On the Generalisation of Soft Margin Algorithms

John Shawe-Taylor and Nello Cristianini

Abstract— **Generalisation bounds depending on the margin of a classifier are a relatively recent development. They provide an explanation of the performance of state-of-the-art learning systems such as Support Vector Machines (SVM) [1] and Adaboost [2]. The difficulty with these bounds has been either their lack of robustness or their looseness. The question of whether the generalisation of a classifier can be more tightly bounded in terms of a robust measure of the distribution of margin values has remained open for some time. The paper answers this open question in the affirmative and furthermore the analysis leads to bounds that motivate the previously heuristic soft margin SVM algorithms as well as justifying the use of the quadratic loss in neural network training algorithms. The results are extended to give bounds for the probability of failing to achieve a target accuracy in regression prediction, with a statistical analysis of Ridge Regression and Gaussian Processes as a special case. The analysis presented in the paper has also lead to new boosting algorithms described elsewhere [3].**

Keywords— **Generalisation, margin, margin distribution, neural networks, pac learning, ridge regression, soft margin, statistical learning, support vector machines**

Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England. Emails: j.shawe-taylor@cs.rhul.ac.uk and nello@support-vector.net

I. Introduction

Both theory and practice have pointed to the concept of the margin of a classifier as being central to the success of a new generation of learning algorithms. This is explicitly true of Support Vector Machines (SVMs) [4], [1], which in their simplest form implement maximal margin hyperplanes in a high dimensional feature space, but has also been shown to be the case for boosting algorithms such as Adaboost [2]. Increasing the margin has been shown to implement a capacity control through data-dependent structural risk minimisation [5], hence overcoming the apparent difficulties of using high dimensional feature spaces.

In the case of SVMs a further computatonal simplification is derived by never explicitly computing the feature vectors, but defining the space implicitly using a kernel function. In contrast Adaboost can be viewed as a sophisticated method of selecting and explicitly computing a small number of features from a vast reservoir of possibilities.

The key bounds on the generalisation typically depend on the minimal distance of the training points from the decision boundary [5], the so-called margin of the classifier. Ignoring log factors, in the realisable case the bound is proportional to the ratio between the fat-shattering dimension measured at a scale proportional to the margin and the size of the training set (see Theorem III.5). This raises concern that they are very brittle in the sense that a single training point can have a very significant influence on the bound, possibly rendering the training set inseparable.

Bartlett [6] extended the analysis to the case where a number of points closest to the boundary are treated as errors and the minimal margin of the remaining points is used. The bound obtained has the disadvantage that now the ratio of the fat-shattering dimension to training set size appears under a square root significantly weakening the power of the result asymptotically. A further problem with this approach is that there are no efficient algorithms for even obtaining a fixed ratio between the number of misclassified training points and the true minimum for linear classifiers unless $P = NP$ [7], [8]. Hence, in SVM practice, so-called soft margin versions of the algorithms are used, that attempt to achieve a (heuristic) compromise between large margin and accuracy. Note, however, that the ν-Support Vector Machine implements a different parameterisation of the soft margin algorithm that results in the parameter ν providing an upper bound on the fraction of margin errors [9].

The question whether it is possible to construct more robust estimators of the margin distribution that can be used to bound generalisation has remained open for some time [2]. The possibility that optimising the measure might lead to a polynomial time algorithm was hardly considered likely.

The current paper not only provides one possible answer to the open question by deriving a robust measure of the margin distribution, but it also shows that the measure can indeed be optimised efficiently for linear function classes – indeed by measuring the margin distribution in two natural ways the two standard SVM algorithms are derived. This derivation shows how the NP-hard problem of approximate agnostic learning for linear classifiers can be overcome by obtaining a more precise bound on the generalisation error that the classifier aimed to minimise in the first place.

Interestingly, the technique turns out to be equivalent to a manipulation of the kernel matrix, as well as being related to common statistical practices like Ridge Regression and Shrinkage methods. There is also a strong link with regularization.

Our analysis and bound make crucial use of a special loss function, that is equivalent to the slack-variables used in optimization theory and is related to the hinge loss. Our analysis was motivated by work of Freund and Schapire [10], though their technique was originally introduced by Klasner and Simon [11].

Furthermore, for neural networks the criterion derived corresponds exactly to that optimised by the back-propagation algorithm using weight decay further clarifying why this algorithm appears to generalise well when training is successful. The bound suggests variations that might be used in the error measure applied in the back-propagation algorithm.

More recent work [3] has derived a precise boosting algorithm directly from the error bounds obtained by the methods developed in this paper. This development parallels the move from hard to soft margin in SVMs since the Adaboost algorithm places an exponentially growing penalty on the margin deficit.

Finally, the results are also extended to the case of regression where they are shown to motivate SVM regression with linear and quadratic ϵ-insensitive loss functions, with ridge regression as the special case of quadratic loss and $\epsilon = 0$. Note that we will refer to this loss as the η-insensitive loss to avoid confusion with the use of ϵ to denote the misclassification probability. They provide probabilistic bounds on the likelihood of large output errors in terms of the least squares training error.

The paper is organised as follows. After summarising our results in Section II, we introduce background material in Section III before giving the key construction in Section IV. Section V derives the results for linear function classes using the 2-norm of the slack variables, which leads naturally into a discussion of the algorithmic implications in Section VI. Section VII extends the results to non-linear function classes, while Section VIII addresses regression estimates.

II. Summary of Results

The results in this section will be given in the $\tilde{O}$ notation indicating asymptotics ignoring log factors. The aim is to give the flavour of the results obtained which might otherwise be obscured by the detailed technicalities of the proofs and precise bounds obtained. We should also emphasise that as with almost all pac style bounds, there is considerable slackness in the constants. For this reason they should be regarded as giving insight into the factors affecting the generalisation performance, rather than realistic estimates for the error. As such they can be used to motivate algorithms and guide model selection.

The first case considered is that of classification using linear function classes. This therefore includes the use of kernel-based learning methods such as those used in the Support Vector Machine [1]. The kernel k provides a direct method of computing the inner product between the projection of two inputs x and x' into a high dimensional feature space via a mapping ϕ:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle.$$

Many algorithms for linear function classes create weight vectors w that can be written as a linear combination of the training feature inputs, $\phi(x_1), \ldots, \phi(x_m)$ with coefficients α_i. Hence, the evaluation of a new point x can be obtained as

$$\begin{aligned} \langle w, \phi(x) \rangle &= \left\langle \sum_i \alpha_i \phi(x_i), \phi(x) \right\rangle \\ &= \sum_i \alpha_i \langle \phi(x_i), \phi(x) \rangle = \sum_i \alpha_i k(x_i, x). \end{aligned}$$

Provided the algorithms only make use of inner products between feature vectors then there will be no need to explicitly compute in the feature space.

When used for classification the real-valued function is thresholded at 0. The margin of a point is the product of its label and the output of the underlying real-valued function. Detailed definitions are given in the next section. The η-insensitive loss function measures the loss in the case of regression by ignoring errors that are less that η and subtracting η from (the absolute value of) larger errors.

For the linear function case consider a target margin γ about the decision hyperplane and for training set S let $(\xi(\gamma)_{(x,y)})_{(x,y)\in S})$ be the vector of the amounts by which the training points fail to achieve the margin γ (these correspond to the slack variables in some formulations of the optimisation problem – for this reason we refer to them as the slack variables). We bound the probability ϵ of misclassification of a randomly chosen test point by (see Theorem V.2)

$$\epsilon \leq \tilde{O}\left(\frac{(R + \|\xi\|_2)^2}{|S|\gamma^2}\right),$$

where R is the radius of a ball about the origin which contains the support of the input probability distribution.

This bound directly motivates the optimisation of the 2-norm of the slack variables originally proposed for SVMs by Cortes and Vapnik [12] (see Section VI for details).

The results are generalized to non-linear function classes using a characterisation of their capacity at scale γ known as the fat-shattering dimension $\mathrm{fat}(\gamma)$. In this case the bound obtained has the form (see Theorem VII.11)

$$\epsilon \leq \tilde{O}\left(\frac{\mathrm{fat}(\gamma/16) + \|\xi\|_2^2/\gamma^2}{|S|}\right),$$

The fat-shattering dimension has been estimated for many function classes including single hidden layer neural networks [13], general neural networks [6] and perceptron decision trees [14]. An important feature of the fat-shattering dimension for these classes is that it does not depend on the number of parameters (for example weights in a neural network), but rather on their sizes. These measures therefore motivate a form of weight decay. Indeed one consequence of the above result is a justification of the standard error function used in back-propagation optimisation incorporating weight decay, as well as suggesting alternative error measures – see Section VII.2 for details.

The above result depends on the 2-norm of the slack variables, while many optimisation procedures use the 1-norm. We have therefore derived the following bound in terms of the 1-norm of the vector ξ (see Theorem VII.14)

$$\epsilon \leq \tilde{O}\left(\frac{\mathrm{fat}(\gamma/16) + \|\xi\|_1/\gamma}{|S|}\right),$$

which can also be applied to the linear case using a bound on the fat-shattering dimension for this class, hence motivating the box constraint algorithm (see Section VI).

Finally, the problem of estimating errors of regressors is addressed with the techniques developed. We bound the probability ϵ that for a randomly chosen test point the absolute error is greater than a given value θ. In this case we define a vector $(\xi(\gamma)_{(x,y)})_{(x,y)\in S}$ of amounts by which the error on the training examples exceeds $\eta = \theta - \gamma \geq 0$. Note that $\|\xi(\theta)\|_2^2$ is simply the least squares error on the training set. We then bound the probability ϵ by (see Theorem VIII.2)

$$\epsilon \leq \tilde{O}\left(\frac{\mathrm{fat}(\gamma/16) + \|\xi(\gamma)\|_2^2/\gamma^2}{|S|}\right).$$

These results can be used for Support Vector Regression [1] and give a criterion for choosing the optimal size $\eta = \theta - \gamma$ of the tube for the η-insensitive loss function. In addition they can be applied to standard least square regression by setting $\gamma = \theta$ to obtain the bound (see Corollary VIII.4)

$$\epsilon \leq \tilde{O}\left(\frac{\mathrm{fat}(\theta/16) + \|\xi(\theta)\|_2^2/\theta^2}{|S|}\right).$$

For the case of linear functions (in a kernel defined feature space) this reduces to a bound for (kernel) Ridge Regression.

III. Background Results

We consider learning from examples, initially of a binary classification. We denote the domain of the problem by X and a sequence of inputs by $\mathbf{x} = (x_1, \ldots, x_m) \in X^m$. A training sequence is typically denoted by

$$S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times \{-1, 1\})^m.$$

The performance of a classification function

$$h : X \longrightarrow \{-1, 1\}$$

on a training set S is measured by the empirical error

$$\mathrm{Er}_S(h) = |\{(x_i, y_i) \in S : h(x_i) \neq y_i\}|.$$

We will say that a classification function h is consistent with S if $\mathrm{Er}_S(h) = 0$, that is h correctly classifies all of the examples in S. We adopt the probably approximately correct (pac) style of analysis of generalisation. This model posits an underlying distribution P generating labeled examples. This distribution is used to generate training sets by independent sampling. It is also used to measure the generalisation error of a classifier by

$$\mathrm{er}(h) = P\{(x, y) : h(x) \neq y\}.$$

The thrust of the results in this paper relies on using real-valued functions for classification by thresholding at a fixed value. In some cases it is useful algorithmically to set the threshold value separately from the function selection. This approach can be continued into the analysis of generalisation, though both algorithmically and theoretically it is possible to simply treat the threshold as part of the function class and therefore fix the threshold at 0 once and for all. We will follow this approach in order to simplify the presentation and form of the results, though explicit treatment of the threshold could be incorporated using the techniques presented.

