer Science, University College London, Gower St., London WC1E 6BT, UK, Email: *m.pontil@cs.ucl.ac.uk*. A longer version of this paper appeared in the *Bulletin of the Italian Artificial Intelligence Association* (AI*IA Notizie), Vol. 1, pp. 8–17, 2003.

for the function with the lowest error. Let $V(y, f(x))$ be a *loss function* measuring the error we make when we predict y by $f(x)$. The *expected error* is defined by

$$E[f] \equiv \int_{X,Y} V(y, f(x))\rho(x, y)\ dxdy.$$

Our desired function is the minimizer of the expected error. We denote this function by f_ρ to emphasize that it depends on the measure ρ. For example, if $V(y, f) = (f - y)^2$ it is easy to see that f_ρ is the regression function, $f_\rho(x) = \int_Y y\rho(y|x)dy$, where $\rho(y|x)$ is the conditional probability of y given x.
Unfortunately, E can not be computed because the measure ρ is unknown. We are only provided with the training set D_m. A natural approach is to replace the expected error with the empirical error

$$E_m(f) = \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(x_i)).$$

We then to minimize E_m in a space $\mathcal{H}$, named the *hypothesis space*. This space reflects our guess about where a good solution could be found. Let f_m be a minimizer of E_m. A main issue in the theory is to study how well f_m "imitates" the true function f_ρ, i.e. to estimate the generalization error: $E(f_m) - E(f_\rho)$. This quantity depends on two competing factors: the number of examples and the size or "capacity" of the hypothesis space. Let $f_{\mathcal{H}}$ be the minimizer of E within the hypothesis space $\mathcal{H}$. The generalization error can be decomposed in two parts: the *sample error*, $E(f_m) - E(f_{\mathcal{H}})$, and the *approximation error*, $E(f_{\mathcal{H}}) - E(f_\rho)$. The latter depends only on $\mathcal{H}$ and ρ but not on the sampled examples. It can be studied using tools from approximation theory. Recent results for RKHS are discussed by Cucker and Smale [8] in the case that V is the square loss. However there is need for more development in this direction. The former is well developed. Its study is rooted in the theory of empirical processes and goes back to the work of Vapnik and Chervonenkis [24] - see also [9] for a nice summary of recent developments in this direction. The general statement of these results is that the inequality

$$E(f_m) - E(f_{\mathcal{H}}) \leq \epsilon(m, 1/h, 1/\delta)$$

holds true with a probability at least $1 - \delta$, with $\delta \in (0, 1)$, and ϵ is a non-decreasing function. The symbol h denotes a collection of parameters which measure the size of $\mathcal{H}$. Appropriate capacity quantities are defined in the theory, the most popular one being the VC-dimension or scale sensitive versions of it [2].
Intuitively, if the capacity of the function space in which we minimize the empirical error is very large and the number of examples is small, the sample error will be large in probability and *overfitting* is very likely to occur. The approximation error, instead, decreases with the size of the hypothesis space. So, in order to achieve good generalization, it is important to find a good trade-off between approximation error and sample error. In Section 4 we discuss regularization-based techniques which provide a general answer to this problem.

3. Reproducing kernel Hilbert spaces

A reproducing kernel Hilbert space (RKHS) [3] is a function space associated to a Mercer kernel.

Definition 3..1 *A function $K : X \times X \to \mathrm{I\!R}$ is called a Mercer kernel if: (a) K is symmetric, $K(x,y) = K(y,x)$ for all $x, y \in X$, (b) K is positive definite, meaning that for all $x_1, \dots, x_\ell \in X$, and $\ell \geq 1$, the matrix with entries $K(x_i, x_j)$ is non-negative definite.*

