Generalization Bounds for Convex Combinations of Kernel Functions

Alex J. Smola, GMD [1] Robert C. Williamson, ANU [2]
Bernhard Schölkopf, GMD [3]

NeuroCOLT2 Technical Report Series

NC2-TR-1998-022

August, 1998[4]

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

[1] smola@first.gmd.de GMD FIRST, Rudower Chaussee 5, 12489 Berlin, Germany

[2] Bob.Williamson@anu.edu.au Department of Engineering, Australian National University, Canberra, ACT 0200, Australia

[3] bs@first.gmd.de GMD FIRST, Rudower Chaussee 5, 12489 Berlin, Germany

[4] Received 07-AUG-98, Revised 22-OCT-98

Abstract

We derive new bounds on covering numbers for hypothesis classes generated by convex combinations of basis functions. These are useful in bounding the generalization performance of algorithms such as RBF-networks, boosting and a new class of linear programming machines similar to SV machines. We show that p-convex combinations with $p > 1$ lead to diverging bounds, whereas for $p = 1$ good bounds in terms of entropy numbers can be obtained. In the case of kernel expansions, significantly better bounds can be obtained depending on the eigenvalues of the corresponding integral operators.

1 Introduction

It has been shown [13] that good bounds on the generalization error can be obtained in the case of Support Vector (SV) Machines. These carry out regularization in feature space by restricting the weight vector w to lie inside some ball of radius R_w in feature space. Recently new methods have been proposed [3, 11] to compute SV like expansions using linear programming algorithms. The method is to

$$\text{estimate} \quad f(x) = \sum_{i=1}^{\ell} \alpha_i k(x_i, x) + b \tag{1}$$

such that f minimizes a risk functional of the type

$$R_{\text{reg}}[f] = R_{\text{emp}}[f] + C \sum_{i=1}^{\ell} |\alpha_i| \tag{2}$$

where $R_{\text{emp}}[f]$ denotes the empirical error (or risk) or some upper bound thereof. Naively, one could try to use bounds for SV machines [10] for model selection purposes in this case, but this would be very wasteful since the shape of the hypothesis class is completely different (we are dealing with scaled convex combinations of base hypotheses represented by the kernels $k(x_i, x)$). Furthermore, the kernels may not even satisfy Mercer's condition, in which case the standard reasoning fails completely. Nevertheless, this new class of algorithms has been reported to yield competitive generalization performance which creates the need for an explanation of this effect and for some new model selection rules.

The plan of this paper is as follows. First we will present some basic tools necessary for bounding generalization performance. Secondly we will derive bounds for the case that only some general smoothness constraints on the functions are known. Finally we show what additional advantage can be obtained from kernel expansions by exploiting the properties of specific kernels. Note that the bounds we give will apply, unlike [10], both to regression and pattern recognition. This is due to a novel approach introduced in [13] which makes it possible to bypass the VC dimension reasoning and directly compute covering numbers of hypothesis classes.

2 Basic Tools

The first ingredient is a bound on the generalization error in terms of the entropy number ϵ_n or its functional inverse, the covering number $\mathcal{N}(\epsilon)$ of a model class.

The covering number $\mathcal{N}(\epsilon, X, d)$ of a set X equipped with a metric $d(x, y)$ is defined as the minimum number of elements x_i of X such that the minimum distance between arbitrary $x \in X$ and these elements is less equal ϵ, i.e. $\min_i d(x, x_i) \leq \epsilon$. The entropy number of X (w.r.t. d) is $\epsilon_n(X) = \epsilon_n(X, d) = \inf\{\epsilon > 0 \colon \mathcal{N}(\epsilon, X, d) \leq n\}$. Consequently $\mathcal{N}(\epsilon, \mathcal{F}, \ell_\infty^m)$ denotes the ϵ–covering number of the model class $\mathcal{F}$ w.r.t. the ℓ_∞^m metric. We set $\mathcal{N}^m(\epsilon, \mathcal{F}) := \sup_{x_1, \ldots, x_m} \mathcal{N}(\epsilon, \mathcal{F}, \ell_\infty^m)$.

