21 Entropy Numbers for Convex Combinations and MLPs

Alexander J. Smola
Department of Engineering
Australian National University
Canberra 0200 ACT
Australia
Alex.Smola@anu.edu.au
http://spigot.anu.edu.au/∼smola/

André Elisseeff
Université Lyon 2,
Laboratoire ERIC,
69676 Bron Cedex
France
aelissee@univ-lyon2.fr
http://eric.univ-lyon2.fr/pages_defaut/aelissee.html

Bernhard Schölkopf
Microsoft Research Limited
St. George House, 1 Guildhall Street
Cambridge CB2 3NH
UK
bsc@microsoft.com
http://www.research.microsoft.com/∼bsc/

Robert C. Williamson
Department of Engineering
Australian National University,
Canberra, ACT, 0200
Australia
bob.williamson@anu.edu.au
http://spigot.anu.edu.au/∼williams/home.shtml

Bounds on the generalization performance of algorithms such as boosting, linear programming machines and (multilayer) RBF-networks require a good estimate of the covering or entropy numbers for the corresponding hypothesis classes. The classes are generated by convex combinations and concatenations of basis functions for which we provide functional analytic bounds on the entropy numbers.

The results are novel in three regards. First, bounds are given for vector valued functions directly without having to use a generalization of the VC dimension or other combinatorial quantities. Secondly, bounds are derived for convex combinations of parametric families. It is shown that significantly better bounds can be obtained depending on the eigenvalues of the corresponding integral operators when one deals with kernel functions. Finally, a concatenation theorem allows the use of the previously established results to images of nonlinear operators, such as the outputs of multilayer networks.

21.1 Introduction

Theoretical bounds on the generalization performance of Support Vector (SV) Machines follow from general results of statistical learning theory along with good bounds on covering numbers for the class of functions induced by such machines (cf., e.g., Section 1.2.3). Williamson et al. [1998] show how bounds can be obtained using the machinery of entropy numbers of operators. This is possible because the class of functions is defined via a restriction of a weight vector w to lie within a certain ball in feature space.

The present chapter extends and modifies the methods of [Williamson et al., 1998] in order to deal with other types of learning machines as well. These include convex combinations of hypotheses as used in boosting [Schapire et al., 1998] and linear programming machines [Bennett, 1999, Weston et al., 1999], or concatenations of hypothesis classes such as multilayer rbf networks. In particular our results apply to the algorithms given in Chapter 8 and Chapter 12. As a by-product we also provide good bounds for the problem of estimating vector valued functions.

Applications:
Kernel Boosting
LP Machines
RBF Networks

The generalization performance of learning machines can be bounded in terms of the covering number (see Definition 1.8) of the loss function induced hypothesis class. The necessary tools can be found in Section 1.2.3. In particular Theorem 1.10 states the connection between $\mathcal{N}$ and $R(f)$.

Before going into the actual calculations let us briefly review existing results on this topic. Covering numbers for classes of functions defined via an ℓ_2 constraint were proven for convex combinations of hypotheses by Lee et al. [1996]. Their result, however, was solely based on a theorem of Maurey [1981] ignoring effects of the kernel. Gurvits and Koiran [1997] give a bound for similar settings and Bartlett [1998] proved bounds on the fat shattering dimension based on a weight constraint of Multilayer Perceptrons. See also [Anthony and Bartlett, 1999, Sec. 14.6] for an overview.

In addition to the techniques pointed out in Chapter 1 we will have to introduce entropy numbers (the functional inverse of covering numbers) along with a number of background results in Section 21.2. Next we present a result concerning arbitrary convex combinations of a parametrized family of functions (Section 21.3).

In Section 21.4 we show how one may exploit the geometry of base hypothesis classes H induced by kernels to obtain bounds on $\mathcal{N}(\epsilon, \mathrm{co}\,(H))$ (the ϵ-covering numbers of the convex hull of H) that are better than those obtainable via general results (such as those by Carl et al. [1999]) which are solely in terms of $\mathcal{N}(\epsilon, H)$.

Finally in Section 21.5 we show how to apply the reasoning based on linear operators and their entropy numbers to classes of functions that certainly cannot be expressed in terms of a single linear operator — the most interesting for learning applications being multilayer perceptrons.

All of the proofs are in the appendix. Also in the appendix is an illustration of the difficulty in using p-convex combinations, with $p > 1$, when there are an infinite number of terms (when $p = 2$, this corresponds to traditional weight decay in the limit of an infinite number of nodes).

21.2 Tools from Functional Analysis

As already pointed out in the introduction, our aim is to provide good bounds on $\mathcal{N}$. While direct computation of the latter is often quite difficult, the use of its functional inverse, the so called entropy number ϵ_n is more amenable to practical analysis.

Definition 21.1 Entropy Numbers

Entropy Numbers

Denote by $U_{\mathcal{A}}$ the unit ball in a metric space $\mathcal{A} = (\mathcal{A}, d)$. The n-th entropy number $\epsilon_n(A) = \epsilon_n(A, d)$ of a set $A \subset \mathcal{A}$ with respect to the metric d is defined as the minimum radius ϵ of balls such that there exist $a_1, \ldots, a_n \in \mathcal{A}$ with $A \subset \bigcup_{i=1}^{n} \epsilon U_{\mathcal{A}} + a_i$.

If $\mathcal{A}$ and $\mathcal{B}$ are normed spaces (e.g., Banach spaces), the entropy number $\epsilon_n(T)$ of an operator $T : \mathcal{A} \to \mathcal{B}$ is defined as the entropy number of the image of the unit ball, i.e., $\epsilon_n(T) := \epsilon_n(T(U_{\mathcal{A}}))$ and the ϵ-covering of $T(U_{\mathcal{A}})$ is with respect to the metric of the space $\mathcal{B}$. We sometimes write $\epsilon_n(T, \mathcal{B})$ to make the metric involved explicit.

By construction ϵ_n is the functional inverse of $\mathcal{N}(\epsilon)$; hence if we can view the class of functions used by a learning machine as generated by applying some operator to a unit ball in some space, we will be able to bound the covering numbers of the machine in terms of the entropy numbers of the operator. If $\mathcal{A}$ and $\mathcal{B}$ are Banach spaces, we will denote by $\mathfrak{L}(\mathcal{A}, \mathcal{B})$ the set of all bounded linear operators mapping from $\mathcal{A}$ to $\mathcal{B}$.

For some learning machines one needs bounds on entropy numbers of convex hulls in terms of the entropy numbers of the base model class. In this chapter we

will demonstrate the difference in scaling between general statements on convex combinations and the improvement that can be obtained for the special class of kernel functions by explicitly exploiting the kernel map. We can use a special case of [Carl et al., 1999, Corollary 4.5].

Proposition 21.2 Entropy numbers of Convex Hulls
For all Banach spaces $\mathcal{A}$ and all precompact subsets $A \subset \mathcal{A}$ satisfying the bound

$$\epsilon_n(A) \leq cn^{-\frac{1}{p}} \text{ with } c, p > 0, n \in \mathbb{N} \tag{21.1}$$

there exists a constant $\rho(p)$ such that for all $n \in \mathbb{N}$,

$$\epsilon_{2^n}(\operatorname{co}(A)) \leq c\rho(p)n^{-\frac{1}{p}}, \tag{21.2}$$

Convex Hulls

where $\operatorname{co}(A) = \bigcup_{n=1}^{\infty}\{\sum_{i=1}^{n}\alpha_i a_i | a_i \in A,\ \sum_i |\alpha_i| \leq 1\}$ is the (symmetric absolute) convex hull of A.