For $x \in X$, we define $K_x : X \to \mathrm{I\!R}$ as $K_x(t) = K(x,t)$. Let H_0 be the space formed by all finite linear combinations of functions $K_x, x \in X$ (i.e., H_0 is the span of functions K_x). If $f, g \in H_0$, $f(x) = \sum_{i=1}^m \alpha_i K_{x_i}$ and $g(x) = \sum_{i=1}^\ell \beta_i K_{t_i}$, we define the scalar product

$$(f,g)_K = \sum_{i=1}^m \sum_{j=1}^\ell \alpha_i \beta_j K(x_i, t_j).$$

The name reproducing kernel is due to the following *reproducing property*, which follows immediately from the definition of the scalar product:

$$(f, K_x)_K = f(x) \text{ for every } f \in H_0, \ x \in X.$$

We show that $(\cdot,\cdot)_K$ is well defined. It is easy to verify that $(f,g)_K = (g,f)_K$ and $(af + bh, g)_K = a(f,g)_K + b(h,g)_K$. Since K is positive definite it also follows that $(f,f)_K \geq 0$. It remains to verify that $(f,f)_K = 0$ implies $f = 0$. Using the reproducing property we have

$$\begin{aligned} (f + aK_x, f + aK_x)_K &= (f,f)_K + 2a(f, K_x)_K \\ + a^2(K_x, K_x)_K &= 2af(x) + a^2 K(x,x) \geq 0. \end{aligned}$$

The choice $a > 0$ gives $f(x) \geq -\frac{a}{2} K(x,x)$, while $a < 0$ gives $f(x) \leq \frac{|a|}{2} K(x,x)$. Then, since a can be any real number, $f(x)$ must be zero. This argument is true for every $x \in X$. We conclude that $f = 0$.

Definition 3..2 *The RKHS is the closure of space H_0 with respect to the norm induced by the scalar product, $\|\cdot\|_K = \sqrt{(\cdot,\cdot)_K}$.*

Besides the reproducing property, the RKHS enjoys few more key properties.

Proposition 3..1 *Let K be a Mercer kernel and $\mathcal{H}$ the associated RKHS. Then, for every $x, y \in X$*

(a) $K(x,x) \geq 0$.

(b) $|K(x,y)| \leq \sqrt{K(x,x)}\sqrt{K(y,y)}$.

(c) $|f(x)| \leq \|f\|_K \sqrt{K(x,x)}$ *for every* $f \in \mathcal{H}$.

Proof: (a): Note that $\|K_x\|_K^2 = K(x,x)$. (b): We have $K(x,y) = (K_x, K_y)_K$. The result follows by the Cauchy-Schwartz inequality. (c): We first note that $\|K_x\|_K^2 = (K_x, K_x)_K = K(x,x)$. Let $f(x) = \sum_{i=1}^m \alpha_i K_{x_i}$. By the reproducing property, $f(x) = (f, K_x)_K$. The result now follows by the Cauchy-Schwartz inequality:

$$|f(x)| \leq \|f\|_K \|K_x\|_K = \|f\|_K \sqrt{K(x,x)}.$$

We remark that it can be also shown [3] that if a Hilbert space $\mathcal{H}$ admits a kernel function $K : X \times X \to \mathrm{I\!R}$, such that: $K_x \in \mathcal{H}$ for every $x \in X$ and $(f, K_x) = f(x)$, for every $f \in \mathcal{H}$, $x \in X$, then K is a Mercer kernel.

3.1. Feature space and Mercer Theorem

Let $\varphi_n : X \to \mathbb{R}$, for $n = 1, \ldots, N$ be a set of functions. For every $x \in X$, let $\Phi : X \to \mathbb{R}^N$ be given by

$$\Phi(x) = (\varphi_1(x), \ldots, \varphi_N(x)).$$

Consider the kernel

$$K(x,t) = \Phi(x) \cdot \Phi(t) \equiv \sum_{n=1}^{N} \varphi_n(x)\varphi_n(t). \qquad (1)$$

K is a Mercer kernel. In fact, by definition K is symmetric and it is easy to verity that

$$\sum_{i,j=1}^{\ell} \alpha_i \alpha_j K(x_i, x_j) = \left(\sum_{i=1}^{\ell} \alpha_i \Phi(x_i) \right)^2$$

showing that K is also positive definite.
The map Φ is called the *feature map* and the space $\mathcal{Z} = \{\Phi(x) : x \in X, \ \|\Phi(x)\| \leq \infty\}$ the *feature space.*

Example 3..1 (Homogeneous polynomial kernel) *Let $X \subset \mathbb{R}^{n+1}$, $x = (x_0, x_1, \ldots, x_n)$,*

$$K(x,y) = (x \cdot y)^d$$

with d a positive integer and "$\cdot$" the scalar product in $\mathbb{R}^{n+1}$. It is easy to verify that K is of the form in Eq. (1) with $\Phi(x) = \{x^q \sqrt{C_q^d}\}_{|q|=d}$, where we use the notation $q = (q_0, q_1, \ldots, q_n)$, $|q| = \sum_{i=0}^{n} q_i$, $C_q^d = \frac{d!}{q_0! q_1! \cdots q_n!}$. The feature space is made of all the monomials in $\mathbb{R}^{n+1}$ of degree d. There are $\frac{(n+d)!}{n!d!}$ such monomials.