Hence, if we are using a real-valued function f, the corresponding classification function is $\mathrm{sign}(f)$, denoting the function giving output 1 if f has output greater than or equal to 0 and -1 otherwise. For a class $\mathcal{F}$ of real-valued functions the class $\mathrm{sign}(\mathcal{F})$ is the set of derived classification functions.

We first consider classical learning analysis which has been shown to be characterised by the VC dimension [15]

Definition III.1: Let H be a set of binary valued functions. We say that a set of points X is *shattered by* H if for all binary vectors b indexed by X, there is a function $f_b \in H$ realising b on X. The *Vapnik-Chervonenkis (VC) dimension*, $\mathrm{VCdim}(H)$, of the set H is the size of the largest shattered set, if this is finite or infinity otherwise.

The following theorem is well known in a number of different forms. We quote the result here as a bound on the generalization error rather than as a required sample size for given generalization.

Theorem III.2: [5] Let H_i, $i = 1, 2, \ldots$ be a sequence of hypothesis classes mapping X to $\{0, 1\}$ such that

$\text{VCdim}(H_i) = d_i$, and let P be a probability distribution on $X \times \{-1, 1\}$. Let p_i be any set of positive numbers satisfying $\sum_{i=1}^{\infty} p_i = 1$. With probability $1 - \delta$ over m independent examples drawn according to P, for any i for which a learner finds a consistent hypothesis h in H_i, the generalization error of h is bounded from above by

$$\epsilon(m, d, \delta) = \frac{4}{m}\left(d \ln\left(\frac{2em}{d}\right) + \ln\left(\frac{1}{p_i}\right) + \ln\left(\frac{4}{\delta}\right)\right),$$

provided $d = d_i \leq m$.

This classical result only considers the classification functions as binary valued. In many practical systems such as Support Vector Machines or Neural Networks the classification is obtained by thresholding an underlying real-valued function. In such cases the distance of the real-valued output from the threshold is known as the margin and the margin values of the training set can provide additional insight into the generalisation performance of the resulting classifier.

We first formalise the notion of the margin of an example and training set.

Definition III.3: For a real-valued function

$$f : X \to \mathbb{R},$$

we define the margin of a training example $(x, y) \in X \times \{-1, +1\}$ to be

$$m(f, (x, y)) = yf(x).$$

Note that $m(f, (x, y)) > 0$ implies correct classification. For a training set

$$S = ((x_1, y_1), \ldots, (x_m, y_m)),$$

we define the (hard) margin of S to be

$$m(f, S) = \min_{1 \leq i \leq m} \{m(f, (x_i, y_i))\}.$$

Hence, $m(f, S) > 0$ implies that $\text{sign}(f)$ is consistent with S.

We now introduce the fat-shattering dimension, a generalisation of the VC dimension that renders it sensitive to the size of the margin.

Definition III.4: Let $\mathcal{F}$ be a set of real valued functions. We say that a set of points X is *γ-shattered by $\mathcal{F}$* if there are real numbers r_x indexed by $x \in X$ such that for all binary vectors b indexed by X, there is a function $f_b \in \mathcal{F}$ satisfying

$$f_b(x) \begin{cases} \geq r_x + \gamma & \text{if } b_x = 1 \\ \leq r_x - \gamma & \text{otherwise.} \end{cases}$$

The *fat-shattering dimension* $\text{fat}_{\mathcal{F}}$ of the set $\mathcal{F}$ is a function from the positive real numbers to the integers which maps a value γ to the size of the largest γ-shattered set, if this is finite or infinity otherwise.

We will make use of the following result contained in Shawe-Taylor et al [5] which involves the fat-shattering dimension of the space of functions.

Theorem III.5: [5] Consider a real valued function class $\mathcal{F}$ having fat-shattering function bounded above by the dimension $\text{fat}_{\mathcal{F}} : \mathbb{R} \to \mathcal{N}$ which is continuous from the right. Then with probability at least $1 - \delta$ over randomly drawn training sets S of size m, a function $h = \text{sign}(f) \in \text{sign}(\mathcal{F})$ consistent with S such that $\gamma = m(f, S) > 0$ will have generalisation error bounded from above by

$$\epsilon(m, d, \delta) = \frac{2}{m}\left(d \log_2\left(\frac{8em}{d}\right) \log_2(32m) + \log_2\left(\frac{8m}{\delta}\right)\right),$$

where $d = \text{fat}_{\mathcal{F}}(\gamma/8) \leq em$.

Note how the fat-shattering dimension at scale $\gamma/8$ plays the role of the VC dimension in this bound. This result motivates the use of the term effective VC dimension for this value. In order to make use of this theorem, we must have a bound on the fat-shattering dimension and then calculate the margin of the classifier. We begin by considering bounds on the fat-shattering dimension. The first bound on the fat-shattering dimension of bounded linear functions in a finite dimensional space was obtained by Shawe-Taylor *et al.* [5]. Gurvits [16] generalised this to infinite dimensional Banach spaces. We will quote an improved version of this bound for inner product spaces which is contained in [17] (slightly adapted here for an arbitrary bound on the linear operators).

Theorem III.6: [17] Consider an inner product space and the class of linear functions $\mathcal{L}$ of norm less than or equal to B restricted to the sphere of radius R about the origin. Then the fat-shattering dimension of L can be bounded by

$$\text{fat}_{\mathcal{L}}(\gamma) \leq \left(\frac{BR}{\gamma}\right)^2.$$

In order to apply Theorems III.5 and III.6 we need to bound the radius of the sphere containing the points and the norm of the linear functionals involved.

IV. Constructing the Auxiliary Functions

As we have seen in the last section, previous margin results bound the generalisation error of a large margin classifier in terms of the fat-shattering dimension measured at a scale proportional to the hard margin. These results can be used to motivate the large margin algorithms which implement the so-called hard margin optimisation, in other words maximise the minimum margin over all the points in the training set. Frequently, the minimum margin can be greatly reduced by a small number of examples either corrupted by noise or simply representing atypical inputs. In such cases the majority of the data still exhibits a large margin, but the hard margin measure is small or even negative.

The new technique we introduce in this section allows us to shift these small number of points back to the larger margin using an auxiliary function space. The cost of performing this shift is seen in an increase in the complexity of the function class used for the classification. Hence, we are able to restore a large hard margin at the expense of additional complexity and we can therefore apply the hard margin generalisation results, using albeit more sophisticated tools for measuring the increased complexity of the function class.

The idea of performing this shift was used by Freund and Schapire [10] for the case of on-line learning algorithms. For this application it is possible to add an extra coordinate for each training example, which makes the presentation easier. Since we are undertaking a pac analysis, we cannot use a data-dependent construction, but must ensure that the input space is defined before learning begins. This fact forces us to construct an auxiliary function class that will enable us to increase the margin of individual training examples. Let X be the input space. We define the following inner product space derived from X.

Definition IV.1: Let $L(X)$ be the set of real valued functions f on X with countable support $\mathrm{supp}(f)$ (that is functions in $L(X)$ are non-zero for only countably many points). We consider two norms, the 2-norm $\|f\|_2$ is defined by

$$\|f\|_2^2 = \sum_{x \in \mathrm{supp}(f)} f(x)^2,$$

while the 1-norm is given by

$$\|f\|_1 = \sum_{x \in \mathrm{supp}(f)} |f(x)|.$$

The subclass of functions with i-norm bounded by B is denoted $L_i^B(X)$, while $L_i(X)$ is the class of functions for which the i-norm is finite. We define the inner product of two functions $f, g \in L_2(X)$, by

$$\langle f, g \rangle = \sum_{x \in \mathrm{supp}(f)} f(x)g(x).$$

Clearly the spaces $L_i(X)$ are closed under addition and multiplication by scalars.

Definition IV.2: Now for any fixed $\Delta > 0$ we define an embedding of X into the inner product space $X \times L(X)$ as follows.

$$\tau_\Delta : x \mapsto X_\Delta = (x, \Delta\delta_x),$$

where $\delta_x \in L(X)$ is defined by

$$\delta_x(y) = \begin{cases} 1; & \text{if } y = x; \\ 0; & \text{otherwise.} \end{cases}$$

We denote by $\tau_\Delta(S)$ the image of S under τ_Δ. For the special case of $\Delta = 1$ we denote τ_1 by τ.

We have defined the augmented input space, but must now describe the auxiliary functions. For a general real-valued function class $\mathcal{F}$ of functions with domain X, we define $\mathcal{F} + L_2(X)$ to be the class

$$\mathcal{F} + L_2(X) = \{(f, g) : f \in \mathcal{F}, g \in L_2(X)\}.$$

The domain of these functions is $X \times L_2(X)$, with their action defined by

$$(f, g)(x, h) = f(x) + \langle g, h \rangle.$$

Definition IV.3: For a real-valued function f on X we define

$$\xi((x, y), f, \gamma) = \max\{0, \gamma - yf(x)\}.$$

This quantity is the amount by which f fails to reach the margin γ on the point (x, y) or 0 if its margin is larger than γ. Similarly for a training set S, we define

$$D(S, f, \gamma) = \sqrt{\sum_{(x,y)\in S} \xi((x, y), f, \gamma)^2} =: \|\xi\|_2.$$

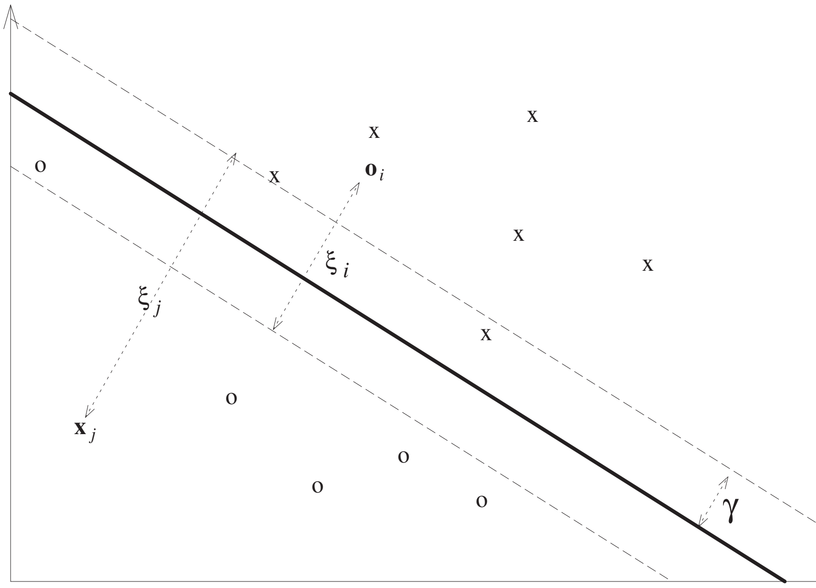

Fig. 1. Two slack variables $\xi_i = \xi((x_i, y_i), f, \gamma)$ and $\xi_j = \xi((x_j, y_j), f, \gamma)$

If we fix a target margin γ, the points with non-zero $\xi((x, y), f, \gamma)$ are those which fail to achieve a positive margin of γ (see Figure 1). Given a real-valued function f and a training set S, we now construct an auxiliary function $g_f \in L(X)$, which will ensure that (f, g_f) achieves a margin γ on S. Note that we assume throughout the paper that there are no duplicates in the training set. The function g_f depends on γ and the training set S, but we suppress this dependency to simplify the notation,

$$g_f = \frac{1}{\Delta} \sum_{(x,y)\in S} y\xi((x, y), f, \gamma)\delta_x.$$

If we now consider the margin of the function (f, g_f) applied to a training point $(\tau_\Delta(x), y) \in \tau_\Delta(S)$, we have

$$\begin{aligned} y(f, g_f)\tau_\Delta(x) &= yf(x) + \\ &\quad \frac{y}{\Delta} \sum_{(x',y')\in S} y'\xi((x', y'), f, \gamma)\langle \delta_{x'}, \Delta\delta_x \rangle \\ &= yf(x) + \xi((x, y), f, \gamma) \\ &\geq yf(x) + \gamma - yf(x) \\ &= \gamma. \end{aligned} \tag{1}$$

Furthermore if we apply the function (f, g_f) to a point $(\tau_\Delta(x), y) \notin \tau_\Delta(S)$ we observe that for $(x', y') \in S$,

$\langle \delta_{x'}, \delta_x \rangle = 0$ and so

$$\begin{aligned}\langle g_f, \delta_x \rangle &= \sum_{(x',y') \in S} y' \xi((x',y'), f, \gamma) \langle \delta_{x'}, \delta_x \rangle \\ &= 0.\end{aligned}$$

Hence, we see that the off-training set peformance of (f, g_f) satisfies

$$(f, g_f)\tau_\Delta(x) = f(x) \tag{2}$$

We have therefore shown the following lemma.