This result will be useful in computing the entropy number of convex combinations, once the entropy number of the base class has been determined. The following proposition which follows directly from volume considerations addresses the latter problem.

Compact and finite dimensional

Proposition 21.3 Compact sets
Given a p-dimensional Banach space $\mathcal{A}$ and a compact set $\Gamma \subset \mathcal{A}$ there exists a constant $c(\Gamma, \mathcal{A}) > 0$ such that the entropy number satisfies

$$\epsilon_n(\Gamma) \leq c(\Gamma, \mathcal{A})\operatorname{vol}(\Gamma)^{\frac{1}{p}}n^{-\frac{1}{p}}. \tag{21.3}$$

The constants depend on the geometrical properties of the space, e.g., whether Γ is a box or a ball. Finally we need another bound to take advantage of the fact that we only evaluate a function $f \in F$ on an m-sample. This can be achieved by Maurey's theorem (see [Carl, 1985]). We state a special case applicable to Hilbert spaces, since that is all we need in the present paper.

Hilbert spaces

Proposition 21.4 Maurey, Carl
Let $m \in \mathbb{N}$, H a Hilbert space and let $S \in \mathfrak{L}(H, \ell_\infty^m)$ be a linear operator. Then there exists a constant c such that

$$\epsilon_{2^n}(S) \leq c\|S\|\left(n^{-1}\log\left(1+\tfrac{m}{n}\right)\right)^{\frac{1}{2}}. \tag{21.4}$$

Vector-valued sequence spaces

We will make use of vector-valued sequence spaces. If $\mathcal{X}$ is a normed space with norm $\|\cdot\|_{\mathcal{X}}$, and $x = (x_1, \ldots, x_m)^T \in \mathcal{X}$, then $\|x\|_{\ell_p^m(\mathcal{X})} := \|(\|x_1\|_{\mathcal{X}}, \ldots, \|x_m\|_{\mathcal{X}}\|_{\ell_p^m}$ where $\|\cdot\|_{\ell_p^m}$ is the traditional ℓ_p^m norm, $\|z\|_{\ell_p^m} = (\sum_{i=1}^m |z_i|^p)^{1/p}$. In particular denote by $\ell_p^q(\ell_r^s)$ a "mixed" norm acting on $\mathbb{R}^{q\cdot s}$. (See [Diestel et al., 1995, p.32].)

Corollary 21.5 Bounds for $\ell_\infty^m(\ell_1^d)$ spaces
Let $m, d \in \mathbb{N}$, H a Hilbert space, and let $S \in \mathfrak{L}(H, \ell_\infty^m(\ell_1^d))$. Then there exists a constant $c = c(d)$ such that

$$\epsilon_{2^n}(S) \leq c\|S\|\left(n^{-1}\log\left(1+\tfrac{md}{n}\right)\right)^{\frac{1}{2}}. \tag{21.5}$$

Since $\ell_\infty^m(\ell_1^d)$ is norm equivalent to ℓ_∞^{md} (albeit with a constant depending on d), the corollary can be seen to follow directly from Proposition 21.4.

Finally one needs methods 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. The following proposition from [Carl and Stephani, 1990] is very useful.

Product inequality

Proposition 21.6 Products of operators
Suppose $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are Banach spaces and $R \in \mathfrak{L}(\mathcal{A}, \mathcal{B})$ and $S \in \mathfrak{L}(\mathcal{B}, \mathcal{C})$. If $n_1, n_2 \in \mathbb{N}$ and $n \geq n_1 n_2$ then the entropy numbers of $RS : \mathcal{A} \to \mathcal{C}$ satisfy

$$\epsilon_n(RS) \leq \epsilon_{n_1}(R)\epsilon_{n_2}(S). \tag{21.6}$$

A simple variation on this standard result is the following.

Proposition 21.7
Suppose $\mathcal{A}$ and $\mathcal{B}$ are Banach spaces, $V \subseteq \mathcal{A}$ and $S \in \mathfrak{L}(\mathcal{A}, \mathcal{B})$. Then for all $n_1, n_2, n \in \mathbb{N}$ such that $n \geq n_1 n_2$,

$$\epsilon_n(S(V)) = \epsilon_n(S(V), \mathcal{B}) \leq \epsilon_{n_1}(V, \mathcal{A})\epsilon_{n_2}(S(U_{\mathcal{A}}), \mathcal{B}) = \epsilon_{n_1}(V)\epsilon_{n_2}(S). \tag{21.7}$$

This is needed if V cannot be seen as generated by a linear operator. In some cases, products of operators will not be sufficient, especially if the overall function class cannot be viewed as generated by a single linear operator itself. However, the effect of a nonlinear operator can be seen as being contained in the union of several linear ones, as the following proposition shows.

Concatenation Result

Proposition 21.8 Sets of operators
Denote by $\mathcal{W}, \mathcal{Y}$ Banach spaces, S a linear operator $S : \mathcal{W} \to \mathcal{Y}$, $\mathfrak{L}(\mathcal{W}, \mathcal{Y})$ the space of such operators and $\mathcal{S} \subset \mathfrak{L}(\mathcal{W}, \mathcal{Y})$. Consider the pseudo norm[1] on $\mathfrak{L}(\mathcal{W}, \mathcal{Y})$ induced by a set $W \subset \mathcal{W}$ in a fashion similar to the standard operator norm on $\mathfrak{L}(\mathcal{W}, \mathcal{Y})$:

$$\|S\|_W := \sup_{w \in W} \|Sw\|_{\mathcal{Y}}. \tag{21.8}$$

Let

$$\mathcal{S}W := \bigcup_{S \in \mathcal{S}} SW. \tag{21.9}$$

---

1. It is easy to check that $\|S\|_W$ is a pseudo norm. In fact we have

$$\begin{aligned}
\|S + S'\|_W &= \sup_{w \in W} \|(S + S')w\|_{\mathcal{Y}} \leq \sup_{w \in W} \|Sw\|_{\mathcal{Y}} + \sup_{w \in W} \|S'w\|_{\mathcal{Y}} = \|S\|_W + \|S'\|_W \\
\|\lambda S\|_W &= \sup_{w \in W} \|\lambda S w\|_{\mathcal{Y}} = |\lambda| \sup_{w \in W} \|Sw\|_{\mathcal{Y}} = |\lambda| \|S\|_W \\
\|S\|_W &= \sup_{w \in W} \|Sw\|_{\mathcal{Y}} \geq \|Sw^*\|_{\mathcal{Y}} \geq 0
\end{aligned}$$

Then for $n, n' \in \mathbb{N}$

$$\begin{aligned}\epsilon_{n\cdot n'}(\mathcal{S}W) &:= \epsilon_{n\cdot n'}(\mathcal{S}W, \|\cdot\|_W) := \epsilon_{n\cdot n'}(\cup_{S\in\mathcal{S}} SW, \mathcal{Y}) \\ &\leq \epsilon_n(\mathcal{S}, \|\cdot\|_W) + \sup_{S\in\mathcal{S},\mathcal{Y}} \epsilon_{n'}(SW).\end{aligned} \tag{21.10}$$

In particular for W being the unit ball, i.e., $W = U_{\mathcal{W}}$ the metric on $\mathfrak{L}(\mathcal{W}, \mathcal{Y})$ reduces to the standard operator norm and we have

$$\epsilon_{n\cdot n'}(\mathcal{S}U_{\mathcal{W}}) \leq \epsilon_n(\mathcal{S}) + \sup_{S\in\mathcal{S}} \epsilon_{n'}(S) \tag{21.11}$$

This proposition will become very useful in the case of concatenations of nonlinear estimators such as in multilayer perceptrons (see Section 21.8 for a proof). There, each subsequent layer can be represented in terms of set of operators acting on the output of the previous layer (cf. Section 21.5).