Example 3..2 (Dishomogeneous polynomial kernel) *Let $X \subset \mathbb{R}^n$, $a > 0$, $d \in N, d \geq 1$:*

$$K(x,y) = (a + x \cdot y)^d, \ \ a > 0.$$

This is the same as the above kernel if we define $x' = (\sqrt{a}, x) \in \mathbb{R}^{n+1}$ and $K'(x', y') = K(x,y)$. The feature space consists of all monomials in $\mathbb{R}^n$ of degree at most *d. For instance, if $n = d = 2$ we have*

$$\Phi(x) = (\sqrt{a}, \sqrt{2a}x_1, \sqrt{2a}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

If we set $a = 0$ we obtain the feature map corresponding to the homogeneous polynomial kernel.

In general the number of features, N, can be infinite (see below) provided that the series in r.h.s. of Eq. (1) converges for every $x, y \in X$. In this case $\mathcal{Z}$ is a subset of ℓ^2, the Hilbert space of square summable sequences. In fact, under some general conditions on the space X any Mercer kernel can be equivalently written in the form in Eq. (1), with $N \in \mathbb{N} \bigcup \{\infty\}$. We now discuss this fact.

3.1.1. Integral operators

Let X be a compact metric space and $\mathcal{L}^2_\nu(X)$ the Hilbert space of square integrable functions on X (w.r.t. a positive measure ν, e.g. the Lebesgue measure). Let $C(X)$ be the space of continuous functions on X w.r.t. the norm $\|f\|_{C(X)} = \sup_{x\in X}|f(x)|$.

Definition 3..3 *If K is a continuous Mercer kernel, we define the operator $L_K : \mathcal{L}^2_\nu(X) \to \mathcal{L}^2_\nu(X)$ by $(L_K f)(x) = \int K(x,t)f(t)dt$.*

Theorem 3..1 (Mercer Theorem) *L_K admits a system $\{(\lambda_n, u_n)\}_{n=1}^\infty$ of eigenvalues/functions: $L_K u_n = \lambda_n u_n$, $n \geq 1$, with $\lambda_n \geq \lambda_{n+1} \geq 0$. In addition for all $x, y \in X$, $K(x,y) = \sum_{n=1}^\infty \lambda_n u_n(x) u_n(y)$, where the convergence is absolute and uniform on $X \times X$.*

The theorems say that a continuous Mercer kernel is of the form in Eq. (1) with $\varphi_n = \sqrt{\lambda_n} u_n$. Note that the decomposition depends on the measure ν used in $\mathcal{L}^2_\nu(X)$ and that the basis functions u_n do not need to be neither orthogonal (e.g, in Examples 3..1−3..2 above, they are not) nor linearly independent.
The map Φ is continuous too. In fact it easily follows that

$$\|\Phi(x) - \Phi(y)\|^2_{\ell^2} = K(x,x) + K(y,y) - 2K(x,y)$$

and, since K is continuous, the l.h.s. tends to zero when x tends to y.
If $f = \sum_{i=1}^\ell \alpha_i K_{x_i}$, it is immediate to verify that f can be equivalently written as $f(x) = \sum_{n=1}^N a_n u_n(x)$, with $a_n = \sqrt{\lambda_n}\sum_{i=1}^\ell \alpha_i u_n(x_i)$, and $\|f\|^2_K = \sum_{n=1}^\infty \frac{a_n^2}{\lambda_n}$. The theorem below makes this connection precise.