A typical uniform convergence result takes the general form

$$P^m\{R - R_{\text{emp}} > \epsilon\} \leq c_1(m)\mathcal{N}(\epsilon, \mathcal{F}, \ell_\infty^m)e^{-\epsilon^\beta m/c_2}. \tag{3}$$

Here m denotes the number of samples, R_{emp} the empirical and R the expected risk. The constants c_2, β and $c_1(m)$ depend on the setting. For instance in the case of realizable models [1] we have $c_1(m) = 12m, \beta = 2, c_2 = 36$. For more details see [13]. These bounds are typically used by setting the right hand side equal to δ and solving for $m = m(\epsilon, \delta)$. This is called the sample complexity. The key contribution in the present paper involves the *direct* computation of $\mathcal{N}^m(\epsilon, \mathcal{F})$ in a manner that does not involve a combinatorial dimension (such as the VC- or the fat-shattering dimension). This should be contrasted with [7, 2]. We omit discussion of how to take account of the loss functions in terms of which the risk R is being measured; for details see [13].

We need bounds on entropy numbers of convex hulls in terms of the entropy numbers of the base model class. These are used to demonstrate the difference in scaling between general statements on convex combinations and the improvement that can be obtained by explicitly exploiting the kernel map. We can use a special case of [5, Proposition 4.4] to obtain

Proposition 1 (Entropy numbers for convex hulls) *For all Banach spaces X and all precompact subsets $A \subset X$ satisfying the bound $\epsilon_n(A) \leq cn^{-\frac{1}{d}}$ with $c, d > 0$ there exists a constant $\rho(d)$ such that for the convex hull of A ($\operatorname{co}(A)$) the following inequality holds*

$$\epsilon_{2^n}(\operatorname{co}(A)) \leq c\rho(d)n^{-\frac{1}{d}}. \tag{4}$$

The following is standard.

Proposition 2 (Entropy numbers of compact sets) *Given a d–dimensional Banach space X and a compact set C there exists a constant $c(C, X) > 0$ such that the entropy number is bounded as follows.*

$$\epsilon_n(C) \leq c(C, X)\operatorname{vol}(C)n^{-\frac{1}{d}} \tag{5}$$

The constants depend on the geometrical properties of the space, e.g. whether C is a box or a ball.

If $S\colon E \to F$ is a linear operator mapping between two Banach spaces, then $\epsilon_n(S)$ (the entropy number of S) is the entropy number of the image of the unit ball in E as measured in the metric induced by F. Now we can state an extended version of Maurey's theorem (due to Carl) applicable to Banach spaces.

Proposition 3 (Maurey-Carl [4]) *Let $n \in \mathbb{N}$, X a Banach space of type p with Rademacher type constant $\tau_p(X)$, and let $S\colon X \to \ell_\infty^m$ be a linear operator. Then there exists a constant c such that:*

$$\epsilon_{2^n}(S) \leq c\tau_p(X)\|S\| \left(n^{-1} \log\left(1 + \frac{m}{n}\right)\right)^{-1+\frac{1}{p}} \tag{6}$$

The rate in n is optimal. Note that the Rademacher type constant p of Banach spaces is 2 in the case of Hilbert spaces and $L_q(\mathbb{R}^d)$ spaces with $q \geq 2$. Finally one needs a method of combining these bounds, e.g. when mapping sets whose entropy numbers are bounded into another space with operators that might restrict the model class even more. From [6] we get the following proposition.

Proposition 4 (Entropy numbers for concatenations of operators) *Suppose X, Y, Z are Banach spaces and $A : X \to Y$, $B : Y \to Z$ are linear operators. Then the entropy numbers of $AB : X \to Z$ satisfy*

$$\epsilon_n(AB) \leq \epsilon_{n_1}(A)\epsilon_{n_2}(B) \quad \textit{with } n_1 n_2 \in \mathbb{N} \textit{ and } n_1 n_2 \geq n \tag{7}$$

3 Bounds for Convex Combinations

Now we are able to combine the above statements. The statement will be formulated in terms of the Lipschitz constant[1] $c_L(k, C)$ of a class of kernels

$$c_L(k,C) := \inf\{c_L | d(k(x_1,y), k(x_2,y)) \leq c_L d(x_1,x_2) \text{ for all } x_1, x_2 \in C, y \in X\} \tag{8}$$

Here $d(\cdot,\cdot)$ represents the metric on the index set C and the image of k respectively. All commonly used translation invariant SV kernels (e.g. $k(x,y) = \exp(-\|x-y\|^2)$) satisfy this property. This leads to the following proposition.