Lemma IV.4: For any training set S, real-valued function f, and target margin γ the function (f, g_f) satisfies the properties

1. $m((f, g_f), \tau_\Delta(S)) \geq \gamma$.
2. For $(x, y) \notin S$ $(f, g_f)(\tau_\Delta(x), y) = f(x)$.

Proof: The properties 1. and 2. are a direct consequence of equations (1) and (2). ■

The construction we have established in this section enables us to force a target margin γ at the cost of reinforcing the function class with the auxiliary functions in $L(X)$. The second property demonstrated in Lemma IV.4 shows that the off-training set performance of the augmented classifier exactly matches the original function. Hence, we can analyse the generalisation of f by considering how (f, g_f) performs on the training set. In the next section we first consider the case where the class of functions is linear.

V. Soft Margins for Linear Function Classes

The construction of the previous section shows how the margins can be forced to a target value of γ at the expense of additional complexity. We consider here how that complexity can be assessed in the case of a linear function class. We first treat the case where the parameter Δ controlling the embedding is fixed. In fact we wish to choose this parameter in response to the data. In order to obtain a bound over different values of Δ, it will be necessary to apply the following theorem several times.

In applying the theorems, a problem can arise if the classification given by the underlying function disagrees with the augmented function and there is a non-trivial measure on the points of disagreement. Since this can only occur on the subset of points in the training set for which $\xi((x, y), f, \gamma) \neq 0$, we always assume that the test function first checks if a test point is one of these points and if so makes appropriate adjustments to the classification given by the underlying (in this case linear) function. We use the phrase *training set filtered* to refer to this procedure. These same observations apply throughout the paper.

Theorem V.1: Fix $\Delta > 0$. Consider a fixed but unknown probability distribution on the space $X \times \{-1, 1\}$ with support in the ball of radius R about the origin in X. Then with probability $1 - \delta$ over randomly drawn training sets S of size m for all $\gamma > 0$ the generalization of a training set filtered linear classifier $\mathbf{u}$ on X with $\|\mathbf{u}\| = 1$, thresholded at 0 is bounded by

$$\begin{aligned}\epsilon(m, d, \delta) &= \frac{2}{m}\left(d \log_2\left(\frac{8em}{d}\right) \log_2(32m) \right. \\ &\qquad \left. + \log_2\left(\frac{8m}{\delta}\right)\right),\end{aligned}$$

where

$$d = \left\lfloor \frac{64.5(R^2 + \Delta^2)(1 + D(S, \mathbf{u}, \gamma)^2/\Delta^2)}{\gamma^2} \right\rfloor,$$

provided $m \geq 2/\epsilon$, $d \leq em$.

Proof: Consider the fixed mapping τ_Δ and the augmented linear function over the space $X \times L(X)$,

$$\mathbf{u}' = (\mathbf{u}, g_{\mathbf{u}}).$$

By Lemma IV.4 $\mathbf{u}'$ has margin γ on the training set $\tau_\Delta(S)$, while its action on new examples matches that of $\mathbf{u}$. Observe that since we are dealing with a class of linear functions on X, $\mathbf{u}'$ is a linear function on the space $X \times L(X)$. It follows that we can form the function

$$\hat{\mathbf{u}} = \frac{\mathbf{u}'}{\|\mathbf{u}'\|},$$

which has norm 1 and satisfies

$$m(\hat{\mathbf{u}}, S) \geq \frac{\gamma}{\|\mathbf{u}'\|} = \frac{\gamma}{\sqrt{1 + D(S, \mathbf{u}, \gamma)^2/\Delta^2}},$$

and also mimics the classification of $\mathbf{u}$ for $(x, y) \notin S$. We can therefore apply Theorems III.5 and III.6 to the training set filtered function. Note that the support for the distribution of $\tau_\Delta(x)$ is contained within a ball of radius $\sqrt{R^2 + \Delta^2}$. The theorem follows. ■

We now apply this theorem several times to allow a choice of Δ which approximately minimises the expression for d. Note that the minimum of the expression (ignoring the constant and suppressing the denominator γ^2) is $(R + D)^2$, attained when $\Delta = \sqrt{RD}$.

Theorem V.2: Consider a fixed but unknown probability distribution on the space $X \times \{-1, 1\}$ with support in the ball of radius R about the origin in X. Then with probability $1 - \delta$ over randomly drawn training sets S of size m for all $\gamma > 0$ such that $\xi((x, y), \mathbf{u}, \gamma) = 0$, for some $(x, y) \in S$, the generalisation of a unit norm training set filtered linear classifier $\mathbf{u}$ on X thresholded at 0 is bounded by

$$\epsilon(m, d, \delta) = \frac{2}{m}\left(d \log_2\left(\frac{8em}{k}\right) \log_2(32m) + \log_2\left(\frac{4m(28 + \log_2(m))}{\delta}\right)\right)$$

where

$$d = \left\lfloor \frac{65[(R + D)^2 + 0.5RD]}{\gamma^2} \right\rfloor,$$

for $D = D(S, \mathbf{u}, \gamma)$, and provided $m \geq 2/\epsilon$, $d \leq em$.

Proof: Consider a fixed set of values for Δ, $\Delta_1 = R\sqrt{2}(m - 1)^{0.25}$, $\Delta_{i+1} = \Delta_i/\sqrt{2}$, for $i = 1, \ldots, t$, where t satisfies, $\sqrt{2}R/64 \geq \Delta_t > R/64$. Hence, $t \leq 2\log_2(128m^{0.25}) = 0.5(28 + \log_2(m))$. We apply Theorem V.1 for each of these values of Δ, using $\delta' = \delta/t$ in each application. For a given value of γ and $D = D(S, \mathbf{u}, \gamma)$, it is easy to check that the value of d is minimal for $\Delta = \sqrt{RD}$ and is monotonically decreasing for smaller values of Δ and monotonically increasing for larger values. Note that $\sqrt{RD} \leq R\sqrt{2\sqrt{m-1}} = R\sqrt{2}(m-1)^{0.25}$, as the largest absolute difference in the values of the linear function on two training points is $2R$ and since $d((x,y), \mathbf{u}, \gamma) = 0$, for some $(x,y) \in S$, we must have $d((x',y'), \mathbf{u}, \gamma) \leq 2R$, for all $(x',y') \in S$. Hence, we can find a value of Δ_i satisfying $\sqrt{RD}/\sqrt{2} \leq \Delta_i \leq \sqrt{RD}$, provided $\sqrt{RD} \geq \sqrt{2}R/64$. In this case the value of the expression

$$(R^2 + \Delta^2)(1 + D(S, \mathbf{u}, \gamma)^2/\Delta^2)$$

at the value Δ_i will be upper bounded by its value at $\Delta = \sqrt{RD}/\sqrt{2}$. A routine calculation confirms that for this value of Δ, the expression is equal to $(R+D)^2 + 0.5RD$. Now suppose $\sqrt{RD} < \sqrt{2}R/64$. In this case we will show that

$$(R^2 + \Delta_t^2)(1 + D^2/\Delta_t^2) \leq \frac{130}{129}\left\{(R+D)^2 + 0.5RD\right\},$$

so that the application of Theorem V.1 with $\Delta = \Delta_t$ covers this case once the constant 64.5 is replaced by 65. Recall that $\sqrt{2}R/64 \geq \Delta_t > R/64$ and note that $\sqrt{D/R} < \sqrt{2}/64$. We therefore have

$$\begin{aligned}
(R^2 + \Delta_t^2)(1 + D^2/\Delta_t^2) &\leq R^2(1 + 2/64^2)(1 + 64^2 D^2/R^2) \\
&\leq R^2\left(1 + \frac{1}{2048}\right)\left(1 + \frac{64^2 4}{64^4}\right) \\
&\leq R^2\left(1 + \frac{1}{2048}\right)\left(1 + \frac{1}{1024}\right) \\
&< \frac{130}{129}R^2 \\
&\leq \frac{130}{129}\left\{(R+D)^2 + 0.5RD\right\}
\end{aligned}$$

as required. The result follows. ∎

VI. Algorithmics

The theory developed in the last two sections provides a way to transform a non linearly separable problem into a separable one by mapping the data to a higher dimensional space, a technique that can be viewed as using a kernel in a similar way to Support Vector Machines.

Is it possible to give an effective algorithm for learning a large margin hyperplane in this augmented space? This would automatically give an algorithm for optimizing the margin distribution in the original space. It turns out that not only is the answer yes, but also that such an algorithm already exists.

The mapping τ_Δ defined in Section IV when applied to a linear space implicitly defines a kernel as follows

$$\begin{aligned}
k_\Delta(x,x') &= \langle \tau_\Delta(x), \tau_\Delta(x') \rangle \\
&= \langle (x, \Delta\delta_x), (x', \Delta\delta_{x'}) \rangle \\
&= \langle x, x' \rangle + \Delta^2 \langle \delta_x, \delta_{x'} \rangle \\
&= \langle x, x' \rangle + \Delta^2 \delta_x(x')
\end{aligned}$$

Note that for the analysis of the algorithms we are allowing a variable threshold b in order to match more closely the definitions in standard usage. By using this kernel, the decision function of the SV machine becomes

$$\begin{aligned}
f(x) &= \sum_{i=1}^{m} \alpha_i y_i k_\Delta(x, x_i) + b \\
&= \sum_{i=1}^{m} \alpha_i y_i \left[\langle x, x_i \rangle + \Delta^2 \delta_{x_i}(x)\right] + b.
\end{aligned}$$

If we begin with a kernel $k(x,x')$ that defines an implicit feature map ϕ satisfying $k(x,x') = \langle \phi(x), \phi(x') \rangle$, we need only consider applying the map τ_Δ to $\phi(x)$ to obtain the new kernel

$$\begin{aligned}
k_\Delta(x,x') &= \langle \tau_\Delta(\phi(x)), \tau_\Delta(\phi(x')) \rangle \\
&= \langle (\phi(x), \Delta\delta_x), (\phi(x'), \Delta\delta_{x'}) \rangle \\
&= k(x,x') + \Delta^2 \langle \delta_x, \delta_{x'} \rangle \\
&= k(x,x') + \Delta^2 \delta_x(x')
\end{aligned}$$

Hence to optimise the new bound, we need only replace the kernel matrix K with the matrix, $K' \leftarrow K + \Delta^2 I$, which has a heavier diagonal, which is equivalent to applying the hard margin algorithm after adding $\Delta^2 I$ to the covariance matrix.