Theorem 3..2 *If $f, g \in \mathcal{L}^2_\nu$, with $f = \sum_{n=1}^\infty a_n u_n$, and $g = \sum_{n=1}^\infty b_n u_n$, we define $\langle f, g\rangle = \sum_{n=1}^\infty \frac{a_n b_n}{\lambda_n}$. Then, the space*

$$H_K = \{f = \sum_{n=1}^\infty a_n u_n \in \mathcal{L}^2_\nu \mid \sum_{n=1}^\infty \frac{a_n^2}{\lambda_n} < \infty\}.$$

is a Hilbert spaces which coincides with the RKHS $\mathcal{H}$.

This different representation of the RKHS helps better understanding the properties of the functions which belong to it. The case of periodic kernels is particularly instructive.

3.2. Translation invariant and periodic kernels

Take $X = [0, \pi]$ and $K(x,y) = h(x-y)$, where h is defined on $[-\pi, \pi]$, it is continuous and periodic. Since K is symmetric, h is even $(h(x) = h(-x))$. It follows that the Fourier expansion of h involves only cosine functions:

$$h(x) = a_0 + \sum_{n=1}^\infty a_n \cos nx$$

where $a_n = 1/\pi \int_{-\pi}^{\pi} h(x)\cos nx$, $n \geq 1$, and $a_0 = \frac{1}{2\pi}\int_{-\pi}^{\pi} h(x)dx$. Using the property $\cos(x-y) = \sin x \sin y + \cos x \cos y$, we have

$$K(x,y) = a_0 + \sum_{n=1}^\infty a_n \cos nx \cos ny + \sum_{n=1}^\infty a_n \sin nx \sin ny.$$

Now, assuming $a_n \geq 0$, we see that K is of the form in Eq. (1) with

$$\begin{aligned}\Phi(x) &= (\sqrt{a_0}, \sqrt{a_1}\sin x, \sqrt{a_1}\cos x, \ldots \\ &\quad \ldots, \sqrt{a_n}\sin nx, \sqrt{a_n}\cos nx, \ldots).\end{aligned}$$

We have then proved:

Theorem 3..3 *Let $K(x, y) = h(x - y)$. Then K is a Mercer kernel iff h is even and its Fourier coefficients are non-negative.*

What is the RKHS of K? Denote the Fourier coefficients of a function f by

$$f_n^c = \frac{2}{\pi}\int_0^\pi f(x)\cos nx dx, \quad f_n^s = \frac{2}{\pi}\int_0^\pi f(x)\sin nx dx$$

According to Theorem 3..2, the scalar product in the RKHS is

$$\langle f, g\rangle_K \equiv \sum_{n=0}^{\infty} \frac{f_n^c g_n^c + f_n^s g_n^s}{a_n}$$

Periodic kernels provides an intuition about the meaning of norm $\|f\|_K^2$ as a measure of the smoothness of function f: since $\lambda_n \to 0$ when n goes to infinity, components with higher frequencies are more penalized, and, thus, functions in $\mathcal{H}$ cannot oscillate "too much".
The analysis can be extended to $X = [a, b]^k$. In this case we have

$$\Phi(x) \equiv \{\sqrt{a_n}\sin(n \cdot x), \sqrt{a_n}\cos(n \cdot x)\}_{n \in \mathbb{N}^k}$$

where $n = (n_1, \ldots, n_k)$, and $n_i \geq 0$ for $i = 1, \ldots, k$.
Periodic kernels are a special case of translation invariant kernels. The latter are of the type $K(x, t) = K(x - t)$ but are not necessarily periodic. The next example is well known but clarifies this important difference.

Example 3..3 (Gaussian Kernel) *Let $X \subset \mathbb{R}^n$, $K(x, t) = h(x - t)$, with $h(x) = \exp(-\beta\|x\|^2)$. We will show below that K is a Mercer kernel. If we choose $X = [0, \pi]$, $h(0) \neq h(\pi)$, showing that K is not periodic.*