Proposition 5 (Entropy numbers in L_∞ spaces) *Let X be a d-dimensional Banach space, $C \subset X$ a compact index set, $k(\cdot,\cdot)$ a kernel function defined on $C \times X$ with finite $c_L(k,C)$ and $k(x,y) \leq 1$ for all $x \in C$, $y \in X$. Let*

$$\mathcal{F} := \left\{ f\colon X \to \mathbb{R} \,\middle|\, f = \sum_i \alpha_i k(x_i, \cdot) \textit{ with } \sum_i |\alpha_i| \leq 1 \right\}. \tag{9}$$

Then there exists a positive constant $c(C,d,X) > 0$ such that

$$\epsilon_{2^n}(\mathcal{F}, L_\infty) \leq c(C,d,X) c_L(k,C) n^{-\frac{1}{d}}. \tag{10}$$

[1] A more general formulation in terms of the modulus of continuity $\omega(k,\delta,C)$ is straightforward but has been omitted for the sake of simplicity.

Proof The first step is to compute an upper bound on $\epsilon_n(\mathcal{K}) = \epsilon_n(\mathcal{K}, L_\infty)$ where

$$\mathcal{K} := \{k(x, \cdot) | x \in C\} \tag{11}$$

in terms of the entropy numbers of C. By definition we have $\|k(x_i, \cdot) - k(x_j, \cdot)\|_{L_\infty} \leq c_L(k, C) d(x_i, x_j)$ for $x_i, x_j \in C$ and therefore

$$\epsilon_n(\mathcal{K}) \leq c_L(k, C) \epsilon_n(C). \tag{12}$$

As we are interested in the absolute convex combination, i.e. $\mathcal{F} = \operatorname{co}(\mathcal{K} \cup -\mathcal{K})$ one has to take into account that the set of base hypotheses is twice as large as $\mathcal{K}$. From proposition 2 we can obtain a bound on $\epsilon_n(C)$ to obtain

$$\epsilon_{2n}(\mathcal{K} \cup -\mathcal{K}) \leq c_L(k, C) c(C, X) \operatorname{vol}(C) n^{-\frac{1}{d}} \tag{13}$$

Now apply proposition 1 to obtain

$$\epsilon_{2^n}(\operatorname{co}(\mathcal{K} \cup -\mathcal{K})) \leq \rho(d) 2^{\frac{1}{d}} c(C, X) \operatorname{vol}(C) c_L(k, C) n^{-\frac{1}{d}}. \tag{14}$$

Collecting the constants into $c(C, d, X)$ gives the desired result. ∎

Next we can bound ϵ_n on an arbitrary m-sample $x^m := \{x_1, \ldots x_m\} \subset X$, in the ℓ_∞^m metric for $\Lambda\mathcal{F}$, i.e. $\mathcal{F}$ scaled by the constant Λ. For this purpose we introduce the *evaluation operator* S_{x^m} as

$$S_{x^m} : L_\infty(X) \to \ell_\infty^m \qquad S_{x^m} f = (f(x_1), \ldots, f(x_m)) \tag{15}$$

The first thing to note is that S_{x^m} is linear and has norm 1 due to the L_∞ norm. Hence we can apply prop. 4 and 3 to bound $\epsilon_n(S_{x^m} \Lambda\mathcal{K})$ by $\epsilon_{n_1}(S_{x^m}) \epsilon_{n_2}(\Lambda\mathcal{K})$ with $n_1 n_2 \geq n$.

