Support Vector Machines for Classification and Regression

Rohan Shiloh Shah

Master of Science

Computer Science

McGill University

Montreal, Quebec

2007-09-31

A thesis submitted to McGill University
in partial fulfillment of the requirements of the
Degree of Master of Science

ABSTRACT

In the last decade Support Vector Machines (SVMs) have emerged as an important learning technique for solving classification and regression problems in various fields, most notably in computational biology, finance and text categorization. This is due in part to built-in mechanisms to ensure good generalization which leads to accurate prediction, the use of kernel functions to model non-linear distributions, the ability to train relatively quickly on large data sets using novel mathematical optimization techniques and most significantly the possibility of theoretical analysis using computational learning theory. In this thesis, we discuss the theoretical basis and computational approaches to Support Vector Machines.

ABRÉGÉ

Au cours des dix dernires années, Support Vector Machines (SVMs) est apparue être une technique importante d'apprentissage pour résoudre des problèmes de classification et de régression dans divers domaines, plus particulièrement en biologie informatique, finance et catégorisation de texte. Ceci est du, en partie aux mécanismes de construction assurant une bonne généralisation qui conduit à une prédiction précise, une utilisation des fonctions de kernel afin de modèliser des distributions non-linéaires, et à la possibilité de tester de façon relativement rapide sur des grands ensemble de données en utilisant de nouvelles techniques d'optimisation, en particulier, la possibilité d'analyses théoriques utilisant la théorie d'apprentissage informatique. Dans cette thèse, nous discutons des bases théoriques et des approches informatiques des Support Vector Machines.

TABLE OF CONTENTS

LIST OF FIGURES

MATHEMATICAL NOTATION

$\mathcal{X} \times \mathcal{Y}$ —— Input-Output (Observation) Space

$\mathcal{S} \in \mathcal{X} \times \mathcal{Y}$ —— Training set of random samples

n —— Size of Training Set

$\mathcal{S}_n \in \mathcal{X}$ —— Input vector set of size n

$\mathcal{F}$ —— Feature Space

$\Phi : \mathcal{X} \to \mathcal{F}$ —— Non-linear embedding into the feature space

$\mathcal{X}$ —— Space of all possible input vectors

$d = \dim(\mathcal{X})$ —— Dimension of the input space (length of $\vec{x}_i$ or the number of explanatory variables)

$\vec{x}_i \in \mathcal{X}$ —— Input vector or random sample

$y_i \in \mathbb{R}$ —— Annotation for regression

$y_i \in \{+1, -1\}$ —— Annotation for binary classification

y_t —— Annotation for test example x_t

$\mathcal{Y}$ —— Annotation (output) Space

$\mathcal{H}$ —— Hypothesis (Hilbert) space

$f \in \mathcal{H} : \mathcal{X} \to \mathcal{Y}$ —— A hypothesis (regression, prediction, decision) function

$\mathcal{B} = \{+1, -1\}^{\mathcal{X}}$ —— Hypothesis space of all binary valued functions

$\mathcal{R} = \mathbb{R}^{\mathcal{X}}$ —— Hypothesis space of all compact real valued functions

$\mathcal{Y}^{\mathcal{X}}$ —— Hypothesis space of *all* functions mapping $\mathcal{X}$ to $\mathcal{Y}$

$\mathcal{J}$ —— Hypothesis space of discriminant hyperplanes

$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ —— Kernel function

$K_{\mathcal{S}}$ —— The restriction of K to $\mathcal{S} = \{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n\}$

$k_{ij} = K_{\mathcal{S}}(x_i, x_j)$ —— Finite kernel matrix

$\mathcal{H}_K$ —— Reproducing Kernel Hilbert Space (RKHS)

$\langle \cdot, \cdot \rangle_{\mathcal{H}_K}, \| \cdot \|_{\mathcal{H}_K}$ —— Inner Product and Norm in a RKHS $\mathcal{H}_K$

$(. \cdot .)$ —— Dot Product in a Euclidean Space

$\forall g \in \mathcal{H},\ F_g : \mathcal{H} \to \mathbb{R}$ —— Linear Functional