3.3. Form of the kernels

If we are given a feature map $\Phi(x)$, we can immediately build a kernel by setting $K(x, y) = \langle\Phi(x), \Phi(y)\rangle$. However in many cases this feature map is unknown or may not even exists. We then need to verify directly whether a given K is a Mercer kernel. Here we discuss a general result which characterizes families of positive definite functions.
Suppose $K_1, \ldots K_n$ are some Mercer kernels. Let $F : \mathbb{R}^n \to \mathbb{R}$. Which properties of F guarantee that $F(K_1, \ldots, K_n)$ is also a Mercer kernel? The next theorem by [11] provides a complete answer to this question. We first introduce some new notation. Let $\mathcal{P}^n$ be the set of functions $F : \mathbb{R}^n \to \mathbb{R}$ such that for every $\ell \in \mathbb{N}$ the following property is true: if $A_1, \ldots, A_n$ are arbitrary $\ell \times \ell$ positive definite matrices, then also $F(A_1, \ldots, A_n)$ is positive definite. If $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$, and $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{N}^n$, we set $z^\beta = z_1^{\beta_1} \cdots z_n^{\beta_n}$.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 185-196

Theorem 3..4 *The function $F : \mathbb{R}^n \to \mathbb{R}$ belongs to $\mathcal{P}^n$ iff F is real entire of the form*

$$F(z) = \sum_{\beta \in \mathbb{N}^n} c_\beta z^\beta$$

where $c_\beta \geq 0$ for all $\beta \in \mathbb{N}^n$.

We discuss few examples which show the value of this result.

Example 3..4 *Let $F(z) = (a + z)^d, a \geq 0,\ d \in \mathbb{N}$. It is easy to verify that $F \in \mathcal{P}^1$. Then, if we choose $X \subset \mathbb{R}^n$ $(a + x \cdot y)^d$ is a Mercer kernel. If $a > 0$ we have the dishomogeneous polynomial kernel of degree d discussed in Example 3..2. Setting $a = 0$ gives the homogeneous polynomial kernel of Example 3..1.*

Example 3..5 *Let $F(z) = e^{\lambda z}$, $\lambda > 0$. $F \in \mathcal{P}^1$ since:*

$$e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!}.$$

Again, if we choose $X \subset \mathbb{R}^n$, the function $\exp\{\lambda x \cdot y\}$ is a Mercer kernel. This analysis also shows that the feature map consists of all monomials with a scaling factor $1/n!$, being n the degree of the monomial.

The next example shows that neural networks do not implement Mercer kernels.

Example 3..6 *$F(z) = \tanh\{\lambda z\}$ does not belong to $\mathcal{P}^1$ for every choice of $\lambda \in \mathbb{R}$.*

4. Learning algorithms in RKHS

The discussion at the end of Section 2 suggests that in order to achieve good generalization it is important to find the best trade-off between sample error and approximation error. This observation leads to the method of *structural risk minimization (SRM)* and ultimately to regularization.
The idea of SRM [24] is to define a nested sequence of hypothesis spaces $H_1 \subset H_2 \subset \ldots \subset H_p$, where each space H_i has finite capacity. Here we choose H_i to be a subset of a RKHS. A natural choice is $H_i = \{f \in \mathcal{H} \mid \|f\|_K \leq A_i\}$ with $A_1 < A_2 < \ldots < A_p$. Let $f_{m,i}$ be the minimizer of the empirical error in $\mathcal{H}_i$, $i = 1, \ldots, p$. Using such a nested sequence of more and more complex hypothesis spaces, SRM consists in choosing the minimizer of the empirical error in the space H_{i^*} for which the bound on the generalization error

$$E(f_{m,i}) - E(f_\rho) = \text{approx. error}(\mathcal{H}_i) + \epsilon(m, h_i, \delta)$$

is minimized. Further information on SRM can be found in [9]. Unfortunately, the implementation of the SRM method is not practical because it requires to look for the solution of a large number of constrained optimization problems. An alternative approach is to search for the minimum of

$$J(f) = \frac{1}{m} \sum_{i=1}^{m} V(y_i, f(x_i)) + \lambda \|f\|_K^2. \tag{2}$$

Note that J contains both the empirical error and the norm (complexity or smoothness) of f in the RKHS, similarly to functionals considered in regularization theory [23]. The second term in the r.h.s. of Eq. (2) is a penalty term for functions with high capacity. In particular, the larger the *regularization parameter* λ, the smaller the norm of the solution. On the other hand the smaller λ the smaller the empirical error of the solution.
The key issue in SRM is the choice of the hypothesis space, i.e. the element i^* of the structure where the generalization error is minimized. In the case of the functional of Eq. (2), the key issue becomes the choice of the regularization parameter λ. These two problems, as discussed in [10], are related, and the SRM method can in principle be used to choose λ [24]. In practice, however, more practical statistical methods are used such as cross-validation, generalized cross validation, finite prediction error, etc. - see [25] for a review.