Proposition 6 *The entropy number of the hypothesis class $\Lambda\mathcal{K}$ evaluated at m arbitrary points $\{x_1, \ldots, x_m\} \subset X$ in the ℓ_∞^m metric is bounded by*

$$\epsilon_n(S_{x^m}(\Lambda\mathcal{K})) \leq \Lambda \tilde{c}(C, d, X) c_L(k, C) \inf_{n_1, n_2 \in \mathbb{N}, n_1 \cdot n_2 \geq n} \left(n_1^{-1} \log \left(1 + \frac{m}{n_1} \right) \right)^{-\frac{1}{2}} n_2^{-\frac{1}{d}} \tag{16}$$

for some constant $\tilde{c}(C, d, X) > 1$. Hence the rate in ϵ_n is of order $O(n^{-\frac{1}{2} - \frac{1}{d}})$ as can be checked by carrying out the inf.

Since x^m was arbitrary, we can thus bound $\mathcal{N}^m(\epsilon, \mathcal{F})$. Although it is certainly not explicitly obvious here, it turns out that the bounds obtained here are tighter than those in [7, 2] derived using the fat-shattering dimension, a version of Maurey's theorem and the generalization of the Sauer-Shelah-Vapnik-Chervonenkis lemma in [1]. As we will show subsequently, one can do much better by exploiting properties of kernels in a more explicit way.

4 A Remark on Traditional Weight Decay

One might conjecture that a similar result could be established for p–convex combinations with $p > 1$, i.e.

$$\mathcal{F}_p := \left\{ f \,\middle|\, f = \sum_i \alpha_i k(x_i, \cdot) \text{ with } \sum_i |\alpha_i|^p \leq 1 \right\} \tag{17}$$

Training large neural networks with weight decay ($p = 2$) is such a case. However, under the assumption of an infinite number of basis functions this conjecture is false. It is sufficient to show that $\mathcal{F}_p$ is unbounded in L_∞. Consider an infinite index set $I \subset C$ for which, for some other set M of nonzero measure and some constant $\kappa > 0$

$$k(x_i, x) \geq \kappa \quad \text{for all} \quad x_i \in I, x \in M. \tag{18}$$

An example is $k(x, y) = e^{(-(x-y)^2)}$. Any compact sets I, M satisfy (18). Obviously $f(x) := \sum_j \alpha_j k(x_i, x) \geq \kappa \sum_j \alpha_j$ for $\alpha_j \geq 0, x_i \in I, x \in M$. Now let $f_n(\cdot) := \sum_{j=1}^n n^{-1/p} k(x_i, \cdot)$. By construction the ℓ_p^n norm of the coefficients equals 1, however $f_n(x) \geq \kappa n^{1-1/p}$ for all $x \in M$. Thus $\lim_{n\to\infty} \|f_n\|_{L_\infty} = \infty$ and therefore $\mathcal{F}_p$ contains unbounded elements for $p > 1$, which leads to infinitely large covering numbers for $\mathcal{F}_p$. Hence $\mathcal{F}_p$ with $p > 1$ is not a suitable choice as a hypothesis class (in the absence of further regularization).

This leads to the question why, despite the previous reasoning, weight decay has been found to work in practice. One reason is that in standard neural networks setttings the number of basis functions is limited (either by construction, via some penalty term, etc.), thus the above described situation might not occur.

Secondly, e.g. in rbf–networks, a clustering step for finding the centers is inserted before training the final weights. This means that the basis functions are sufficiently different from each other — observe that the similarity of some basis functions was explicitly exploited in the counterexample above.

Finally, also by the distance of the basis functions, penalization with a diagonal matrix is not too different from penalization via a kernel matrix (provided the widths of the basis functions is equal, and not significantly larger than the distance between the centers) — the main diagonal elements will be 1 and the off diagonal elements rather small, thus an approximation by the unit matrix is not too unrealistic.

There exists, however, a case where this reasoning might go wrong in practice. Assume one wants to modify a boosting algorithm in such a way that instead of convex combinations one would like to have p–convex combinations with $p > 1$. After iterating a sufficiently long time the situation described above might occur as the number of basis functions (i.e. weak learners) keeps on increasing with the iterations.

5 The Kernel Advantage

To get better bounds one has to take a completely different view of the kernels. The strategy is to view the hypothesis class as contained in the image of a linear operator (as in the SV case), however with possibly different constraints. It will become clear that the statements about the image of the data in feature space can be kept. The weight vector instead is not constrained any more to a ball of some fixed radius Λ but to a convex set, identical in shape to the images of the data, i.e. a box with rapidly decaying sidelengths.