Now that all the basic ingredients have been presented we may proceed by proving bounds for the classes of functions used by practical learning algorithms.

21.3 Convex Combinations of Parametric Families

Vector Valued Convex Combination

Consider the class of functions $\mathrm{co}_\Lambda F$ obtained by an absolute convex combination of some parametric family of basis functions $F = F_\Gamma := \{f_\gamma | f_\gamma : \mathcal{X} \to \mathbb{R} \text{ with } \gamma \in \Gamma\}$:

$$\mathrm{co}_\Lambda F := \left\{ f \,\middle|\, f = \sum_i \alpha_i f_{\gamma_i} \text{ with } \alpha_i \in \mathbb{R}^d,\ \sum_i \|\alpha_i\|_{\ell_1^d} \leq \Lambda, \gamma_i \in \Gamma \right\}. \tag{21.12}$$

Observe elements $f \in \mathrm{co}_\Lambda F$ map $f: \mathcal{X} \to \mathbb{R}^d$. Recall the L_∞ norm for functions $f: \mathcal{X} \to \mathbb{R}$ is $\|f\|_{L_\infty} = \sup_{x\in\mathcal{X}} |f(x)|$. The $L_\infty(\ell_1^d)$ norm for functions $f: \mathcal{X} \to \mathbb{R}^d$ is $\|f\|_{L_\infty(\ell_1^d)} = \sup_{x\in\mathcal{X}} \|f(x)\|_{\ell_1^d}$.

For functions f_γ Lipschitz continuous in their parametrization γ with compact finite dimensional index sets Γ one obtains the following statement.

Proposition 21.9 Convex Combinations in $L_\infty(\ell_1^d)$ spaces
Denote by $\Gamma \subset \mathcal{X}$ a compact p-dimensional index set, and F_Γ the corresponding parametric family with $|f_\gamma(\mathbf{x})| \leq 1$ for all $\mathbf{x} \in \mathcal{X}$ and $f_\gamma \in F_\Gamma$. Moreover denote by $c_L(\Gamma, X)$ a Lipschitz constant satisfying

$$\sup_{\gamma,\gamma'\in\Gamma} \sup_{\mathbf{x}\in\mathcal{X}} |f_\gamma(\mathbf{x}) - f_{\gamma'}(\mathbf{x})| \leq c_L(\Gamma, X)\|\gamma - \gamma'\|. \tag{21.13}$$

Then there exists a positive constant $c(\Gamma, p, \mathcal{X}) > 0$ such that

$$\epsilon_{2^n}(\mathrm{co}_\Lambda F, L_\infty(\ell_1^d)) \leq c(\Gamma, p, \mathcal{X}) c_L(\Gamma, \mathcal{X}) \Lambda n^{-\frac{1}{pd}}. \tag{21.14}$$

Evaluation operator

Next we can bound $\epsilon_n(\mathrm{co}_\Lambda F, \ell_\infty^m(\ell_1^d))$ which corresponds to measuring the richness of $\mathrm{co}_\Lambda F$ on an arbitrary m-sample $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \subset \mathcal{X}$. For this purpose we introduce the *evaluation operator* S_X as

$$S_X : L_\infty(\ell_1^d) \to \ell_\infty^m(\ell_1^d) \tag{21.15}$$

$$S_X : f \mapsto (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)). \tag{21.16}$$

The first thing to note is that S_X is linear and has norm 1 due to the $L_\infty(\ell_1^d)$ norm. Hence we can apply Proposition 21.7 to bound $\epsilon_n(S_X(\mathrm{co}_\Lambda F), \ell_\infty^m(\ell_1^d))$ by $\epsilon_{n_1}(S_X)\epsilon_{n_2}(\mathrm{co}_\Lambda F)$ with $n_1 n_2 \leq n$. Next we use Propositions 21.4 and 21.9 to bound the terms $\epsilon_{n_1}(S_X)$ and $\epsilon_{n_2}(\mathrm{co}_\Lambda F)$ respectively.[2]

Simple Rate Bound

Proposition 21.10 Convex Combinations in $\ell_\infty^m(\ell_1^d)$ spaces The entropy number (with respect to the $\ell_\infty^m(\ell_1^d)$ metric) of the Λ-convex combination $\mathrm{co}_\Lambda F$ evaluated at m arbitrary points $X := (\mathbf{x}_1, \ldots, \mathbf{x}_m) \subset \mathcal{X}$ satisfies

$$\begin{aligned} e_n &= \epsilon_{2^n}(S_X \mathrm{co}_\Lambda F, \ell_\infty^m(\ell_1^d)) \\ &\leq \Lambda \tilde{c}(\Gamma, p, X) c_L(\Gamma, X) \inf_{n_1, n_2 \in \mathbb{N}, n_1 + n_2 \leq n} \left(n_1^{-1} \log\left(1 + \tfrac{md}{n_1}\right)\right)^{\frac{1}{2}} n_2^{-\frac{1}{pd}} \end{aligned} \tag{21.17}$$

for some constant $\tilde{c}(\Gamma, p, X) > 1$.

By setting $n_1 = n_2 = \lfloor n/2 \rfloor$ one can check that $e_n = O(n^{-\frac{1}{2}-\frac{1}{pd}})$. Since X was arbitrary, we can thus bound $\sup_{X \in \mathcal{X}^m} \mathcal{N}(\epsilon, (\mathrm{co}_\Lambda F)(X), \ell_\infty^m(\ell_1^d))$ by inverting the bound on $\epsilon_n(\mathrm{co}_\Lambda F)$. Ignoring $\log(m)$ terms, one gets

$$\log \sup_{X \in \mathcal{X}^m} \mathcal{N}(\epsilon, (\mathrm{co}_\Lambda F)(X), \ell_\infty^m(\ell_1^d)) = O\left(\epsilon^{-\frac{2pd}{pd+2}}\right) \tag{21.18}$$

For large p or d this is roughly $O(1/\epsilon^2)$ which is similar to the results obtained in [Gurvits, 1997, Bartlett, 1998] derived using the fat-shattering dimension, a version of Maurey's theorem and the generalization of the Sauer-Shelah-Vapnik-Chervonenkis lemma of Alon et al. [1997]. When $\log(m)$ factors are taken into account the above result is slightly better than those previous results.

Better Bounds via Kernels

As we will show subsequently, one can do much better by exploiting properties of kernels in a more explicit way. The reason we can do better is that we take more account of the geometry of F than just its covering numbers. The fact that information about $\mathcal{N}(\epsilon, F)$ alone can not provide tight bounds on $\mathcal{N}(\epsilon, \mathrm{co}_\Lambda F)$ has been observed previously by Talagrand [1993]. The easiest way to see that bounds such as Proposition 21.2 can not be always tight is to observe that $\mathrm{co}\,\mathrm{co}\, F = \mathrm{co}\, F$, but the bound of the proposition would not even apply in that case.