4.1. Form of the solution

An important feature of the above regularization functional is that, independently of the loss function V, any minimizer has the same general form

$$f(x) = \sum_{i=1}^{m} \alpha_i K(x, x_i). \tag{3}$$

There are different proofs of this fact which is sometimes named the *representer theorem* [25]. A general approach is to reduce the minimization of 2 to a *minimal interpolation problem* (see, e.g., [16]).

Lemma 4..1 *The solution to the problem:*

$$\min_f \{\|f\|_K \text{ such that} : f(x_i) = y_i, \; i = 1, \ldots, m\}$$

is unique and has the (not unique) form $f = \sum_{i=1}^{m} \alpha_i K(x_i, x)$.

Proof of Eq. (3): Let f_m be a minimizer of 2. Consider the minimum interpolation problem:

$$\min_f \{\|f\|_K \text{ such that} : f(x_i) = f_m(x_i), \; i = 1, \ldots, m\}.$$

Lemma 4..1 tells us that the solution is unique, call it f, and has the form $f = \sum_{i=1}^{m} \alpha_i K(x_i, x)$. Now set $g = f_m - f$. By definition, $g(x_i) = 0$ for $i = 1, \ldots, m$ and, thus, $V(y_i, f_m(x_i)) = V(y_i, f(x_i))$. Note that $\|f_m\|_K^2 = \|f\|_K^2 + 2(f, g)_K + \|g\|_K^2$. But: $(f, g)_K = \sum_{i=1}^{m} \alpha_i (K_{x_i}, g)_K = \sum_{i=1}^{m} \alpha_i g(x_i) = 0$. We conclude that $J(f_m) = J(f) + \lambda \|g\|_K^2$, and, so, $g = 0$.
Eq. (3) establishes a representation of the function f as a linear combination of kernels centered on each data point. This compact representation is of great advantage for learning. It permits to avoid working with the representation $f = \sum_{n=1}^{\infty} a_n u_n$, which requires estimating an infinite number of parameters. In fact, placing Eq. (3) in (2) we have

$$J = \frac{1}{m} \sum_{i=1}^{m} V(y_i, \sum_{j=1}^{m} K_{ij} \alpha_j) + \lambda \sum_{i,j=1}^{m} \alpha_i K_{ij} \alpha_j. \tag{4}$$

Now J depends on f only trough the m parameters α_i.

4.2. Regularization-based learning algorithms

We discuss few learning techniques based on the minimization of functionals of the form (2) by specifying the loss function V.

4.2.1. Regularization Networks

Regularization networks arise from the minimization of the quadratic functional

$$\frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2 + \lambda\|f\|_K^2 \qquad (5)$$

for a fixed λ is a special form of regularization. It can be easily verified (see, e.g., [10]) that the coefficients α_i of the minimizer of (5) are the solution of the following linear system of equations:

$$(G + \lambda I)\alpha = y \qquad (6)$$

where I is the identity $m \times m$ matrix, and we have defined

$$(y)_i = y_i, \quad (\alpha)_i = \alpha_i, \quad (G)_{ij} = K(x_i, x_j).$$

There are many numerical algorithms which can be used to solve the linear system (6). In particular, least squares algorithms are well established - see, e.g., [22].

4.2.2. Support Vector Machines

We distinguish between real output (regression) and binary output (classification, $y \in \{-1, 1\}$) problems. SVM classification corresponds to the following loss function