In addition to the previous assumptions require that k is symmetric, bounded, and that the kernel expansion consists only of functions with $k(x_i, \cdot)$ where $x_i \in \mathcal{X}$. If one requires that the training data be constrained to some compact set $C \subset X$ one can find an expansion of k in terms of its eigenfunctions by

$$k(x,y) = \sum_i \lambda_i \phi_i(x) \phi_i(y) \tag{19}$$

similar to the expansion in terms of eigenfunctions as stated in Mercer's theorem (cf. [8]) — the assumption of positive symmetric kernel is losened to a solely symmetric kernel (and corresponding integral operator). Thus positivity of the eigenvalues cannot be ensured. However the restrictions on $\phi_i(x)$ such as boundedness of ϕ_i and orthogonality still apply. Without loss of generality, assume that the coefficients λ_i have been ordered in decreasing order of their absolute value.

Positivity of the kernel was a central condition in the SV approach [13] but irrelevant in the Linear Programming setting. The only thing one cannot assume in the general case any more is that the bilinear form has positive signature. But this is not a major restriction as this effect could be taken care of by a redefinition of the weight vector. The advantage is, that nearly any symmetric function $k(x, y)$ can be brought into this form. For instance B_q–splines of even order which do not satisfy Mercer's condition can be employed in linear programming type learning algorithms.

In any case functions $f \in \mathcal{F}$ can be viewed as dot products in feature space by transforming

$$f(x) = \sum_{i=1}^{\ell} \alpha_i k(x_i, x) = \sum_{i=1}^{\ell} \alpha_i \sum_j \lambda_j \phi_j(x_i) \phi_j(x) = \langle w, \Phi(x) \rangle. \tag{20}$$

Here w and $\Phi(x)$ are defined as follows (for SV kernels this definition coincides with the standard form derived from Mercer's theorem, cf. [13]):

$$\begin{aligned} \Phi(x) &:= \left(\sqrt{|\lambda_1|}\phi_1(x), \sqrt{|\lambda_2|}\phi_2(x), \dots\right) & (21) \\ w &:= \left(\sqrt{|\lambda_1|}\operatorname{sign}(\lambda_1)\sum_{i=1}^{\ell}\alpha_i\phi_1(x_i), \sqrt{|\lambda_2|}\operatorname{sign}(\lambda_2)\sum_{i=1}^{\ell}\alpha_i\phi_2(x_i), \dots\right) & (22) \end{aligned}$$

Moreover we can reuse the reasonings of [13] to infer that that $\Phi(C) \cup -\Phi(C)$ is contained in some hyperellipsoid $\mathcal{E}$ in feature space. The exact shape of

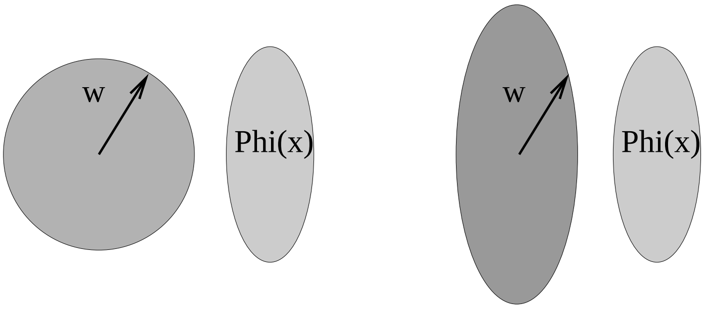

Figure 1 Left: In the SV case the weight vector w is contained in a ball of some (given) radius and the data lies inside some hyperellipsoid. Right: In the convex combination algorithms the weight vector is contained in a scaled version of the convex hull of the data, i.e. a hyperellipsoid of identical shape but different size.