21.4 Convex Combinations of Kernels

Better bounds on ϵ_n can be obtained for convex combinations of kernels. Specifically we are interested in computing $\mathcal{N}(\epsilon, \mathrm{co}\, F)$ when $F = \{\mathbf{x} \mapsto k(\mathbf{x}, \mathbf{x}_1) | \mathbf{x}_1 \in X\}$. In order to do this we take the point of view in [Williamson et al., 1998]: the hypothesis class is considered as the image of a linear operator.

2. Note that $\epsilon_n(S_X(\mathrm{co}_\Lambda F)) = \epsilon_n((\mathrm{co}_\Lambda F)(X))$.

21.4.1 Feature Space

Generalized kernels

We use the definitions of Section 1.3.2, however with a deviation from the conditions on k imposed by Mercer's Theorem (Th. 1.16). Specifically we only require $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ to be a bounded symmetric function in its arguments (no positivity needed). Note that the requirement is similar to the one in Chapter 8.

Moreover we assume that there exists an expansion of k into the eigensystem $(\lambda_i, \psi_i(\mathbf{x}))$ of the corresponding symmetric integral operator (cf. (1.65))

$$Tf(\mathbf{x}) := \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} \tag{21.19}$$

such that (cf. (1.67))

$$k(\mathbf{x}, \mathbf{y}) = \sum_i \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}). \tag{21.20}$$

We will require that k induces a trace-class operator, i.e., that $\sum_i |\lambda_i|$ is finite, and that moreover there exists a constant C_k such that

$$\sup_{i \in \mathbb{N}} \sup_{\mathbf{x} \in \mathcal{X}} |\psi_i(\mathbf{x})| \leq C_k. \tag{21.21}$$

The latter is standard for Mercer kernels [Mercer, 1909], however for general symmetric operators this need not automatically be the case. Consequently the class of admissible functions is significantly larger than the one suitable for SV machines. For instance, B_n spline kernels (i.e., $n+1$ times convolutions of the unit interval) of arbitrary order $n \in \mathbb{N}$ can be used, whereas SV machines would only allow spline kernels of odd order. The crucial point in dealing with convex combinations of kernels is that elements of

Extending the Kernel Trick

$$\mathrm{co}_\Lambda F = \left\{ f : \mathcal{X} \to \mathbb{R}^d \,\middle|\, f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}), \alpha_i \in \mathbb{R}^d,\ \sum_i \|\alpha_i\|_{\ell_1^d} \leq \Lambda,\ \mathbf{x}_i \in \mathcal{X} \right\} \tag{21.22}$$

still can be written as a dot product in some feature space. (By definition of $\mathrm{co}_\Lambda F$ we have $\sum_i |\alpha_{ij}| \leq \Lambda_j$ with $\sum_j \Lambda_j \leq \Lambda$.) This is done by setting

$$f(\mathbf{x}) = \sum_{i=1}^m \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{21.23}$$

$$= \sum_{i=1}^m \alpha_i \sum_j \lambda_j \psi_j(\mathbf{x}_i) \psi_j(\mathbf{x}) = ((\mathbf{w}_1, \Phi(\mathbf{x})), \ldots, (\mathbf{w}_d, \Phi(\mathbf{x}))). \tag{21.24}$$

Here $\mathbf{w}_j$ and $\Phi(\mathbf{x})$ are defined as follows (for SV kernels this definition coincides with the standard form derived from Mercer's theorem (1.68)):

$$\Phi(\mathbf{x}) := \left(\sqrt{|\lambda_1|} \psi_1(\mathbf{x}), \sqrt{|\lambda_2|} \psi_2(\mathbf{x}), \ldots \right) \tag{21.25}$$

$$\mathbf{w}_j := \left(\sqrt{|\lambda_1|}\, \mathrm{sgn}\,(\lambda_1) \sum_{i=1}^m \alpha_{ij} \psi_1(\mathbf{x}_i), \sqrt{|\lambda_2|}\, \mathrm{sgn}\,(\lambda_2) \sum_{i=1}^m \alpha_{ij} \psi_2(\mathbf{x}_i), \ldots \right) \tag{21.26}$$

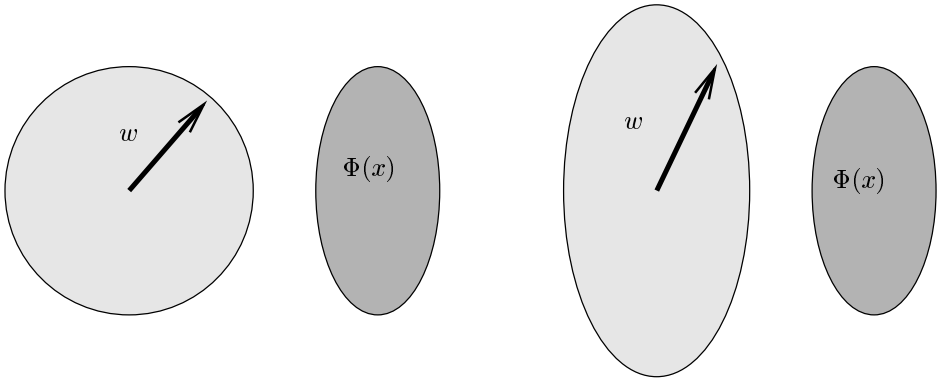

Figure 21.1 Left: In the SV case the weight vector $\mathbf{w}$ is contained in a ball of some (given) radius and the data lies inside some hyperellipsoid. Right: In the convex combination algorithms the weight vector $\mathbf{w}_j$ is contained in a scaled version of the convex hull of the data $\Phi(\mathcal{X})$, e.g., a hyperellipsoid of identical shape but different size.

It is understood that α_{ij} denotes the j-th component of α_i. From the assumptions above one can see that in analogy to [Williamson et al., 1998] again $\bigcup_{\mathbf{x}\in\mathcal{X}} \pm\Phi(\mathbf{x})$ is contained inside a box B with sidelengths $2C_k\sqrt{\lambda_i}$. Hence also $\mathbf{w}_j$ is contained in a scaled version $\Lambda_j B$ since it is a convex combination of elements from B and moreover by construction $\sum_j \Lambda_j \leq \Lambda$. This restriction of $\mathbf{w}_j$ is exactly the property we take advantage of to derive the new bounds.

21.4.2 Scaling and Evaluation Operators

Rather than dealing with parallelepipeds B we will use hyperellipsoids $\mathcal{E}$ for convenience, since the latter can be seen to have been generated by scaling the unit ball in ℓ_2 according to some operator A. With slight abuse of notation, the situation we construct is summarised in the following diagram.