This technique is well known in classical statistics, where it is sometimes called the "shrinkage method" (see Ripley [18]). In the context of regression with squared loss it is better known as Ridge Regression (see [19] for an exposition of dual Ridge Regression), and in this case leads to a form of weight decay. It is a regularization technique in the sense of Tikhonov [20]. Another way to describe it, is that it reduces the number of effective free parameters, as measured by the trace of K. Note finally that from an algorithmic point of view these kernels still give a positive definite matrix, in fact a better conditioned one, though one that may lead to less sparse solutions.

Using the kernel $K + \Delta^2 I$ is equivalent to solving the soft margin problem for the case $\sigma = 2$, as stated by Cortes and Vapnik [12], minimise $\langle \mathbf{u}, \mathbf{u} \rangle + C\sum_{i=1}^{m} \xi_i^2$ subject to $y_j[\langle \mathbf{u}, x_j \rangle - b] \geq 1 - \xi_j$ and $\xi_j \geq 0$. The solution obtained is

$$\begin{aligned}
L(\alpha) &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \frac{1}{2C}\sum_{i=}
\end{aligned}$$

$$\text{subject to the constraint} \quad : \quad \sum_{i=1}^{m} \alpha_i y_i = 0,$$

which makes clear how the trade-off parameter C in their formulation is related to the kernel parameter Δ, namely

$$C = \frac{1}{\Delta^2}.$$

Note that this approach to handling non-separability goes back to Smith [21], with Bennett and Mangasarian [22] giving essentially the same formulation as Cortes and Vapnik [12], but with a different optimisation of the function class.

The expression also shows how moving to the soft margin ensures separability of the data, since both primal and dual problems are feasible. The soft margin has introduced a type of weight decay factor on the dual variables.

The analysis we have performed so far is applicable to the case of $\sigma = 2$ in the terminology of Cortes and Vapnik [12]. Though this approach has been extensively used, Cortes and Vapnik favoured setting $\sigma = 1$ arguing that it is closer to the minimisation of the training error that results from taking $\sigma = 0$. This leads to the so-called 1-norm optimisation problem

$$\begin{array}{ll} \text{minimise}_{\xi,\mathbf{u},b} & \langle \mathbf{u},\mathbf{u}\rangle + C\sum_{i=1}^{\ell}\xi_i \\ \text{subject to} & y_i\left(\langle \mathbf{u}, x_i\rangle + b\right) \geq 1-\xi_i,\, i=1,\ldots,\ell, \\ & \xi_i \geq 0,\, i=1,\ldots,\ell. \end{array} \tag{3}$$

The dual of this problem is maximisation of the Lagrangian

$$L = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle$$

subject to the constraints

$$\begin{aligned} \sum_{i=1}^{\ell}\alpha_i y_i &= 0 \\ 0 \leq \alpha_i \leq C, & \qquad i=1,\ldots,\ell. \end{aligned}$$

The second set of constraints has resulted in the method being known as the box constraint algorithm. It again shows how non-separability has been overcome since the dual feasible region is now bounded ensuring the existence of an optimal solution. In contrast to the weight decay style of constraint introduced by the 2-norm criterion, the dual variables are now restricted to a finite region.

Viewing the two primal objective functions the change in the loss function is evident. The trade off parameter C controls the relative importance given to controlling the loss as opposed to regularising the function. The cases considered so far have all been linear function classes using 2-norm regularisation giving rise to the 2-norm of the weight vector in the primal objective.

The next section will further develop the techniques we have introduced in order to bound the generalisation in terms of quantities optimised by the box constraint algorithm as well as extending the results beyond 2-norm regularisation and beyond linear function classes.

An example applying the approach when using the 1-norm of the dual variables as a regulariser is given in [23].

VII. Non-linear Function Spaces

A. Further Background Results

In order to develop the theory for the case of non-linear function classes we must introduce some of the details of the large margin proof techniques. The first we need is the concept of covering numbers – this is used to replace an infinite function class by a finite set of functions characterising its performance to a given accuracy.

Definition VII.1: Let (X, d) be a (pseudo-) metric space, let A be a subset of X and $\epsilon > 0$. A set $B \subseteq X$ is an *ϵ-cover* for A if, for every $a \in A$, there exists $b \in B$ such that $d(a,b) \leq \epsilon$. The *ϵ-covering number* of A, $\mathcal{N}_d(\epsilon, A)$, is the minimal cardinality of an ϵ-cover for A (if there is no such finite cover then it is defined to be ∞). We will say the cover is proper if $B \subseteq A$.

Note that we have used less than or equal to in the definition of a cover. This is somewhat unconventional, but will not change the bounds we use. It does, however, prove technically useful in the proofs. The idea is that B should be finite but approximate all of A with respect to the pseudometric d. The pseudometric we consider is the l^∞ distance over a finite sample $\mathbf{x} = (x_1, \ldots, x_m)$ in the space of functions,

$$d_{\mathbf{x}}(f,g) = \max_{1\leq i\leq m} |f(x_i) - g(x_i)|.$$

We write $\mathcal{N}(\epsilon, \mathcal{F}, \mathbf{x})$ for $\mathcal{N}_{d_{\mathbf{x}}}(\epsilon, \mathcal{F})$. For a training set

$$S = ((x_1,y_1),\ldots,(x_m,y_m)),$$

we will also denote the covering numbers for the sequence of inputs, $\mathbf{x} = (x_1, \ldots, x_m)$ by $\mathcal{N}(\epsilon, \mathcal{F}, S) = \mathcal{N}(\epsilon, \mathcal{F}, \mathbf{x})$. We will consider the covers to be chosen from the set of all functions with the same domain as $\mathcal{F}$ and range the reals.

We now quote a lemma from [5] which follows immediately from a result of Alon *et al.* [24].

Corollary VII.2: [5] Let $\mathcal{F}$ be a class of functions $X \to [a,b]$ and P a distribution over X. Choose $0 < \epsilon < 1$ and let $d = \mathrm{fat}_{\mathcal{F}}(\epsilon/4)$. Then

$$\sup_{\mathbf{x}\in X^m} \mathcal{N}(\epsilon, \mathcal{F}, \mathbf{x}) \leq 2\left(\frac{4m(b-a)^2}{\epsilon^2}\right)^{d\log_2(2em(b-a)/(d\epsilon))}.$$

For a monotonic function $f(\gamma)$ we define

$$f(\gamma^-) = \lim_{\alpha\to 0^+} f(\gamma - \alpha),$$

that is the left limit of f at γ.

Note that the minimal cardinality of an ϵ-cover is a monotonically decreasing function of ϵ, as is the fat-shattering dimension as a function of γ. Hence, we can write $\mathcal{N}(\gamma^-, \mathcal{F}, \mathbf{x})$ for the limit of the covering number as γ' tends to γ from below.

Definition VII.3: We say that a class of functions $\mathcal{F}$ is *sturdy* if for all sequences of inputs $\mathbf{x} = (x_1, \ldots, x_m)$ its image under the multiple evaluation map

$$\tilde{\mathbf{x}}_{\mathcal{F}}: \mathcal{F} \longrightarrow \mathbb{R}^m, \quad \text{defined by} \quad \tilde{\mathbf{x}}_{\mathcal{F}}: f \mapsto (f(x_1), \ldots, f(x_m))$$

is a compact subset of $\mathbb{R}^m$.

Note that this definition differs slightly from that introduced in [25]. The current definition is more general, but at the same time simplifies the proof of the required properties.

Lemma VII.4: Let $\mathcal{F}$ be a sturdy class of functions. Then for each $N \in \mathbb{N}$ and any fixed sequence $\mathbf{x} \in X^m$, the infimum $\gamma_N = \inf\{\gamma | \mathcal{N}(\gamma, \mathcal{F}, \mathbf{x}) \leq N\}$, is attained.

Proof: The straightforward proof follows exactly the proof of Lemma 2.6 of [25]. ∎

We will make use of the following lemma, which in the form below is due to Vapnik [26, page 168].

Lemma VII.5: Let X be a set and S a system of sets on X, and P a probability measure on X. For $\mathbf{x} \in X^m$ and $A \in S$, define $\nu_{\mathbf{x}}(A) := |\mathbf{x} \cap A|/m$. If $m > 2/\epsilon$, then

$$\begin{aligned} P^m\{\mathbf{x} \quad &: \quad \sup_{A \in S} |\nu_{\mathbf{x}}(A) - P(A)| > \epsilon\} \\ &\leq \quad 2P^{2m}\left\{\mathbf{xy}\colon \sup_{A \in S} |\nu_{\mathbf{x}}(A) - \nu_{\mathbf{y}}(A)| > \epsilon/2\right\}. \end{aligned}$$

The following two results are essentially quoted from [5] but they have been reformulated here in terms of the covering numbers involved. The difference will be apparent if Theorem VII.7 is compared with Theorem III.5 quoted in Section III.

Lemma VII.6: Suppose $\mathcal{F}$ is a sturdy set of functions that map from X to $\mathbb{R}$. Then for any distribution P on X, and any $k \in \mathbb{N}$ and any $\theta \in \mathbb{R}$

$$P^{2m}\Bigg\{\mathbf{xy}\colon \exists f \in \mathcal{F}, r = \max_j\{f(x_j)\}, 2\gamma < \theta - r, \lceil \log_2(\mathcal{N}(\gamma, \mathcal{F}, \mathbf{xy}))\rceil \leq k, \\ \frac{1}{m}|\{i | f(y_i) \geq \theta\}| \geq \epsilon(m,k,\delta) \Bigg\} < \delta,$$

where $\epsilon(m,k,\delta) = \frac{1}{m}(k + \log_2 \frac{2}{\delta})$.

Proof: We have omitted the detailed proof since it is essentially the same as the corresponding proof in [5] with the simplification that Corollary VII.2 is not required and that Lemma VII.4 ensures we can find a γ_k cover where

$$\gamma_k = \inf\{\gamma | \mathcal{N}(\gamma, \mathcal{F}, \mathbf{xy}) \leq 2^k\}$$

which can be used for all γ satisfying $\lceil \log_2(\mathcal{N}(\gamma, \mathcal{F}, \mathbf{xy}))\rceil \leq k$. Note also that an inequality is required $2\gamma < \theta - r$, as we have coverings using closed rather than open balls. ∎

The next result is couched in terms of a bound on the covering numbers in order to make explicit the fact that all applications of these results make use of such bounds and to avoid using the limits implicit in the argument γ^-. This does not have any implications for the tightness of the result.