$\mathcal{E}$ depends on the kernel k at hand. This is exactly the property one has to take advantage of to derive good bounds. In particular, we may construct an operator A mapping the unit ball in ℓ_2 to $\mathcal{E}$, i.e.

$$\begin{tikzcd}
\mathcal{X} \arrow[r, "\Phi"] & \Phi(\mathcal{X}) \arrow[r, "A^{-1}"] \arrow[d, phantom, "\cap"] & U_{\ell_2} \arrow[dl, "A"] \\
& \mathcal{E} &
\end{tikzcd} \tag{23}$$

In particular, the operator A will be useful for computing the entropy numbers of concatenations of operators. One thus seeks an operator $A : \ell_2 \to \ell_2$ such that

$$A(U_{\ell_2}) \subseteq \mathcal{E}. \tag{24}$$

This can be ensured by constructing A such that

$$A\colon (x_j)_j \mapsto (R_A \cdot a_j \cdot x_j)_j \tag{25}$$

with $R_A := C_k \|(\sqrt{|\lambda_j|}/a_j)_j\|_{\ell_2}$ where C_k is a bound on $|\phi_i(x)$ for all i, x. Moreover note that the absolute convex combination, i.e. $\operatorname{co}(\Phi(C) \cup -\Phi(C)) \subset \mathcal{E}$ as $\mathcal{E}$ is convex, and the weight vector w is contained in a scaled version of the hyperellipsoid, i.e. $w \in \Lambda\mathcal{E}$ with $\Lambda = \sum_{i=1}^{\ell} |\alpha_i|$ by construction.

Hence the situation (see Fig. 1) is quite similar to the SV case [13]. The mapped data is contained inside some hyperellipsoid. The weight vector w, however, is constrained to a ball in the SV case and to a hyperellipsoid of the same shape as the original data in the LP case. This means that while in SV machines capacity is allocated equally in all directions, in the convex combination algorithms much capacity is allocated in those directions where the data is spread out a lot and little capacity where there is little spread. Before doing the exact calculations one has to define an appropriate sampling operator S_{X^m}. Be $\Phi(X^m) := \{\Phi(x_1), \ldots, \Phi(x_m)\} \subset \mathcal{E}$. Then $S_{\Phi(X^m)}$ can be defined as follows:

$$\begin{array}{lcl} S_{\Phi(X^m)}\colon U_{\ell_2} & \to & \ell_\infty^m \\ S_{\Phi(X^m)}\colon w & \mapsto & (\langle w, \Phi(x_1)\rangle, \ldots, \langle w, \Phi(x_m)\rangle) \end{array} \tag{26}$$

The present considerations lead to the following theorem for Linear Programming capacity control in analogy to the results in [13].

Theorem 7 (Bounds for Linear Programming Machines) *Let k be a symmetric bounded kernel, let Φ be induced via (21) and let $T := S_{\Phi(X^m)}\Lambda$ where $S_{\Phi(X^m)}$ is given by (26) and $\Lambda \in \mathbb{R}^+$. Let A be defined by (25). Then the entropy numbers of T (and hence the entropy numbers of $\Lambda\mathcal{F}_1$) satisfy the following inequalities:*

$$
\begin{aligned}
\epsilon_n(T) &\leq \mathfrak{c}\|A\|^2\Lambda \log^{-1/2} n \log^{-1/2}\left(1+\tfrac{m}{\log n}\right) && (27)\\
\epsilon_n(T) &\leq 6\Lambda\epsilon_n(A^2) && (28)\\
\epsilon_{nt}(T) &\leq 6\mathfrak{c}\Lambda \log^{-1/2} n \log^{-1/2}\left(1+\tfrac{m}{\log n}\right)\epsilon_t(A^2) && (29)
\end{aligned}
$$

where $\mathfrak{c} \leq 5.3771$ for the case of Hilbert spaces in proposition 3. For a proof see [12].

This result (and also its proof) is quite similar to that in [13], just that the weight vector is constrained to a different set, thus the double appearance of the operator A.