Scaling Operator

$$\begin{array}{ccccc} \mathcal{X} & \xrightarrow{\Phi} & \Phi(\mathcal{X}) & \xrightarrow{A^{-1}} & U_{\ell_2} \\ & & \cap & \nwarrow^{A} & \\ & & \mathcal{E} & & \end{array} \tag{21.27}$$

That is, we seek an operator $A : \ell_2 \to \ell_2$ such that $AU_{\ell_2} =: \mathcal{E} \supseteq \Phi(\mathcal{X})$ which implies $A^{-1}\Phi(\mathcal{X}) \subseteq U_{\ell_2}$. This can be ensured by constructing A such that

$$A\colon (\mathbf{x}_j)_j \mapsto (R_A \cdot a_j \cdot \mathbf{x}_j)_j \text{ where } a_j \in \mathbb{R}^+ \tag{21.28}$$

with $R_A := C_k\|(\sqrt{|\lambda_j|}/a_j)_j\|_{\ell_2}$ where C_k is the constant from (21.21). Hence the situation (see Figure 21.1) is quite similar to the SV case [Williamson et al., 1998]. The mapped data is contained inside some hyperellipsoid. The weight vectors $\mathbf{w}_j$, however, are constrained to a ball in the SV case and to a hyperellipsoid $\mathrm{diag}(\Lambda_1, \Lambda_2, \ldots)\mathcal{E}$ of the same shape as the original data in the case of convex combinations. This means that while in SV machines capacity is allocated equally

Two Ellipsoids

along all directions, in the present case much capacity is allocated in those directions where the data is spread out a lot and little capacity where there is little spread. Since $f(\mathbf{x}) \in \mathbb{R}^d$ one has to apply the scaling operator A for each output dimension separately, i.e., one effectively has to apply the operator A_d (in a similar fashion to [Smola et al., 1999]) with

Multiple Scaling

$$A_d : \ell_2(\ell_2^d) \to \ell_2(\ell_2^d) \text{ with } A_d := \underbrace{A \times A \times \ldots \times A}_{d-\text{times}}. \tag{21.29}$$

Before carrying out the exact calculations we define an appropriate evaluation operator $S_{\Phi(X)}$. We set

Multiple Evaluation

$$S_{\Phi(X)} : \ell_2(\ell_2^d) \to \ell_\infty^m(\ell_1^d) \tag{21.30}$$

$$S_{\Phi(X)} : (\mathbf{w}_1, \ldots \mathbf{w}_d) \mapsto \begin{pmatrix} ((\Phi(\mathbf{x}_1), \mathbf{w}_1), \ldots, (\Phi(\mathbf{x}_1), \mathbf{w}_d)) \\ \vdots \\ ((\Phi(\mathbf{x}_m), \mathbf{w}_1), \ldots, (\Phi(\mathbf{x}_m), \mathbf{w}_d)) \end{pmatrix}. \tag{21.31}$$

This operator evaluates the estimate f on the dataset X, and it is precisely the entropy number of the image of $S_{\Phi(X)}$ we are seeking. The present considerations lead to the following theorem for convex combinations of kernels in analogy to the results in [Williamson et al., 1998].

21.4.3 Bounds on Entropy Numbers

Theorem 21.11 Bounds for Linear Programming Machines
Denote by k a symmetric bounded kernel, let Φ be induced via (21.25) and let $S_{\Phi(X)}$ be given by (21.30). Moreover let A be defined by (21.28) and A_d by (21.29). Then the entropy numbers of $\mathrm{co}_\Lambda F$ satisfy the following inequalities: For $n, t \in \mathbb{N}$,

$$\epsilon_n(\mathrm{co}_\Lambda F) \leq c\Lambda \|A_d\|^2 \log^{-1/2}(n) \log^{1/2}\left(1 + \tfrac{dm}{\log n}\right) \tag{21.32}$$

$$\epsilon_n(\mathrm{co}_\Lambda F) \leq \Lambda \epsilon_n(A_d^2) \tag{21.33}$$

$$\epsilon_{nt}(\mathrm{co}_\Lambda F) \leq c\Lambda \log^{-1/2}(n) \log^{1/2}\left(1 + \tfrac{dm}{\log n}\right) \epsilon_t(A_d^2) \tag{21.34}$$

where c is a constant as defined in Corollary 21.5.

This result (and also its proof) is a modified combination of the results in [Williamson et al., 1998, Smola et al., 1999]; the key difference is that the weight vector is constrained to a different set and that is why the operator A_d appears twice.

It remains to bound the entropy number of A_d. We use a slight variation on a result from [Smola et al., 1999].

Corollary 21.12 Entropy numbers for the vector valued case
Let k be a kernel which induces a trace-class integral operator and satisfies (21.21). Let A be defined by (21.28) and A_d by (21.29). Then

$$\begin{aligned}&\epsilon_n(A_d\colon \ell_2(\ell_2^d) \to \ell_2(\ell_2^d))\\ \leq &\inf_{(a_s)_s:\left(\frac{\sqrt{\lambda_s}}{a_s}\right)_s \in \ell_2} \sup_{j\in\mathbb{N}} 6C_k\sqrt{d}\left\|\left(\frac{\sqrt{\lambda_s}}{a_s}\right)_s\right\|_{\ell_2} n^{-\frac{1}{j\cdot d}}(a_1a_2\cdots a_j)^{\frac{1}{j}}.\end{aligned} \tag{21.35}$$

21.4.4 Applications to Kernel Functions

Although the above results seem rather abstruse and complex, it turns out they can be applied without too much pain. By using arguments as in [Williamson et al., 1998] (subsequently further simplified by Guo et al. [1999]) one can explicitly compute the entropy numbers of the A_d operator. The following two propositions follow immediately from their counterparts for the case of SV regularization.

Proposition 21.13 Polynomial Decay
Let k be a symmetric kernel with eigenvalues $|\lambda_j| = O(j^{-(\alpha+1/2)})$ and $\alpha > 0$. Then

$$\epsilon_n(A_d^2\colon \ell_2(\ell_2^d) \to \ell_2(\ell_2^d)) = O(\ln^{-\alpha} n). \tag{21.36}$$

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. Moreover, the dimensionality of the output does not change the rate of decay in terms of the eigenvalues λ_i except for a constant factor. Comparing the result with its SV counterpart in [Williamson et al., 1998] 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.

Since the eigenfunctions of a translation invariant kernel are the traditional Fourier bases, the eigenvalues can be determined in terms of Fourier transform coefficients. We then have:

Proposition 21.14 Polynomial Exponential Decay in $\mathbb{R}^d$
For translation invariant kernels, $k(\mathbf{x},\mathbf{x}') = k(\mathbf{x}-\mathbf{x}')$ in $\mathbb{R}^d \times \mathbb{R}^d$ with Fourier transform satisfying $\ln|F[k](\omega)| \leq O(\|\omega\|^p)$ with $p > 0$ and corresponding operator A_d one has

$$\ln \epsilon_n^{-1}(A_d^2\colon \ell_2(\ell_2^d) \to \ell_2(\ell_2^d)) = O(\ln^{\frac{p}{p+d}} n). \tag{21.37}$$

Analogous results hold for the other propositions obtained in [Williamson et al., 1998]. Note that whereas in Proposition 21.13 an improvement of the rates in n was achievable (over those in [Williamson et al., 1998]), in Proposition 21.14 no such thing happened since the bound is in terms of $\ln \epsilon_n$ instead of ϵ_n. The constants would be quite different though.

Of course the above considerations only indicate that the class of functions implemented by linear programming machines is smaller (in the sense of smaller covering numbers) than that implemented by traditional support vector machines. This affects the bound on generalization error, but does not imply that error will be smaller: for some problems traditional SV machines may achieve smaller error.

The point is that the capacity is distributed *differently* among the class of kernel expandable functions, i.e., a different structure (in the sense of structural risk minimization) 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 [Schölkopf et al., 1999b]), one should consider using such a regularizer.

Examples of kernels with rapid decay of the first eigenvalues are Gaussian RBF-kernels $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2)$ $(p = 2)$, or the "damped harmonic oscillator" kernel $k(\mathbf{x}, \mathbf{x}') = 1/(1 + \|\mathbf{x} - \mathbf{x}'\|)$ $(p \geq 1)$. Since $\epsilon_n(A_d^2)$ enters into the overall bound the overall covering numbers can be smaller than in the SV case where we have to bound $\epsilon_n(A_d)$.