Theorem VII.7: Consider a sturdy real valued function class $\mathcal{F}$ having a uniform bound on the covering numbers

$$\mathcal{N}(\gamma^-, \mathcal{F}, \mathbf{x}) \leq \mathcal{B}(\ell, \gamma),$$

for all $\mathbf{x} \in X^\ell$, for all ℓ. Consider a fixed but unknown probability distribution P on $X \times \{-1, 1\}$. Then with probability $1 - \delta$ over randomly drawn training sets S of size m, a function $h = \text{sign}(f) \in \text{sign}(\mathcal{F})$ consistent with S such that $\gamma = m(f, S) > 0$, will have generalisation error bounded from above by

$$\epsilon(m,k,\delta) = \frac{2}{m}\left(k + \log_2\left(\frac{2m}{\delta}\right)\right),$$

where $k = \lceil \log_2 \mathcal{B}(2m, \gamma/2)\rceil$.

Proof: Making use of Lemma VII.5 we will move to the double sample and stratify by k. By the union bound, it thus suffices to show that $\sum_{k=1}^{m/2} P^{2m}(J_k) < \delta/2$, where

$$\begin{aligned} J_k = \{SS' \quad &: \quad \exists f \in \mathcal{F}, \gamma = m(f,S) > 0, k \geq \lceil \log_2 \mathcal{B}(2m, \gamma/2)\rceil, \\ &\quad \text{Er}_{S'}(\text{sign}(f)) \geq m\epsilon(m,k,\delta)/2\}. \end{aligned}$$

(The largest value of k we need consider is $m/2$, since for larger values the bound will in any case be trivial). It is sufficient if $P^{2m}(J_k) \leq \frac{\delta}{m} = \delta'$. We will in fact work with the set

$$\begin{aligned} J'_k = \{SS' \quad &: \quad \exists f \in \mathcal{F}, \gamma' < m(f,S), k \geq \lceil \log_2 \mathcal{N}(\gamma'/2, \mathcal{F}, SS')\rceil, \\ &\quad \text{Er}_{S'}(\text{sign}(f)) \geq m\epsilon(m,k,\delta)/2\}. \end{aligned}$$

We will show that $P^{2m}(J'_k) \leq \delta'$. The result will then follow as $J_k \subseteq J'_k$. To show this consider any $SS' \in J_k$. Therefore, $\exists f \in \mathcal{F}$, such that $\gamma = m(f,S) > 0$, $k \geq \lceil \log_2 \mathcal{B}(2m, \gamma/2)\rceil$, and $\text{Er}_{S'}(\text{sign}(f)) \geq m\epsilon(m,k,\delta)/2$. By the bound on $\mathcal{B}(2m, \gamma/2)$, there exists $\gamma' < \gamma$, such that

$$\mathcal{N}(\gamma'/2, \mathcal{F}, SS') \leq \mathcal{B}(2m, \gamma/2),$$

so that we have

$$k \geq \lceil \log_2 \mathcal{B}(2m, \gamma/2)\rceil \geq \lceil \log_2 \mathcal{N}(\gamma'/2, \mathcal{F}, SS')\rceil,$$

implying that $SS' \in J'_k$, as asserted. It therefore remains to show that $P^{2m}(J'_k) \leq \delta'$.

Consider the function class $\hat{\mathcal{F}}$ acting on $\hat{X} = X \times \{-1, 1\}$ defined by

$$\hat{\mathcal{F}} = \{\hat{f}\colon f \in \mathcal{F}\}, \quad \text{where} \quad \hat{f} : (x,y) \mapsto -yf(x).$$

Hence, we have

$$\begin{aligned} J'_k \subseteq \Big\{SS' \quad &: \quad \exists \hat{f} \in \hat{\mathcal{F}}, r = \max\{\hat{f}(x,y)\colon (x,y) \in S\}, \gamma' < -r, \\ &k \geq \lceil \log \mathcal{N}(\gamma'/2, \hat{\mathcal{F}}, SS')\rceil, \\ &\Big|\{(x,y) \in S'\colon \hat{f}(x,y) \geq 0\}\Big| \geq m\epsilon(m,k,\delta)/2\Big\} \end{aligned}$$

using the fact that $\mathcal{N}(\gamma'/2, \hat{\mathcal{F}}, SS') = \mathcal{N}(\gamma'/2, \mathcal{F}, SS')$. Replacing γ by $\gamma'/2$ and setting $\theta = 0$ in Lemma VII.6 we obtain $P^{2m}(J'_k) \leq \delta'$, for

$$\epsilon(m,k,\delta) = \frac{2}{m}\left(k + \log\left(\frac{2}{\delta'}\right)\right),$$

as required. Note that the condition of Lemma VII.5 are satisfied by ϵ and m. ∎

B. Margin distribution and fat-shattering

In this section we will generalise the results of Section IV to function classes for which a bound on their fat-shattering dimension is known. The basic trick is to bound the covering numbers of the sum of two function classes in terms of the covering numbers of the individual classes. If $\mathcal{F}$ and $\mathcal{G}$ are real valued function classes defined on a domain X we denote by $\mathcal{F}+\mathcal{G}$ the function class

$$\mathcal{F}+\mathcal{G}=\{f+g|f\in\mathcal{F},g\in\mathcal{G}\}.$$

Lemma VII.8: Let $\mathcal{F}$ and $\mathcal{G}$ be two real valued function classes both defined on a domain X. Then we can bound the cardinality of a minimal γ cover of $\mathcal{F}+\mathcal{G}$ by

$$\mathcal{N}(\gamma,\mathcal{F}+\mathcal{G},\mathbf{x})\leq\mathcal{N}(\gamma/2,\mathcal{F},\mathbf{x})\mathcal{N}(\gamma/2,\mathcal{G},\mathbf{x}).$$

Proof: Fix $\eta\in(0,\gamma)$ and let B (respectively C) be a minimal η (respectively $\gamma-\eta$) cover of $\mathcal{F}$ (respectively $\mathcal{G}$) in the $d_{\mathbf{x}}$ metric. Consider the set of functions $B+C$. For any $f+g\in\mathcal{F}+\mathcal{G}$, there is an $f_i\in B$ within η of f in the $d_{\mathbf{x}}$ metric and a $g_j\in C$ within $\gamma-\eta$ of g in the same metric. For $x\in\mathbf{x}$,

$$\begin{aligned}|(f+g)(x)-(f_i+g_j)(x)| &\leq |f(x)-f_i(x)|+|g(x)-g_j(x)| &(4)\\ &\leq \eta+\gamma-\eta=\gamma. &(5)\end{aligned}$$

Hence, $B+C$ forms a γ cover of $\mathcal{F}+\mathcal{G}$. Since

$$|B+C|\leq\mathcal{N}(\eta,\mathcal{F},\mathbf{x})\mathcal{N}(\gamma-\eta,\mathcal{G},\mathbf{x}),$$

the result follows by setting $\eta=\gamma/2$. ∎

Before proceeding we need a further technical lemma to show that the property of sturdiness is preserved under the addition operator.

Lemma VII.9: Let $\mathcal{F}$ and $\mathcal{G}$ be sturdy real valued function classes. Then $\mathcal{F}+\mathcal{G}$ is also sturdy.

Proof: Consider $\mathbf{x}\in X^m$. By the sturdiness of $\mathcal{F}$, $\tilde{\mathbf{x}}_{\mathcal{F}}(\mathcal{F})$ is a compact subset of $\mathbb{R}^m$ as is $\tilde{\mathbf{x}}_{\mathcal{G}}(\mathcal{G})$. Note that

$$\tilde{\mathbf{x}}_{\mathcal{F}+\mathcal{G}}(\mathcal{F}+\mathcal{G})=\tilde{\mathbf{x}}_{\mathcal{F}}(\mathcal{F})+\tilde{\mathbf{x}}_{\mathcal{G}}(\mathcal{G}),$$

where the addition of two sets A and B of real vectors is defined

$$A+B=\{a+b|a\in A,b\in B\}.$$

Since, $\tilde{\mathbf{x}}_{\mathcal{F}}(\mathcal{F})\times\tilde{\mathbf{x}}_{\mathcal{G}}(\mathcal{G})$ is a compact set of $\mathbb{R}^2$ and $+$ is a continuous function from $\mathbb{R}^2$ to $\mathbb{R}$, we have that $\tilde{\mathbf{x}}_{\mathcal{F}}(\mathcal{F})+\tilde{\mathbf{x}}_{\mathcal{G}}(\mathcal{G})$ is the image of a compact set under $+$ and is therefore also compact. ∎

Recall the definition of the auxiliary function space given in Definition IV and the mapping $\tau=\tau_1$ given in Definition IV.2. We make use of this same construction in the following proposition. Hence, for $f\in\mathcal{F}$, $g_f\in L(X)$ is defined with $\Delta=1$.

Proposition VII.10: Let $\mathcal{F}$ be a sturdy class of real-valued functions having a uniform bound on the covering numbers

$$\mathcal{N}(\gamma^-,\mathcal{F},\mathbf{x})\leq\mathcal{B}(\ell,\gamma),$$

for all $\mathbf{x}\in X^\ell$, for all ℓ. Let $\mathcal{G}$ be a sturdy subset of $L(X)$ with the uniform bound on the covering numbers,

$$\mathcal{N}(\gamma^-,\mathcal{G},\mathbf{x})\leq\mathcal{A}(\ell,\gamma),$$

for $\mathbf{x}\in\Gamma^\ell$, where $\Gamma=\{\delta_x|x\in X\}$. Consider a fixed but unknown probability distribution on the space $X\times\{-1,1\}$. Then with probability $1-\delta$ over randomly drawn training sets S of size m for all $\gamma>0$ the generalization of a training set filtered function $\mathrm{sign}(f)\in\mathrm{sign}(\mathcal{F})$ satisfying $g_f\in\mathcal{G}$ is bounded by

$$\epsilon(m,k,\delta)=\frac{2}{m}\left(k+\log_2\left(\frac{2m}{\delta}\right)\right),$$

where

$$k=\lceil\log_2\mathcal{B}(2m,\gamma/4)+\log_2\mathcal{A}(2m,\gamma/4)\rceil,$$

provided $m\geq 2/\epsilon$.

Proof: Consider the fixed mapping $\tau=\tau_1$. By Lemma IV.4, we have

1. $m((f,g_f),\tau(S))\geq\gamma$.
2. For $(x,y)\notin S$ $(f,g_f)(\tau(x),y)=f(x)$.

Hence, the off training set behaviour of the classifier f can be characterised by the behaviour of $f+g_f$, while $f+g_f$ is a large margin classifier in the space $X\times L(X)$. In order to bound the generalization error we will apply Theorem VII.7 for $\mathcal{F}+\mathcal{G}$, which gives a bound in terms of the covering numbers. These we will bound using Lemma VII.8. The space $\mathcal{F}+\mathcal{G}$ is sturdy by Lemma VII.9, since both $\mathcal{F}$ and $\mathcal{G}$ are. In this case we obtain the following bound on the covering numbers,

$$\begin{aligned}\lim_{\alpha\to0^+} &\log_2\left(\mathcal{N}((\gamma-\alpha)/2,\mathcal{F}+\mathcal{G},SS')\right)\\ &\leq \lim_{\alpha\to0^+}\log_2\left(\mathcal{N}((\gamma-\alpha)/4,\mathcal{F},SS')\right)\\ &\quad+\lim_{\alpha\to0^+}\log_2\left(\mathcal{N}((\gamma-\alpha)/4,\mathcal{G},SS')\right)\\ &\leq \log_2(\mathcal{B}(2m,\gamma/4))+\log_2(\mathcal{A}(2m,\gamma/4)),\end{aligned}$$

as required. ∎

Proposition VII.10 gives a general framework for deriving margin distribution generalisation bounds for general function classes using different bounds on the slack variables. The next theorem considers a function class with bounded fat-shattering dimension, and combines this with the 2-norm bound on the slack variables. We will see that this combination is applicable to the back-propagation algorithm training of neural networks when the quadratic loss is used for the error function.