$$V(y, f(x)) = (1 - yf(x))_+ \qquad (7)$$

where $(x)_+ = x$ if $x > 0$ and zero otherwise. SVM regression uses the loss

$$V(y, f(x)) = (|y - f(x)| - \epsilon)_+.$$

The two losses share the property of being zero below a certain "scale". For SVM regression the mechanism is clear: if $|y - f(x)|$ is less than ϵ the loss is zero. For SVM classification case, instead, the loss is zero if $yf(x) \geq 1$. What does this mean? Since f is a linear function, $f(x) = 0$ is an hyperplane in the RKHS (passing trough the origin) and $|f(x)|/\|f\|_K$ is the distance of x to the hyperplane. Thus, the condition $yf(x) \geq 1$ says that example (x, y) lies of the correct side of the hyperplane (positive if $y = 1$ and negative if $y = -1$) and has a distance of at least $1/\|f\|_K$ to the hyperplane. So, the examples which have zero loss are those which are "easy" to classify. If f separates the examples the SVM algorithm finds, among the infinitely many separating hyperplanes, the one with the smallest norm or, equivalently, with the largest margin[1] $1/\|f\|_K$. Another remarkable property of the SVM losses is that they lead to *sparse* solutions, meaning that, usually, only a small fraction of the coefficients α_i in Eq. (3) are nonzero. The data points x_i associated with the nonzero α_i are called *support vectors*. Those are either the points which have a positive loss or a subset of the points that are at the "edge" between zero and positive loss[2]. We note that the SVM classification technique was originally introduced by Cortes and Vapnik [7] as a quadratic programming problems [24].

[1] This notion of margin should not be confused with the margin of an example (x, y), which is defined to be $yf(x)$.

[2] In practice, all such points – they are also called the points on the margin – have non-zero coefficients, but one can construct special cases where this is not true.

5. Discussion

5.1. How to choose a good kernel?

To fully take advantage of the regularization formulation above it is important to chose a kernel which is appropriate for the problem at hand. A general statement is that it is better to choose a kernel whose associated RKHS is "dense" in the space of continuous functions, meaning that the approximation error is zero. It is also important that the kernel depends on a small number of parameters. The typical choice for $X \subset \mathbb{R}^d$ is the gaussian kernel, $K(x,t) = \exp(-\beta\|x-t\|^2)$, with $\beta > 0$. This kernel works typically very well provided that the parameter deviation β is set appropriately. This parameter as well as the regularization parameter are treated as constants when we minimize functional in Eq. 2. Finding their optimal value is the subject of *model selection*, a problem which is addressed both in machine learning and statistics. The methods which usually work well are those based on using a validation set or $k-$fold cross validation. These methods are well known (see, e.g., the discussion in [25]).

5.2. Historical notes

Positive definite kernels were developed by Mercer in 1909. Subsequently, the theory grew from the contribution of several mathematicians, among whom we remember Bocher, Moore, Schoenberg, and, especially, Aronszajn, who gave the first systematic treatment on the theory of RKHS in his famous 1950 paper. The paper by Cucker and Smale [8] contains an introduction to the subject relevant to learning theory.

The idea of using Mercer kernels in pattern recognition goes back the mid 60's to the work on potential functions by Aizerman *et al.* [1] Around the same period, RKHS were also used, from a different perspective, in approximation theory and statistics (see, e.g., the monograph by Wahba [25]).

Regularization theory was developed in the 60's by the Russian school of mathematicians lead by Tickonov. Its application in learning was championed by Poggio and Girosi in the late 80's to study radial basis functions. [20]. This framework was later extended by Evegniou *et al.* [10] to include SVMs.

The idea of maximum margin classifiers was introduced by Boser *et al.* in 1992 [4] and later refined to SVMs by Cortes and Vapnik [7]. Vapnik also extended SVMs to regression [24]. After these works, SVMs and related kernel methods became increasingly important and are now a main toolbox in computer science and engineering. A substantial part of this development was driven by real applications in different fields, especially those arising in computer vision, natural language processing, speech and sound analysis, and bioinfromatics.

5.3. Applications

We give a brief overview of few of the many applications of the learning techniques discussed in the Section 4, especially SVM classification.

The first application of SVMs dealt with the problem of optical character recognition (see [24] and references therein). Soon after SVMs started to be used as the core classifier of vision systems, for example to identify faces [18], people [19], and for appearance-based 3D object recognition [21]. In all these cases the proposed vision systems were able to deal with *objects* difficult to model due to significant variety of geometry, color, texture, and viewing conditions. At the same time SVM established as the state of the art tool for text categorization problems [13]. Among the more recent applications we recall those on stop word detection in speech signals [17] and on microarray data analysis in bioinformatics

- see, e.g., [5]. The widespread use of SVM in bioinformatics is particularly impressive but also raises the question about how much speculation can influence the diffusion of a learning method in a field where the statistical significance of the results is often a taboo.