21.5 Multilayer Networks

Whilst the techniques presented so far provide efficient tools for dealing with the capacity of sets generated by linear operators in some Hilbert spaces, many practical cases fail to satisfy these assumptions (e.g., multilayer perceptrons, rbf-networks, or combinations thereof). However, many of the latter ways of representing functions can be seen as generated by (nonlinear) concatenations of linear operators.

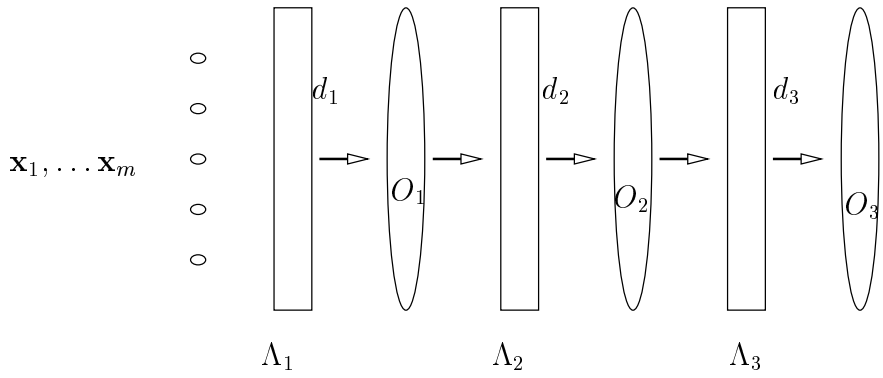

Figure 21.2 Structure of a Multilayer Network. Data is fed in on the left hand side. Each processing layer maps a (with respect to this layer) fixed input into a set of outputs via the evaluation operator S_X. Thus the possible outputs of a layer consist of the union of outputs for all different evaluation operators S_X. The output dimensionality is denoted by d_i, the size of the model class per layer by Λ_i.

Figure 21.2 depicts a multilayer network. The i first layers create, for fixed input, a *set* of outputs $O_i \subset \ell_\infty^m(\ell_1^{d_i})$ when the constraint Λ_i is imposed on the model class of the corresponding layer by requiring that the sum of the absolute values of the weights in the ith layer is Λ_i. The outputs O_i can be seen as generated by a class of operators $\mathcal{S}_i$ defined as the set of all possible evaluation operators

$$\mathcal{S}_i := \bigcup_{X \in O_{i-1}} S_{\Phi(X)} \text{ with } i > 1 \tag{21.38}$$

For $i = 1$ we set $\mathcal{S}_1 := \{S_{\Phi(X)}\}$ where X is the actual training data. Thus at each layer we have a situation as in Proposition 21.8.

We need to compute the entropy number of $\mathcal{S}_i$ in order to compute the entropy number of O_i. The following proposition uses the connection between $\epsilon_n(O_{i-1})$ and $\epsilon_n(\mathcal{S}_i)$. Moreover, in order to apply our result to Regularization Network type Multilayer Perceptrons it pays off to have a specific connection between $\mathcal{S}_i$ and O_i for Tikhonov regularizers as well (see (21.42) and (21.43)).

Proposition 21.15 Entropy numbers for classes of operators
Let $k(\mathbf{x}, \mathbf{x}')$ be a kernel with Lipschitz constant l_k, i.e.,

$$|k(\mathbf{x}, \mathbf{x}') - k(\mathbf{x}, \mathbf{x}'')| \leq l_k \|\mathbf{x}' - \mathbf{x}''\| \text{ for all } \mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \mathcal{X}, \tag{21.39}$$

where $\mathcal{X}$ is an index set with entropy number $\epsilon_n(\mathcal{X})$, and $\mathcal{S}_i$ the set of operators as defined in (21.38). Then the following bound holds: For W defined according to (21.12), i.e.,

$$W_\Lambda := \left\{ (\mathbf{w}_1, \ldots, \mathbf{w}_{d_o}) \,\middle|\, \mathbf{w}_j = \textstyle\sum_j \alpha_{ij} \Phi(\mathbf{x}_i) \text{ where } \alpha_{ij} \in \mathbb{R}, \sum_{i,j} |\alpha_{ij}| \leq \Lambda \right\} \tag{21.40}$$

and the $\ell_\infty^m(\ell_1^{d_{i-1}})$ on $\mathcal{X}$ and the $\ell_1^{d_i}$ metric on $\mathcal{Y}$ we obtain (recall the definition (21.9))

$$\epsilon_n(\mathcal{S}W_\Lambda) \leq \Lambda l_k \epsilon_n(\mathcal{X}). \tag{21.41}$$

Moreover for W defined as

$$W_\Lambda := \left\{ (\mathbf{w}_1, \ldots, \mathbf{w}_{d_o}) \,\middle|\, \sum_{i=1}^{d_o} \|\mathbf{w}_i\|^2 \leq \Lambda \right\} \tag{21.42}$$

and the mixed Euclidean metric on both $\mathcal{X}$ and $\mathcal{Y}$ respectively, i.e., $\ell_\infty^m(\ell_2^d)$, we have

$$\epsilon_n(\mathcal{S}) \leq \sqrt{2\Lambda l_k \epsilon_n(\mathcal{X})}. \tag{21.43}$$

Now we can just go and daisy-chain the separate layers and repeatedly apply Proposition 21.8. For simplicity we will only carry out this calculation for MLPs with a convexity constraint. The second case (via (21.43)) is straightforward. We obtain the following corollary.

Corollary 21.16 Entropy Numbers for Multilayer Perceptrons
For an l layer network MLP as in Figure 21.2 we obtain (set $F := F_1$)

$$\epsilon_n(\text{MLP}) \leq \sum_{i=1}^{l} \epsilon_{n_i}(F)\, l_k^{l-i} \prod_{j=i}^{l} \Lambda_j \tag{21.44}$$

where $n \geq \prod_{i=1}^{l} n_i$.

For the sake of simplicity we assumed that all layers are built in the same way. That is why only $\epsilon_n(F)$ appears in the inequality instead of one different F_i for each layer. In order to see the implications of this result we apply it to kernels satisfying the conditions of Proposition 21.14.

Corollary 21.17 MLPs from kernels with rapidly decaying spectra
For Multilayer Perceptrons built from translation invariant kernels, $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$ in $\mathbb{R}^d \times \mathbb{R}^d$ with Fourier transform satisfying $\ln |F[k](\omega)| = O(\|\omega\|^p)$ with $p > 0$, corresponding operator A_d, and ℓ_1 type convexity constraint on the weights one has

$$\ln \epsilon_n^{-1}(MLP\colon \ell_\infty^m(\ell_2^{d'}) \to \ell_\infty^m(\ell_2^d)) = O(\ln^{\frac{p}{p+d}} n) \tag{21.45}$$

This can be seen by applying Proposition 21.14 to each single layer, noticing that the finite sample size part completely dominated by the behaviour of the eigenvalues of the kernel and finally applying Corollary 21.16. There, in particular, note that since in (21.44) only the products of the entropy numbers of the individual layers appear, their effect is equal when taking the logarithm of the overall term.

Hence the asymptotic rate of growth of the covering numbers of an MLP built with such smooth kernels is the same as that for a network with a single hidden layer. Consequently from the point of the asymptotic speed of statistical convergence the class of functions of an MLP cannot be effectively more complicated than that of a single hidden layer network.