Theorem VII.11: Let $\mathcal{F}$ be a sturdy class of real-valued functions with range $[-a,a]$ and fat-shattering dimension bounded by $\mathrm{fat}_{\mathcal{F}}(\gamma)$. Fix a scaling of the output range $\kappa\in\mathbb{R}^+$. Consider a fixed but unknown probability distribution on the space $X\times\{-1,1\}$. Then with probability $1-\delta$ over randomly drawn training sets S of size m for all $a\geq$

$\gamma > 0$ the generalisation of a training set filtered function $\mathrm{sign}(f) \in \mathrm{sign}(\mathcal{F})$ is bounded by $\epsilon(m, d, \delta)$ defined as

$$\frac{2}{m}\left(d \log_2\left(256m\,(c/\gamma)^2\right) \log_2\left(16emc/\gamma\right) + \log_2\left(\frac{16m^{1.5}a}{\delta\kappa}\right)\right),$$

where $c = \max\{a, D(S, f, \gamma) + \kappa\}$ and

$$d = \left\lceil \mathrm{fat}_{\mathcal{F}}(\gamma^-/16) + \left(\frac{16(D(S,f,\gamma)+\kappa)}{\gamma}\right)^2 \right\rceil$$

provided $m \geq 2/\epsilon$.

Proof: Consider the sequence of function classes $\mathcal{G}_j = L_2^{B_j}(X)$, where $B_j = j\kappa$, for $j = 1, \ldots, \ell = 2\sqrt{m}a/\kappa$ (see Definition IV.1) where we assume κ is chosen to make ℓ a whole number. We will apply Proposition VII.10 with $\mathcal{G} = \mathcal{G}_j$ for each class G_j. Note that the image of $\mathcal{G}_j$ under any multiple evaluation map is a closed bounded subset of the reals and hence is compact. It follows that G_j is sturdy. It has range $[-B_j, B_j]$ on the space Γ. We have $B_\ell = 2\sqrt{m}a \geq D(S, f, \gamma)$, for all $f \in \mathcal{F}$ and all $\gamma \leq a$. Hence, for any value of $D = D(S, f, \gamma)$ obtained there is a value of B_j satisfying $D \leq B_j < D + \kappa$. Substituting the upper bound $D + \kappa$ for this B_j will give the result, when we use $\delta' = \delta/\ell$ and bound the covering numbers of the component function classes using Corollary VII.2 and Theorem III.6. In this case we obtain the following bounds on the covering numbers,

$$\begin{aligned}\lim_{\alpha\to 0^+} \quad & \log_2\left(\mathcal{N}((\gamma-\alpha)/4, \mathcal{F}, \mathbf{x})\right) \\ \leq \quad & 1 + d_1 \log_2\left(\frac{256ma^2}{\gamma^2}\right)\log_2\left(\frac{16ema}{d_1\gamma}\right) \\ =: \quad & \log_2(\mathcal{B}(2m, \gamma/4))\end{aligned}$$

where $d_1 = \mathrm{fat}_{\mathcal{F}}(\gamma^-/16)$, and

$$\begin{aligned}\lim_{\alpha\to 0^+} \quad & \log_2\left(\mathcal{N}((\gamma-\alpha)/4, \mathcal{G}_j, \mathbf{x})\right) \\ \leq \quad & 1 + d_2 \log_2\left(\frac{256mB_j^2}{\gamma^2}\right)\log_2\left(\frac{16emB_j}{d_2\gamma}\right) \\ =: \quad & \log_2(\mathcal{A}(2m, \gamma/4))\end{aligned}$$

where $d_2 = (16B_j/\gamma)^2$. Hence, in this case we can bound $\lceil \log_2 \mathcal{B}(2m, \gamma/4) + \log_2 \mathcal{A}(2m, \gamma/4) \rceil$ by

$$\begin{aligned}& \lceil \log_2 \mathcal{B}(2m, \gamma/4) + \log_2 \mathcal{A}(2m, \gamma/4) \rceil \\ \leq \quad & 2 + \left\lceil \mathrm{fat}_{\mathcal{F}}(\gamma^-/16) + \left(\frac{16B_j}{\gamma}\right)^2 \right\rceil \\ & \log_2(256m(b/\gamma)^2)\log_2(16emb/\gamma)\end{aligned}$$

giving the result where the 2 contributes a factor of 4 into the argument of the final log term. ∎

The theorem can be applied to a wide range of function classes for which bounds on their fat-shattering dimensions are known. For example Gurvits [16] bounds the fat-shattering dimension of single hidden layer neural networks. Bartlett extends these results to multi-layer sigmoidal neural networks [6]. Bartlett argues that neural network training algorithms are designed to enlarge the margin of the classifier and hence gives algorithms such as back-propagation a theoretical justification. The back-propagation algorithm performs gradient descent over the weights w of the quadratic loss of the neural network function $f_w : X \to (-1, 1)$ given by

$$E(w) = \sum_{i=1}^{m} (f_w(x_i) - y_i)^2.$$

If we consider a target margin of 1, this is precisely the square of the 2-norm of the slack variables

$$D(S, f_w, 1) = \|\xi\|_2 = \sqrt{E(w)}.$$

Hence, Theorem VII.11 provides a direct motivation for the back-propagation algorithm and in particular the quadratic loss function with a weight decay term to control the growth of the fat-shattering dimension. Since the bound is not proven to be tight the algorithm is still only optimising a heuristic, but one that has been proven to upper bound the error. It also suggests that different target margins could be considered. For example if we take $\gamma = 0.5$ training points with margin already greater than 0.5 will be ignored, while the loss for those with smaller margins will be evaluated as

$$E(w) = \sum_{i:|f_w(x_i)|<0.5} (f_w(x_i) - 0.5y_i)^2.$$

The theorem can of course be applied for linear function classes, using the bound on the fat-shattering dimension given in Theorem III.6. The bound obtained is worse since separately estimating the covering numbers for function class and slack variables incurs extra factors when compared to Theorem V.2.

C. 1-norm bounds on the margin distribution

We now consider altering the measure used to assess the slack variables. As mentioned above the box constraint algorithm optimises the 1-norm of the slacks. We will therefore concentrate on deriving a bound in terms of this norm, though there is no reason why other norms could not be considered. The most appropriate norm will depend on the type of noise that is affecting the data. For example the 1-norm will be more appropriate if the noise distribution has a longer tail.

Definition VII.12: For a training set S, we define

$$D'(S, f, \gamma) = \sum_{(x,y)\in S} \xi((x,y), f, \gamma) =: \|\xi\|_1.$$

The following lemma bounds the covering numbers of the relevant subset of $L(X)$ when bounding the 1-norm of the slacks. The result is a special case of a more general development given by Carl and Stephani [27].

Lemma VII.13: Consider the function class $\mathcal{L}^B$ defined as

$$\left\{\sum_{(x,y)\in S'} y a_x \delta_x : S' \text{ any finite set of labelled examples, } a_x \geq 0 \text{ and } \sum_{(x,y)\in S'} a_x \leq B\right\} \subseteq \mathcal{L}_1^B(X).$$

There exists a γ-covering $\mathcal{B}$ of $\mathcal{L}$ in the ℓ_∞^m metric with respect to $\tau(S)$ for any set of labelled points S with $|S| = m$, that has size bounded by

$$\log_2 |\mathcal{B}| \leq d \log_2 \left(\frac{e(m+d-1)}{d}\right),$$

where $d = \left\lfloor \frac{B}{2\gamma} \right\rfloor$.

Proof: First note that any points in $S' \setminus S$ have no effect on the value of the function on points in S. Hence, we can construct the cover from the points of S provided we allow $\sum_{(x,y)\in S} a_x$ to take any value in the interval $[0, B]$. We explicitly construct the covering $\mathcal{B}$ by choosing the functions

$$g_{\mathbf{i}} = \sum_{(x,y)\in S} y(2i_x + 1)\gamma\delta_x,$$

where $\mathbf{i} = (i_x)_{(x,y)\in S} \in \mathbb{N}^S$ satisfies

$$\sum_{(x,y)\in S} i_x \leq \left\lfloor \frac{B}{2\gamma} \right\rfloor.$$

To see that $\mathcal{B}$ does indeed form a cover, consider any

$$g = \sum_{(x,y)\in S} y a_x \delta_x,$$

with $a_x \geq 0$ and $\sum_{(x,y)\in S} a_x \leq B$, and choose i_x as

$$i_x = \operatorname{argmin}_i |(2i_x + 1)\gamma - a_x|.$$

Hence, $|a_x - (2i_x + 1)\gamma| \leq \gamma$ and so

$$|\langle g, \tau(x)\rangle - \langle g_{\mathbf{i}}, \tau(x)\rangle| \leq \gamma.$$

At the same time $(2i_x + 1)\gamma \leq a_x + \gamma$, implying that

$$\sum_{(x,y)\in S} 2i_x\gamma \quad \leq \quad \sum_{(x,y)\in S} a_x \leq B,$$

so that, taking into account that $i_x \in \mathbb{N}$, we have

$$\sum_{(x,y)\in S} i_x \leq \left\lfloor \frac{B}{2\gamma} \right\rfloor.$$

It remains to estimate $|\mathcal{B}|$. Consider first those elements of B for which $\sum_{(x,y)\in S} i_x = k$. There is a 1-1 correspondence between the allocations to the i_x and the choice of $m-1$ distinct boundaries between elements in a sequence of $m+k$ 1's (so as to form a subsequence for each $(x,y) \in S$). The correspondence is made with i_x being one fewer than the number of 1's in x's partition. Since we must choose the $m-1$ boundaries from among $m+k-1$ positions, the number of allocations is

$$\binom{m+k-1}{m-1} = \binom{m+k-1}{k}.$$

Hence, if we set $d = \left\lfloor \frac{B}{2\gamma} \right\rfloor$ we can bound the number of elements in $|\mathcal{B}|$ by

$$\begin{aligned}
|\mathcal{B}| &\leq \sum_{k=0}^{d} \binom{m+k-1}{k} \\
&\leq \sum_{k=0}^{d} \binom{m+d-1}{k} \\
&\leq \left(\frac{e(m+d-1)}{d}\right)^d,
\end{aligned}$$

where the last inequality follows from a similar bound to that used in the application of Sauer's Lemma. The result follows. ∎

Putting together the result of Lemma VII.13 with Proposition VII.10 gives the following generalisation bound in terms of the fat-shattering dimension of the function class and the 1-norm of the slack variables.

Theorem VII.14: Let $\mathcal{F}$ be a sturdy class of real-valued functions with range $[-a, a]$ and fat-shattering dimension bounded by $\mathrm{fat}_{\mathcal{F}}(\gamma)$. Fix a scaling of the output range $\kappa \in \mathbb{R}^+$. Consider a fixed but unknown probability distribution on the input space X. Then with probability $1-\delta$ over randomly drawn training sets S of size m for all $a > \gamma > 0$ the generalization of a training set filtered function $\mathrm{sign}(f) \in \mathrm{sign}(\mathcal{F})$ is bounded by

$$\begin{aligned}
&\epsilon(m, d_1, d_2, \delta) \\
&= \frac{2}{m}\left(d_1 \log_2\left(256m\left(\frac{a}{\gamma}\right)^2\right)\log_2\left(\frac{16ema}{\gamma}\right) + d_2 \log_2(2em)\right. \\
&\quad \left. + \log_2\left(\frac{8m^2 a}{\delta\kappa}\right)\right),
\end{aligned}$$

where

$$d_1 = \mathrm{fat}_{\mathcal{F}}(\gamma^-/16), \quad \text{and} \quad d_2 = \left\lfloor \frac{2(D'(S,f,\gamma)+\kappa)}{\gamma} \right\rfloor$$

provided $m \geq 2/\epsilon$.