All the works mentioned so far used simple kernels in high dimensional vector spaces, such as the gaussian or polynomial kernels. They provided the first indications that SVMs can deal with sparse data in high dimensional spaces. More recent works started to address the problem of building kernels in non Euclidean data spaces. Part of these works focus on defining data representations, formed by features $u_n(x)$, and, afterwards, show that the linear kernel in those features, $K(x,t) = \sum_n u_n(x)u_n(t)$, can be computed efficiently. This includes data such as text documents [5], parsing trees in natural language [6], DNA sequences [15], and so on. The data representation considered in these methods are in the style of the bag of word representation for text documents (see, e.g., [13]) and, therefore, the associate kernel can model only a narrow set of functions. The contribution of these approaches seems to be mainly on feature extraction for complex tasks rather than substantial Mercer kernel development. Other works attempt to developed directly families of kernel functions bypassing the definition of a feature map. In particular, Haussler [12] discusses kernels for recursive structures such as sequences and trees, and Kondor and Laffery [14] use ideas from spectral graph theory to build kernels on graph structures. Interesting, both studies include Euclidean spaces, in which case the proposed kernels reduce to the gaussian kernel. A main drawback of these works is that computing the kernel may be highly time consuming.

5.4. Future directions

We already mentioned in Section 2 the need for more studies on the approximation error in learning theory. Assume that the target function f_ρ belongs to a large space $\mathcal{F}$ (a standard choice is the space of continuous functions). In this setting it would be interesting to study the approximation properties of families of RKHS which are dense in $\mathcal{F}$.

A second research direction which is still mainly unexplored is the development of Mercer kernels in non Euclidean spaces. In particular an important area is learning in discrete structured domains. Examples are spaces of graphs, e.g. trees or sequences. The above approximation problem is also relevant in this context.

Finally, there is need to develop theory and methods for problems beyond the standard classification and regression ones. For instance, there are other learning problems which have received much less attentions: learning order relations, multi-label classification, multiple output regression. A better understanding and theoretical development of those cases will open the way to new applications areas.

References

[1] M.A. Aizerman, A.V. Braverman, L.I. Rozonoer. The problem of pattern recognition learning and the method of potential functions. Autom. Remote Control, 25, 821-837, 1964.

[2] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergnce, and learnability. *J. of the ACM*, 44(4):615–631, 1997.

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

[4] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifier. In *Proc. 5th ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, July 1992.

[5] M.P.S. Brown, W.N Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares Jr., and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl.Acad. Sci.*, (97):262–267, 2000.

[6] M. Collins and N. Duffy. Convolution kernels for natural language. In *Proc. of NIPS 14*, 2002.

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

[8] F. Cucker and S. Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc.*, 39(1):1–49, 2002.

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

[10] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

[11] C.H. FitzGerald, C. A. Micchelli, and A. Pinkus. Functions that preserves families of positive definite functions. *Linear Algebra and its Appl.*, 221:83–102, 1995.

[12] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California, Santa Cruz, 1999.

[13] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *International Conference on Machine Learning (ECML)*, 1998.

[14] R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input structure. In *Proc. of the Int. Conf. Machine Learning*, 2002.

[15] C. Leslie, E. Eskin, and W. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proc. of Pacific. Symp. on Bioinformatics*, 2002.

[16] C. Micchelli and M. Pontil. Learning vector valued functions. Working paper, Department of Mathematics, State University of Ney York, 2002.

[17] P. Niyogi, C. Burges, and P. Ramaesh. Distinctive feature detection using support vector machines. In *Proc. Int. Conf. Acustic, Speech, and Signal Processing*, Phoenix, Arizona, 1999.

[18] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *IEEE Workshop on Neural Networks and Signal Processing*, Amelia Island, FL, September 1997.

[19] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision*, Bombay, India, January 1998.

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

[21] M. Pontil and A. Verri. Support vector machines for 3-d object recognition. *IEEE Trans. PAMI*, pages 637–646, 1998.

[22] *Least Squares Support Vector Machines*, J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, World Scientific, Singapore, 2002.

[23] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston, Washington, D.C., 1977.

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

[25] G. Wahba. *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.