Proof Equation (30) indicates line of reasoning which shall be followed in bounding $\epsilon_n(T\colon \ell_2 \to \ell_\infty^m)$.

$$
\begin{tikzcd}
U_{\ell_2} \arrow[rr, "T"] \arrow[d, "\Lambda"'] & & \ell_\infty^m \\
\Lambda U_{\ell_2} \arrow[r, "A"] & \Lambda\mathcal{E} = \Lambda A U_{\ell_2} \arrow[ur, "S_{\Phi(X^m)}"] \arrow[r, "A"] & \Lambda A^2 U_{\ell_2} \arrow[u, "S_{A^{-1}\Phi(X^m)}"']
\end{tikzcd}
\qquad (30)
$$

In order to bound ℓ_∞^m entropy numbers of the hypothesis class evaluated on an m–sample test set X^m, one has to bound $\epsilon_n(S_{\Phi(X^m)}(\Lambda\operatorname{co}(\Phi(C) \cup -\Phi(C))))$. Thus the diagram yields

$$
\begin{aligned}
S_{X^m}(\Lambda\operatorname{co}(\Phi(C) \cup -\Phi(C))) &\subset S_{X^m}(\Lambda\mathcal{E})\\
&= S_{X^m}(\Lambda A U_{\ell_2}) && (31)\\
&= S_{A^{-1}X^m}(\Lambda A A U_{\ell_2}).
\end{aligned}
$$

where for the last step the equality $S_{X^m}Ax = S_{A^{-1}X^m}x$ was used, which holds due to the representation of f as a linear functional in some feature space. Using (31) and proposition 4 one obtains

$$
\begin{aligned}
\epsilon_n(S_{\Phi(X^m)}\Lambda\operatorname{co}(\Phi(C) \cup -\Phi(C))) &\leq \epsilon_n(T)\\
&\leq \Lambda\epsilon_n(S_{A^{-1}\Phi(X^m)}A^2)\\
&\leq \inf_{n_1,n_2\in\mathbb{N}, n_1 n_2 \geq n} \epsilon_{n_1}(S_{A^{-1}X^m})\epsilon_{n_2}(A^2).
\end{aligned}
\qquad (32)
$$

Combining the factorization properties obtained above with proposition 4 yields the desired results: by construction, due to the Cauchy–Schwartz inequality

$\|S_{A^{-1}\Phi(X^m)}\| = 1$. Since $S_{A^{-1}\Phi(X^m)}$ is an operator mapping from a Hilbert space ℓ_2 into an m–dimensional Banach space ℓ_∞^m one can use Maurey's theorem (see proposition 3 in the special case of $p = 2$). ∎

This allows one to leverage the results from [13], obtained on the properties of $\mathcal{E}$ and consequently also A, for specific kernels. In particular one gets the following two propositions which follow immediately from their counterparts for the case of SV regularization.

Proposition 8 (Polynomial Decay) *Let k be a symmetric kernel with eigenvalues satisfying $|\lambda_j| = \beta^2 i^{-(\alpha+1/2)}$ for some $\alpha > 0$. Then*

$$\epsilon_n(A^2 \colon \ell_2 \to \ell_2) = O\left((\ln n)^{-\alpha + O(\ln^{-2} \ln n)}\right) = O(\ln^{-\alpha} n). \tag{33}$$

This result can be seen as follows. As A is a diagonal scaling operator, the scaling factors of A^2 are simply those of A squared, i.e. decaying twice as fast. Comparing the result with its SV counterpart in [13] shows that the condition on the eigenvalues was changed from $i^{-(\alpha/2+1)}$ into $i^{-(\alpha+1/2)}$. The conclusions and the method of proving this, however, remain unchanged. A similar result can be stated for exponentially decaying eigenvalues of k.

Proposition 9 (Polynomial Exponential Decay in $\mathbb{R}^d$) *For translation invariant kernels $k(x, x') = k(x - x')$ in $\mathbb{R}^d \times \mathbb{R}^d$ with Fourier transform satisfying $|F[k](\omega)| \leq \beta^2 e^{-\alpha\|\omega\|^p}$ with $\alpha, \beta, p > 0$ and corresponding operator A one has*

$$\ln \epsilon_n^{-1}(A^2 \colon \ell_2 \to \ell_2) = O(\ln^{\frac{p}{p+d}} n) \tag{34}$$

Analogous results hold for the other propositions obtained in the previous chapter. Note that whereas in the first case an improvement of the rates in n was achievable, in the latter case no such thing happened — this is due to the fact that in the latter case one is dealing with bounds on the rate of $\ln \epsilon_n$ instead of ϵ_n. It is worth while noticing that the constants, however, do change and thus make the overall bounds behave significantly better than before.