Proof: Consider the sequence of function classes $\mathcal{G}_j = \mathcal{L}^{B_j}(X)$, where $B_j = j\kappa$, for $j = 1, \ldots, \ell = 2ma/\kappa$ (see Definition IV.1) where we assume κ is chosen to make ℓ a whole number. We will apply Proposition VII.10 with $\mathcal{G} = \mathcal{G}_j$ for each class G_j. Note that the image of $\mathcal{G}_j$ under any multiple evaluation map is a closed bounded subset of the reals and hence is compact. It follows that G_j is sturdy. It has range $[-B_j, B_j]$ on the space Γ. We have $B_\ell = 2ma \geq D'(S, f, \gamma)$, for all $f \in \mathcal{F}$ and all $\gamma \leq a$. Hence, for any value of $D' = D'(S, f, \gamma)$ obtained there is a value of B_j satisfying $D \leq B_j < D + \kappa$. Substituting the upper bound $D + \kappa$ for this B_j will give the result, when we use $\delta' = \delta/\ell$ and bound the covering numbers of the component function classes using Corollary VII.2, Theorem III.6 and Lemma VII.13. In this case we obtain the following bounds on the covering numbers,

$$\begin{aligned}\lim_{\alpha\to 0^+} \quad & \log_2\left(\mathcal{N}((\gamma-\alpha)/4, \mathcal{F}, SS')\right) \\ \leq \quad & 1 + d_1 \log_2\left(\frac{256ma^2}{\gamma^2}\right)\log_2\left(\frac{16ema}{d_1\gamma}\right) \\ =: \quad & \log_2(\mathcal{B}(2m, \gamma/4))\end{aligned}$$

where $d_1 = \mathrm{fat}_{\mathcal{F}}(\gamma^-/16)$, and

$$\begin{aligned}\lim_{\alpha\to 0^+} \quad & \log_2\left(\mathcal{N}((\gamma-\alpha)/4, \mathcal{G}_j, SS')\right) \\ \leq \quad & d_2 \log_2\left(\frac{e(2m-d_2-1)}{d_2}\right) \\ =: \quad & \log_2(\mathcal{A}(2m, \gamma/4))\end{aligned}$$

where $d_2 = \left\lfloor \frac{2B_j}{\gamma} \right\rfloor$. Hence, in this case we can bound $\lceil \log_2 \mathcal{B}(2m, \gamma/4) + \log_2 \mathcal{A}(2m, \gamma/4) \rceil$ by

$$\begin{aligned} & \lceil \log_2 \mathcal{B}(2m, \gamma/4) + \log_2 \mathcal{A}(2m, \gamma/4) \rceil \\ \leq \quad & 1 + \left\lfloor \frac{2B_j}{\gamma} \right\rfloor \log_2(2em) + \\ & \mathrm{fat}_{\mathcal{F}}(\gamma^-/16)\log_2(256m(a/\gamma)^2)\log_2(16ema/\gamma)\end{aligned}$$

giving the result where the 1 contributes a factor of 2 into the the argument of the final log term. ∎

If we use Theorem III.6 to bound the fat-shattering dimension of the underlying linear classifier, Theorem VII.14 is directly applicable to the box constraint algorithm for Support Vector Machines [1]. Note that in this case the target margin is 1 and the fat-shattering dimension is given by $\|w\|^2 R^2$. Hence, ignoring the logarithmic factors the quantity to be minimised to improve the generalisation is

$$\|w\|^2 + C\|\xi\|_1,$$

precisely the quantity optimised by the box constraint algorithm.

VIII. Regression

In order to apply the results of the last section to the regression case we formulate the error estimation as a classification problem. Consider a real-valued function class $\mathcal{F}$ with domain X. For $f \in \mathcal{F}$ we define the function $e(f)$ on the domain $X \times \mathbb{R}$ and hence the class $e(\mathcal{F})$,

$$\begin{aligned} e(f)(x,y) &= |f(x) - y|, \\ e(\mathcal{F}) &= \{e(f) | f \in \mathcal{F}\}. \end{aligned}$$

Note that we could use any loss function and apply the subsequent analysis to the loss function class[1]. The size of the slack variables would change as would the corresponding covering numbers at different scales resulting in different optimisation criteria and bounds.

[1] We are grateful to an anonymous referee for pointing out this natural generalisation.

We now fix a target accuracy $\theta > 0$. For a training point $(x, y) \in X \times \mathbb{R}$ we define

$$\xi((x,y), f, \gamma) = \max\{0, |f(x) - y| - (\theta - \gamma)\}.$$

This quantity is the amount by which f exceeds the error margin $\theta - \gamma$ on the point (x, y) or 0 if f is within $\theta - \gamma$ of the target value. Hence, this is the η insensitive loss measure considered by Drucker *et al.* [28] with $\eta = \theta - \gamma$. Let $g_f \in L_f(X)$ be the function

$$g_f = -\sum_{(x,y)\in S} \xi((x,y), f, \gamma)\delta_x.$$

As in the classification case when evaluating a function on a test point it is possible for there to be a disagreement between the underlying function and the augmented function if $\xi((x,y), f, \gamma) \neq 0$. We again use the phrase *training set filtered* to refer to the procedure that first checks if the test point is one of these training points and if so makes an appropriate adjustment to the the underlying function.

Proposition VIII.1: Fix $\theta \in \mathbb{R}$, $\theta > 0$. Let $\mathcal{F}$ be a sturdy class of real-valued functions having a uniform bound on the covering numbers

$$\mathcal{N}(\gamma^-, \mathcal{F}, \mathbf{x}) \leq \mathcal{B}(m, \gamma),$$

for all $\mathbf{x} \in X^m$. Let $\mathcal{G}$ be a sturdy subset of $L(X)$ with the uniform bound on the covering numbers,

$$\mathcal{N}(\gamma^-, \mathcal{G}, \mathbf{x}) \leq \mathcal{A}(m, \gamma),$$

for $\mathbf{x} \in \Gamma^m$, where $\Gamma = \{\delta_x | x \in X\}$. Consider a fixed but unknown probability distribution on the space $X \times \mathbb{R}$. Then with probability $1 - \delta$ over randomly drawn training sets S of size m for all $\gamma > 0$ the probability that a training set filtered function $f \in \mathcal{F}$ has error greater than θ on t on a randomly chosen input is bounded by

$$\epsilon(m, k, \delta) = \frac{2}{m}\left(k + \log_2\left(\frac{8m}{\delta}\right)\right),$$

where

$$k = \lceil \log_2 \mathcal{B}(2m, \gamma/4) + \log_2 \mathcal{A}(2m, \gamma/4) \rceil,$$

provided $m \geq 2/\epsilon$ and $g_{e(f)} \in \mathcal{G}$

Proof: The result follows from an application of Proposition VII.10 to the function class $e(\mathcal{F}) - \theta$, noting that we treat all training examples as negative, and hence correct classification corresponds to having error less than 0. Finally, we can bound the covering numbers

$$\mathcal{N}(\gamma, e(\mathcal{F}), \mathbf{x}) \leq \mathcal{N}(\gamma, \mathcal{F}, \mathbf{x}) \leq \mathcal{B}(m, \gamma).$$

The result follows. ∎

For a training set S, we define

$$\mathcal{D}(S,f,\gamma)=\sqrt{\sum_{(x,y)\in S}\xi((x,y),f,\gamma)^2}.$$

The above result can be used to obtain a bound in terms of the observed value of $D(S,f,\gamma)$ and the fat-shattering dimension of the function class.

Theorem VIII.2: Let $\mathcal{F}$ be a sturdy class of real-valued functions with range $[-a,a]$ and fat-shattering dimension bounded by $\mathrm{fat}_{\mathcal{F}}(\gamma)$. Fix $\theta\in\mathbb{R}$, $\theta>0$ and a scaling of the output range $\kappa\in\mathbb{R}^+$. Consider a fixed but unknown probability distribution on the space $X\times\mathbb{R}$. Then with probability $1-\delta$ over randomly drawn training sets S of size m for all γ with $\theta\geq\gamma>0$ the probability that a training set filtered function $f\in\mathcal{F}$ has error larger than θ on a randomly chosen input is bounded by $\epsilon(m,d,\delta)$ given by

$$\frac{2}{m}\left(d\log_2\left(256m\left(\frac{c}{\gamma}\right)^2\right)\log_2\left(16em\left(\frac{c}{\gamma}\right)\right)+\log_2\left(\frac{16m^{1.5}a}{\delta\kappa}\right)\right),$$

where $c=\max\{a,\mathcal{D}(S,f,\gamma)+\kappa\}$ and

$$d=\left[\mathrm{fat}_{\mathcal{F}}(\gamma^-/16)+\left(\frac{16(\mathcal{D}(S,f,\gamma)+\kappa)}{\gamma}\right)^2\right]$$

provided $m\geq 2/\epsilon$.

Proof: This follows from a direct application of Theorem VII.11. ∎

A special case of this theorem is when the function classes are linear. We present this case as a special theorem. Again by using the techniques of Section V we could improve the constants, but because the norm is no longer 1, the results are not directly applicable. We therefore present a weaker version.

Theorem VIII.3: Let $\mathcal{F}$ be a the set of linear functions with norm at most B restricted to inputs in a ball of radius R about the origin. Fix $\theta\in\mathbb{R}$, $\theta>0$ and a scaling of the output range $\kappa\in\mathbb{R}^+$. Consider a fixed but unknown probability distribution on the space $X\times\{-1,1\}$. Then with probability $1-\delta$ over randomly drawn training sets S of size m for all γ, with $\theta\geq\gamma>0$ the probability that a training set filtered function $f\in\mathcal{F}$ has error larger than θ on a randomly chosen input is bounded by $\epsilon(m,d,\delta)$ given by

$$\frac{2}{m}\left(d\log_2\left(256m\left(\frac{c}{\gamma}\right)^2\right)\log_2\left(16em\frac{c}{\gamma}\right)+\log_2\left(\frac{16m^{1.5}BR}{\delta\kappa}\right)\right),$$

where $c=\max\{BR,\mathcal{D}(S,f,\gamma)+\kappa\}$ and

$$d=\left[\left(\frac{16BR}{\gamma}\right)^2+\left(\frac{16(\mathcal{D}(S,f,\gamma)+\kappa)}{\gamma}\right)^2\right]$$

provided $m\geq 2/\epsilon$.

Proof: The range of linear functions with weight vectors bounded by B when restricted to the ball of radius R is $[-BR,BR]$. Their fat-shattering dimension is bounded by Theorem III.6. The result follows. ∎

This theorem is directly applicable to Support Vector Regression (SVR) [28], [1]. Again $\mathcal{D}(S,f,\gamma)$ is the sum of the slack variables using the $\eta=\theta-\gamma$-insensitive loss function. The SVR algorithm minimises the quantity $B^2+C\mathcal{D}^2$, hence optimising the bound of Theorem VIII.3.