21.6 Discussion

We showed that linear programming machines do carry out a form of regularization, which is quite different from the regularization of SV machines. Furthermore, by taking advantage of the specific properties of kernels bounds on the covering numbers of the class of functions computed by such machines can be obtained which are better than those which ignore the effect of the kernel. Specifically, for some kernels (e.g., Gaussian RBF) exponentially better rates (Proposition 21.14) than those for arbitrary kernels (Prop 21.10) can be obtained — observe the ln in (21.37). In addition, we showed that one can extend the techniques to classes of functions such as those generated by multilayer RBF networks. The proofs relied on an operator theoretic viewpoint. The slower rates of growth of covering numbers obtained from the LP regularizer of course do not imply that LP machines

perform better (or worse) than SV machines; just that the "size" of the two effective classes differs and thus so do the generalization error bounds obtained via uniform convergence theorems.

In this extended summary we have limited ourselves to outlining how the rate of growth of covering numbers can be determined. For a successful learning algorithm, however, good estimates of the constants (and not only the rates) are crucial. We refer the reader to [Williamson et al., 1998, Guo et al., 1999] for the calculation of tighter bounds, by more carefully evaluating the inf and sup in the bounds on $\epsilon_n(A)$. It would be of some interest to see if such a more refined calculation could be experimentally corroborated.

21.7 Appendix: A Remark on Traditional Weight Decay

One might conjecture that a result similar to that for ordinary convex hulls could be established for q-convex combinations with $q > 1$, i.e.,

$$F_q := \left\{ f \,\middle|\, f = \sum_j \alpha_j f_{\gamma_j} \text{ with } \sum_j |\alpha_j|^q \leq 1 \right\} \tag{21.46}$$

(For the sake of simplicity we only consider $\mathcal{Y} = \mathbb{R}$ in this section.) Training large neural networks with weight decay ($q = 2$) is such a case. However, under the assumption of an infinite number of basis functions the conjecture is false. It is sufficient to show that for $q > 1$, F_q is unbounded in L_∞. Consider an infinite index set $I \subset \Gamma$ for which, for some other set M of nonzero measure and some constant $\kappa > 0$

$$f_\gamma(\mathbf{x}) \geq \kappa \quad \text{for all} \quad f_\gamma \in I, \mathbf{x} \in M. \tag{21.47}$$

An example is $f_\gamma(\mathbf{x}) = e^{-(\mathbf{x}-\gamma)^2}$ for which any compact sets I, M satisfy (21.47). Obviously

$$f(\mathbf{x}) := \sum_j \alpha_j f_{\gamma_j}(x) \geq \kappa \sum_j \alpha_j \text{ for } \alpha_j \geq 0, \gamma_j \in I, \mathbf{x} \in M. \tag{21.48}$$

For $n \in \mathbb{N}$, let $\hat{f}_n := \sum_{j=1}^n n^{-1/q} f_{\gamma_i}$. By construction, the ℓ_q^n norm of the coefficients equals 1, however $\hat{f}_n(x) \geq \kappa n^{1-1/q}$ for all $x \in M$. Thus $\lim_{n\to\infty} \|\hat{f}_n\|_{L_\infty} = \infty$ and therefore F_q contains unbounded elements for $q > 1$, which leads to infinitely large covering numbers for F_q. Thus F_q with $q > 1$ 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 settings 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 centers of the basis functions (thus of their peaks), 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.

21.8 Appendix: Proofs

Proof **(Proposition 21.7)** Suppose we have an ϵ_{n_1}, n_1 cover of V. Hence we can find $a_1, \ldots, a_{n_1} \in \mathcal{A}$ such that

$$V \subseteq \bigcup_{i=1}^{n_1} \{\epsilon_{n_1} U_{\mathcal{A}} + a_i\}. \tag{21.49}$$

Exploiting the linearity of S yields that

$$S(V) \subseteq \bigcup_{i=1}^{n_1} \{\epsilon_{n_1} S(U_{\mathcal{A}}) + S(a_i)\}. \tag{21.50}$$

Hence, constructing an ϵ_{n_2}, n_2 cover of $S(U_{\mathcal{A}})$ by $b_1, \ldots, b_{n_2} \in \mathcal{B}$ leads to an $\epsilon_{n_1}\epsilon_{n_2}, n_1 n_2$ cover of $S(V)$. Thus we get

$$S(V) \subseteq \bigcup_{i=1}^{n_1} \bigcup_{j=1}^{n_2} \{\epsilon_{n_1}\epsilon_{n_2} b_i + S a_i\} \tag{21.51}$$

which completes the proof. ■

Proof **(Proposition 21.8)** The proof works by constructing the $\epsilon_{n\cdot n'}(\mathcal{S}W)$ cover explicitly.[3] Denote by $\mathcal{S}_\epsilon = \{S_1, \ldots, S_n\} \subset \mathcal{S}$ a set achieving an ϵ cover of $\mathcal{S}$ wrt. the norm induced by W. Moreover denote by $Y_{\epsilon'}(S_i) := \{y_{i1}, \ldots, y_{in'}\}$ an ϵ' cover of $S_i W$. What we have to show is that $\bigcup_{1\le i\le n} Y_{\epsilon'}(S_i)$, which has cardinality at most $n \cdot n'$, is an $\epsilon + \epsilon'$ cover of $\mathcal{S}W$.

For any $y = S\mathbf{w} \in \mathcal{S}W$ there exists an operator S_i with $\|S\mathbf{w}' - S_i\mathbf{w}'\| \le \epsilon$ for all $\mathbf{w}' \in W$, hence in particular $\|y - S_i\mathbf{w}\| \le \epsilon$ as $\mathbf{w} \in W$. Furthermore there exists a $y_{ij} \in Y_{\epsilon'}(S_i)$ with $\|y_{ij} - S_i\mathbf{w}\| \le \epsilon'$ which leads to $\|y - y_{ij}\| \le \epsilon + \epsilon'$. Finally, such an n cover with ϵ' is always possible for all $S_i W$ since by construction $\epsilon' = \sup_{S\in\mathcal{S}} \epsilon_{n'}(S)$. ■

Proof **(Proposition 21.9)** The first step is to compute an upper bound on $\epsilon_n(F_\Gamma) = \epsilon_n(F_\Gamma, L_\infty(\ell_1^p))$ in terms of the entropy numbers of Γ. By definition we have

$$\|f_\gamma - f_{\gamma'}\| \le c_L(\Gamma, \mathcal{X}) d(\gamma, \gamma') \tag{21.52}$$

and therefore

$$\epsilon_n(F_\Gamma) \le c_L(\Gamma, \mathcal{X}) \epsilon_n(\Gamma). \tag{21.53}$$

3. A related approach was taken by Bartlett [1998] to compute the fat shattering dimension of multilayer perceptrons by exploiting a Lipschitz condition. Moreover a similar result was stated in [Haussler, 1992, Lemma 8, pg. 123].