It might look like as if, due to the considerations above, linear programming machines should be preferred to Support Vector regularization machines. This need not necessarily be the case — the rate bounds only tell how "well-behaved" a certain class of models is, not how small the empirical error for a comparable bound of the generalization error might be.

What is happening is that the capacity is distributed *differently* among the class of kernel expandable functions, i.e. a different structure $\mathfrak{F}$ is chosen. More emphasis is put on the first eigenfunctions of the kernel. If one has experimental evidence that this might be useful (say, e.g. from compression experiments [9]), one should consider using such a regularizer.

Examples of such kernels are Gaussian RBF–kernels $k(x, x') = \exp(-\|x - x'\|^2)$ $(p = 2)$, or the "damped harmonic oscillator" kernel $k(x, x') = 1/(1 + \|x - x'\|)$ $(p \geq 1)$. As we are dealing with $\epsilon_n(A^2)$ we are able to get better bounds (in terms of learning rates) than in the SV case where we have to bound $\epsilon_n(A)$. Finally one has to combine proposition 9 and (3) to obtain the overall rates.

Unlike in the previous case, however, the rate given by the specific properties of the kernel is much faster than the one obtained from the Maurey–Carl result and thus determines the overall learning rate.

6 Discussion

Due to space constraints, it is impossible to explain these results (including constants) in more detail here. In particular we have limited ourselves in this exposition to outlining how the learning rates could be computed. For a successful learning algorithm, however, good estimates of the constants (and not only the rates) are crucial. We refer the reader to [13] for calculation of the latter and algorithms to obtain even tighter bounds, by evaluating numerically what had been simply majorized in the proofs. It is worthwhile noticing that by using suitable kernels (e.g. Gaussian RBF) exponentially better rates (Prop. 9) than those for arbitrary kernels (Prop. 6) can be obtained — observe the ln in (34).

As a consequence we showed that linear programming machines in fact do carry out a reasonable way of regularization, yet quite different from the regularization of SV machines, and that by taking advantage of the specific properties of kernels good bounds can be obtained. Note that this is no statement that those machines perform better or worse than SV machines, just that the "size" of the classes differs and thus so do the generalization error bounds obtained via uniform convergence theorems.

Acknowledgements: This work was supported by the ARC and the DFG (# Ja 379/71).

References

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

[2] P.L. Bartlett. The sample complexity of pattern classsification with neural networks: the size of the weights is more important than the size of the network. *IEEE Trans. Information theory*, 44(2):525–536, 1998.

[3] K. Bennett. Combining support vector and mathematical programming methods for induction. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods - SV Learning*, Cambridge, MA, 1999. MIT Press.

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

[5] B. Carl, I. Kyrezi, and A. Pajor. Metric entropy of convex hulls in Banach spaces. preprint, 1997.

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

[7] L. Gurvits. A note on a scale-sensitive dimension of linear bounded functionals in Banach spaces. In M. Li and A. Maruoka, editors, *Algorithmic Learning Theory ALT-97*, LNAI-1316, pages 352–363, Berlin, 1997. Springer.

[8] F. Riesz and B.S. Nagy. *Functional Analysis.* Frederick Ungar Publishing Co., 1955.

[9] B. Schölkopf, S. Mika, A.J. Smola, G. Rätsch, and K.-R. Müller. Kernel PCA pattern reconstruction via approximate pre–images. In L. Niklasson, M. Boden, and T. Ziemke, editors, *International Conference on Artificial Neural Networks ICANN'98*, Lecture Notes In Computer Science. Springer, 1998.

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

[11] J. Weston, A Gammerman, M. O. Stitson, V. Vapnik, V.Vovk, and C. Watkins. Density estimation using sv machines. Technical Report CSD-TR-97-23, Royal Holloway, University of London, Egham, UK, 1997.

[12] R.C. Williamson, B. Schölkopf, and A.J. Smola. A Maximum Margin Miscellany. Typescript, December 1998.

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