Note that we obtain a generalization bound for standard least squares regression by taking $\gamma=\theta$ in Theorem VIII.2. In this case $\mathcal{D}(S,f,\theta)$ is the least squares error on the training set, while the bound gives the probability of a randomly chosen input having error greater than θ. This is summarised in the following corollary.

Corollary VIII.4: Let $\mathcal{F}$ be a sturdy class of real-valued functions with range $[-a,a]$ and fat-shattering dimension bounded by $\mathrm{fat}_{\mathcal{F}}(\gamma)$. Fix $\theta\in\mathbb{R}$, $\theta>0$ and a scaling of the output range $\kappa\in\mathbb{R}^+$. Consider a fixed but unknown probability distribution on the space $X\times\{-1,1\}$. Then with probability $1-\delta$ over randomly drawn training sets S of size m the probability that a training set filtered function $f\in\mathcal{F}$ has error larger than θ on a randomly chosen input is bounded by $\epsilon(m,d,\delta)$ given by

$$\frac{2}{m}\left(d\log_2\left(256m\left(\frac{c}{\gamma}\right)^2\right)\log_2\left(16em\frac{c}{\gamma}\right)+\log_2\left(\frac{16m^{1.5}BR}{\delta\kappa}\right)\right),$$

where $c=\max\{a,\mathcal{D}(S,f,\gamma)+\kappa\}$ and

$$d=\left[\mathrm{fat}_{\mathcal{F}}(\theta^-/16)+\left(\frac{16\left(\sqrt{\sum_{(x,y)\in S}(f(x)-y)^2}+\kappa\right)}{\theta}\right)^2\right]$$

provided $m\geq 2/\epsilon$.

For the case of linear functions this is a special case of Theorem VIII.3, namely that obtained by taking 0-insensitive loss. In this case the algorithm to optimise the bound reduces to Ridge Regression or kernel Ridge Regression [19], [1] for a kernel defined feature space.

As mentioned in the section dealing with classification we could bound the generalization in terms of other norms of the vector of slack variables

$$(\xi((x,y),f,\gamma))_{(x,y)\in S}.$$

The aim of this paper, however, is not to list all possible results, it is rather to illustrate how such results can be obtained.

Another application of these results is to choose the best η for the η-insensitive loss function for Support Vector Regression [1]. This problem has usually been solved by using a validation set, but Corollary VIII.3 could be used by choose the value of η which gives the best bound on the generalization. We assume here that a target accuracy θ has been set and we wish to minimise the probability that the error exceeds this value. The optimum will be the η which minimises

$$\frac{R^2 + \mathcal{D}(S, f_\eta, \theta - \eta)^2}{(\theta - \eta)^2},$$

where f_η is the solution obtained when using the η-insensitive loss function.

IX. Conclusions

The key contribution of this paper is a technique that enables us to transform hard margin bounds into soft margin ones, by trading in slacks of individual training points for increases in function complexity. The advantage of this exchange is that we are able to analyse function complexity more easily than taking into account individual margin errors.

The analysis for SVMs has placed the heuristic approach of Cortes and Vapnik [12] on a firm theoretical foundation. It has therefore demonstrated that by a more direct optimisation of the desired property (generalisation) of the linear classifier, the impasse of the NP-hardness of minimising the training error has been avoided and an efficient agnostic learning algorithm developed for linear classifiers. Though the algorithm is not new, the analysis has already given further insights for SVMs that have been used to tune their application to Microarray data [29].

The analysis has also placed the optimisation of the quadratic loss used in the back-propagation algorithm on a firm footing, though in this case no polynomial time algorithm is known. The paper has, however, described variations of the back-propagation algorithm suggested by the analysis and we expect that further applications of the approach will emerge as more large margin algorithms are developed.

The paper has contained only a few applications of the techniques in order to demonstrate their generality. As mentioned before the approach has already been applied to develop a soft margin boosting algorithm [3]. Standard boosting has been shown to perform gradient descent in function space optimising the negative exponential of the margins of the training points [30]. The exponential function applies something close to a hard margin penalty to individual margin errors and hence can suffer from overfitting if the training data is noisy and difficult to separate with the available weak learners. Some heuristic algorithms have been derived for soft margin boosting [31], but Bennett *et al.* [23] show how optimising a soft margin bound derived using the techniques of this paper reduces to solving a linear programme via column generation techniques. The dual variables of the linear programme perform the role of the boosting weighting of the training examples. Hence, not only does the algorithm optimise a well-founded criterion, but instead of being an approximate gradient descent method, it optimises the criterion exactly in polynomial time.

In the case of neural networks the question naturally arises as to whether there might exist a polynomial time algorithm for optimising the soft margin bound. This seems very unlikely but hardness results have always considered minimising classification error as in the case of linear classifiers, so the possibility is not as yet excluded.

From a theoretical point of view the bounds are only as tight as the results on which they depend. There has been a significant tightening of the covering number bounds for linear classifiers taking into account the structure of the training data itself [32], [33], [25] and all of these results could be combined with the techniques described here to give equivalent soft margin bounds.

Acknowledgements

The research was supported in part by the European Commission under the NeuroCOLT2 Working Group, EP27150, and the KerMIT project, IST-2001-25431, as well as by the EPSRC under grant no. GR/N08575.

References

[1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000, See also the web site **www.support-vector.net**.

[2] R. Schapire, Y. Freund, P. Bartlett, and W. Sun Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.

[3] Kristin P. Bennett, Ayhan Demiriz, and John Shawe-Taylor, "A column generation algorithm for boosting," in *Machine Learning: Proceedings of the Seventeenth International Conference, ICML'2K*, 2000.

[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed. 1992, pp. 144–152, ACM Press.

[5] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony, "Structural risk minimization over data-dependent hierarchies," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1926–1940, 1998.

[6] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[7] K. U. Höffgen, K. S. van Horn, and H. U. Simon, "Robust trainability of single neurons," *Journal of Computer and System Sciences*, vol. 50(1), pp. 114–125, 1995.

[8] S. Arora, L. Babai, J. Stern, and Z. Sweedyk, "Hardness of approximate optima in lattices, codes and linear systems," *Journal of Computer and System Sciences*, vol. 54(2), pp. 317–331, 1997.

[9] B. Schölkopf, A. Smola, R. Williamson, and P.L. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[10] Y. Freund and R.E. Schapire, "Large margin classification using the perceptron algorithm," in *Machine Learning: Proceedings of the Fifteenth International Conference*, J. Shavlik, Ed. 1998, Morgan Kaufmann.

[11] N. Klasner and H. U. Simon, "From noise-free to noise-tolerant and from on-line to batch learning," in *Proceedings of the Eighth Annual Conference on Computational Learning Theory*. July 1995, pp. 250–257, ACM Press.

[12] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[13] Leonid Gurvits and Pascal Koiran, "Approximation and learning of convex superpositions," in *Lecture Notes in Artificial Intelligence 904. Proceedings of the 2nd European Conference on Computational Learning Theory, EuroCOLT'95*. 1995, pp. 222–236, Springer-Verlag.

[14] K. Bennett, N. Cristianini, J. Shawe-Taylor, and D. Wu, "Enlarging the margin in perceptron decision trees," *Machine Learning*, vol. 41, pp. 295–313, 2000.

[15] V. Vapnik and A. Chervonenkis, "Uniform convergence of frequencies of occurence of events to their probabilities," *Dokl. Akad. Nauk SSSR*, vol. 181, pp. 915 – 918, 1968.

[16] L. Gurvits, "A note on a scale-sensitive dimension of linear bounded functionals in Banach spaces," in *Proceedings of Algorithmic Learning Theory, ALT-97*, 1997.

[17] P. Bartlett and J. Shawe-Taylor, "Generalization performance of support vector machines and other pattern classifiers," in *Advances in Kernel Methods – Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. 1999, pp. 43–54, MIT Press.

[18] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

[19] C. Saunders, A. Gammermann, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Machine Learning: Proceedings of the Fifteenth International Conference*, J. Shavlik, Ed. 1998, Morgan Kaufmann.

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

[21] F. W. Smith, "Pattern classifier design by linear programming," *IEEE Transactions on Computers*, vol. C-17, pp. 367–372, 1968.

[22] K. P. Bennett and O. L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimization Methods and Software*, vol. 1, pp. 23–34, 1992.

[23] Kristin P. Bennett, Ayhan Demiriz, and John Shawe-Taylor, "A column generation algorithm for boosting," *Machine Learning*, 2001, (to appear).

[24] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler, "Scale-sensitive dimensions, uniform convergence, and learnability," *Journal of the ACM*, vol. 44, no. 4, pp. 615–631, 1997.

[25] J. Shawe-Taylor and R.C. Williamson, "Generalization performance of classifiers in terms of observed covering numbers," in *Lecture Notes in Artificial Intelligence 1572. Proceedings of the European Conference on Computational Learning Theory, EuroCOLT'99*. 1999, pp. 274–284, Springer-Verlag.

[26] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer Verlag, 1982.

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

[28] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems 9*, M. Mozer, M. Jordan, and T. Petsche, Eds. 1997, MIT Press.

[29] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data using support vector machines," 1999, [http://www.cse.ucsc.edu/research/compbio/genex/genex.html]. Santa Cruz, University of California, Department of Computer Science and Engineering.

[30] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean, "Boosting algorithms as gradient descent in function space," Tech. Rep., RSISE, Australian National University, 1999.

[31] G. Ratsch, B. Schölkopf, A.J. Smola, S. Mika, T. Onoda, and K-R. Muller, "Robust ensemble learning," in *Advances in Large Margin Classifiers*, A.J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., Cambridge, MA, 2000, pp. 208–222, MIT Press.

[32] R. C. Williamson, A. J. Smola, and B. Schölkopf, "Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators," in *Lecture Notes in Artificial Intelligence 1572. Proceedings of the European Conference on Computational Learning Theory, EuroCOLT'99*. 1999, pp. 285–299, Springer-Verlag, to appear in IEEE Transactions on Information Theory.

[33] Ying Guo, Peter L. Bartlett, John Shawe-Taylor, and Robert C. Williamson, "Covering numbers for support vector machines," *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 239–250, 2002.

John Shawe-Taylor John Shawe-Taylor obtained a PhD in Mathematics at Royal Holloway, University of London in 1986. He subsequently completed an MSc in the Foundations of Advanced Information Technology at Imperial College. He was promoted to Professor of Computing Science in 1996. He has published over 100 research papers and is co-author of the book “An Introduction to Support Vector Machines”. He has been participated in and coordinated a number of European wide projects investigating the theory and practice of Machine Learning. He has served on the editorial board of Machine Learning, the Journal of Machine Learning Research and Neurocomputing. He co-founded the European Conference on Computational Learning Theory, EuroCOLT.

Nello Cristianini Nello Cristianini is currently a Visiting Researcher at UC Berkeley. He has a PhD in Engineering Mathematics from University of Bristol, a MSc in Computational Intelligence from Royal Holloway University of London, and a BSc in Physics from University of Trieste. He has been publishing in the fields of learning theory, pattern recognition and bioinformatics since 1996, and is coauthor of the book 'An introduction to Support Vector Machines'. He is currently Action Editor of the "Journal of Machine Learning Research", has been guest editor of two special issues and organized 2 meetings on the topics of Kernel Methods for Pattern Analysis.