As we are interested in the absolute convex combination in d dimensions, we need to take into account that we have to add in F_Γ for each dimension separately. Let

$$B = \left(\overbrace{(F_\Gamma, \underbrace{0, \ldots, 0}_{d-1 \text{ times}}) \cup \ldots \cup (0, \ldots, 0, F_\Gamma)}^{d \text{ times}} \right) \tag{21.54}$$

denote the base hypothesis class. Clearly if F_Γ is indexed by p-dimensions, B is indexed by pd-dimensions. From Proposition 21.3 we can obtain

$$\epsilon_n(B) \leq c(\Gamma, \mathcal{X}) \mathrm{vol}\,(\Gamma)^{\frac{1}{p}} c_L(\Gamma, \mathcal{X}) n^{-\frac{1}{pd}} \tag{21.55}$$

Now apply Proposition 21.2 to obtain

$$\epsilon_{2^n}(\mathrm{co}_\Lambda F) \leq \Lambda \rho(p) c(\Gamma, \mathcal{X}) \mathrm{vol}\,(\Gamma)^{\frac{1}{p}} c_L(\Gamma, \mathcal{X}) \left(\tfrac{1}{n}\right)^{\frac{1}{pd}}. \tag{21.56}$$

Collecting the constants into $c(\Gamma, p, \mathcal{X})$ gives the desired result. ■

Proof **(Theorem 21.11)** The diagram in Equation (21.57) indicates the line of reasoning we use for bounding $\epsilon_n(F_\Lambda)$.

$$\begin{array}{ccccccc} U_{\ell_2(\ell_2^d)} \subset \ell_2(\ell_2^d) & & \xrightarrow{\qquad T \qquad} & & & & \ell_\infty^m(\ell_1^d) \\ \Big\downarrow \Lambda & & & & & \nearrow S_{\Phi(X)} & \Big\uparrow S_{A^{-1}\Phi(X)} \\ \Lambda U_{\ell_2} \subset \ell_2(\ell_2^d) & \xrightarrow{A_d} & \Lambda A_d U_{\ell_2(\ell_2^d)} & \subset & \ell_2(\ell_2^d) & \xrightarrow{A_d} & \Lambda A_d^2 U_{\ell_2} \subset \ell_2(\ell_2^d) \end{array} \tag{21.57}$$

Here $T : \ell_2(\ell_2^d) \to \ell_\infty^m(\ell_1^d)$ depicts the linear operator corresponding to F_Λ. In order to bound $\ell_\infty^m(\ell_1^d)$ entropy numbers of the hypothesis class evaluated on an m-sample test set X, one has to bound $\epsilon_n(S_{\Phi(X)}(\Lambda A_d U_{\ell_2(\ell_2^d)}))$, since the weight vectors $(\mathbf{w}_1, \ldots, \mathbf{w}_d)$ will be contained in $\Lambda A_d U_{\ell_2(\ell_2^d)}$. Moreover we have by construction

$$S_{\Phi(X)}(\Lambda A_d U_{\ell_2(\ell_2^d)}) = S_{A^{-1}\Phi(X)}(\Lambda A_d A_d U_{\ell_2}). \tag{21.58}$$

where we used $(\Phi(x), \mathbf{w}_i) = (A^{-1}\Phi(\mathbf{x}), A\mathbf{w}_i)$ which is applicable since f can be represented as a linear functional in some feature space. Using (21.58) and Proposition 21.6 one obtains

$$\begin{array}{rcl} \epsilon_n(S_{\Phi(X)}(\Lambda A_d U_{\ell_2(\ell_2^d)})) & \leq & \Lambda \epsilon_n(S_{A^{-1}\Phi(X)} A_d^2) \\ & \leq & \inf_{n_1, n_2 \in \mathbb{N}, n_1 n_2 \leq n} \epsilon_{n_1}(S_{A^{-1}\Phi(X)}) \epsilon_{n_2}(A_d^2). \end{array} \tag{21.59}$$

Combining the factorization properties obtained above with Proposition 21.6 yields the desired results: by construction, due to the Cauchy-Schwartz inequality $\|S_{A^{-1}\Phi(X)}\| = 1$. Since $S_{A^{-1}\Phi(X)}$ is an operator mapping from a Hilbert space ℓ_2 into an ℓ_∞^m one can use Maurey's theorem (see Proposition 21.4). ■

Proof **(Proposition 21.15)** We have to show that $\sup_{\mathbf{w}\in W_\Lambda} \|S_{\mathbf{x}'}\mathbf{w} - S_{\mathbf{x}''}\mathbf{w}\| \leq \epsilon$ if $\|\mathbf{x}' - \mathbf{x}''\| \leq \epsilon'$. For the case of a linear programming regularizer one has

$$\|S_{\mathbf{x}'}\mathbf{w} - S_{\mathbf{x}''}\mathbf{w}\|_{\ell_\infty^m(\ell_1^{d_o})} = \max_{1\leq n\leq m} \sum_{i=1}^{d_o} \left| \left(\sum_j \alpha_{ij} \Phi(\mathbf{x}_j),\ \Phi(\mathbf{x}'_n) - \Phi(\mathbf{x}''_n) \right) \right| \tag{21.60}$$

$$\leq \max_{1\leq n\leq m} \sum_{i=1}^{d_o} \sum_j l_k |\alpha_{ij}|\ \|\mathbf{x}'_n - \mathbf{x}''_n\|_{\ell_1^{d_i}} \tag{21.61}$$

$$\leq l_k \Lambda \|\mathbf{x}' - \mathbf{x}''\|_{\ell_\infty^m\left(\ell_1^{d_i}\right)} \tag{21.62}$$

Assuming that there exists an ϵ cover of X with n points, this automatically generates an $l_k\Lambda\epsilon$ cover of $\mathcal{S}$ with the same number of points, which proves the theorem.

The second part can be shown in a similar manner by exploiting that

$$|(\Phi(\mathbf{x}) - \Phi(\mathbf{x}'), w)|^2 \leq \|\Phi(\mathbf{x}) - \Phi(\mathbf{x}')\|^2 \|w\|^2 \tag{21.63}$$

$$= (k(\mathbf{x},\mathbf{x}) - k(\mathbf{x},\mathbf{x}') + k(\mathbf{x}',\mathbf{x}') - k(\mathbf{x},\mathbf{x}'))\|w\|^2 \tag{21.64}$$

$$\leq 2l_k \|\mathbf{x} - \mathbf{x}'\|_{\ell_2^{d_i}} \|w\|^2 \tag{21.65}$$

Hence we have

$$\|S_{\mathbf{x}}\mathbf{w} - S_{\mathbf{x}'}\mathbf{w}\|^2_{\ell_\infty^m(\ell_2^{d_o})} = \max_{1\leq n\leq m} \sum_{i=1}^{d_o} |(\mathbf{w}_i, \Phi(\mathbf{x}'_n) - \Phi(\mathbf{x}''_n))|^2 \tag{21.66}$$

$$\leq \max_{1\leq n\leq m} \sum_{i=1}^{d_o} 2l_k \|\mathbf{x}'_n - \mathbf{x}''_n\|_{\ell_2^{d_i}} \|\mathbf{w}_i\|^2 \tag{21.67}$$

$$\leq 2l_k \Lambda \|\mathbf{x}' - \mathbf{x}''\|_{\ell_\infty^m\left(\ell_2^{d_i}\right)} \tag{21.68}$$

Again, assuming that there exists an ϵ cover of X with n points, this automatically generates a $\sqrt{2l_k\Lambda\epsilon}$ cover of $\mathcal{S}$ with the same number of points. ■

Acknowledgements

This work was supported in part by grants of the Australian Research Council, the DFG (# Ja 379/51,71,91), and the European Commission under the Working Group Nr. 27150 (NeuroCOLT2). Parts of this work were done while AS and BS were at GMD FIRST.