$\mathcal{E}_{\vec{x}} : \mathcal{H} \to \mathbb{R}$ —— Evaluation Functional

$P : \mathcal{H} \to L$ —— Projection operator of $\mathcal{H}$ onto a subspace L

$L^2(\mathcal{X})$ —— Space of square integrable functions

$T_K : L^2(\mathcal{X}) \to L^2(\mathcal{X})$ —— Integral operator

υ_i —— Eigenvalue of T_K associated with eigenvector ς_i

ς_i —— Eigenvector of T_K associated with eigenvalue υ_i

$\mathfrak{H} \in \mathcal{J}$ —— Decision (Hyperplane) Boundary

$\mathfrak{H}_+$ and $\mathfrak{H}_-$ —— The margin boundaries on either side of the decision boundary

$\mathfrak{h}$ —— Linear function parametrized in terms of a weight vector $\vec{w}$ and scalar bias b

$\mathfrak{h}'$ —— First derivative of the linear function $\mathfrak{h}$

R_ν —— Empirical Margin Error

$R_{\mathcal{X}}$ —— Expected Risk

$R_{\mathcal{S}}$ —— Sample Error

$\hat{R}_n$ —— Empirical Risk

f^* —— Function that minimizes the expected risk

f_n^* —— Function that minimizes the empirical risk

$\mathcal{H}_{\mathcal{T}}$ —— RKHS $\mathcal{H}$ that is bounded $\|\mathcal{H}\|_{\mathcal{K}} \leq \mathcal{T}$

$\mathcal{L}(\mathcal{H}, \mathcal{X})$ —— Loss Class

$\ell(f, \{\vec{x}, y\})$ —— Loss Function

$\mathcal{V}$ —— VC-Dimension

$\Pi_{\mathcal{B}}(n)$ —— Growth Function

$\mathcal{N}(\mathcal{B}, \mathcal{S}_n)$ —— VC-Entropy

$\mathcal{N}(\mathcal{H}, \epsilon)$ —— Covering Number with radius ϵ

$\mathcal{D}(\mathcal{H}, \epsilon)$ —— Packing Number with radius ϵ

1

Introduction

The first step in supervised learning is the observation of a phenomenon or random process which gives rise to an annotated training data set:

$$\mathcal{S} = \{\vec{x}_i, y_i\}_{i=1}^n \quad \vec{x}_i \in \mathcal{X},\ y_i \in \mathcal{Y}$$

The output or annotation space $\mathcal{Y}$ can either be discrete or real valued in which case we have either a classification or a regression task. We will assume that the input space $\mathcal{X}$ is a finite dimensional real space $\mathbb{R}^d$ where d is the number of explanatory variables.

The next step is to model this phenomenon by attempting to make a causal link $f : \mathcal{X} \to \mathcal{Y}$ between the observed inputs $\{\vec{x}_i\}_{i=1}^n$ from the input space $\mathcal{X}$ and their corresponding observed outputs $\{y_i\}_{i=1}^n$ from the annotation space $\mathcal{Y}$; in a classification task the hypothesis/prediction function f is commonly referred to as a decision function whereas in regression it is simply called a regression function. In other words we seek to estimate the unknown conditional probability density function that governs the random process, which can then be used to define a suitable hypothesis: $f(\vec{x}_t) = \max_{y \in \mathcal{Y}} P(y|\vec{x}_t)$.

The hypothesis must minimize some measure of error over the observed training set while also maintaining a simple functional form; the first condition ensures that a causal link is in fact extracted from the observed data while the second condition avoids over-fitting the training set with a complex function that is unable to *generalize* or accurately predict the annotation of a test example.

The complexity of the hypothesis f can be controlled by restricting the capacity of the hypothesis space; but what subset of the space of all possible maps between the input and output spaces $\mathcal{Y}^X$ should we select as the hypothesis space $\mathcal{H} \subset \mathcal{Y}^X$? It must be rich or large enough to include a hypothesis function that is a good approximation of the target concept (the *actual* causal

link) but it must be poor enough to not include functions that are unnecessarily complex and are able to fit the observed data perfectly while lacking generalization potential.

The Support Vector Machine (SVM) is one approach to supervised learning that takes as input an annotated training data set and outputs a generalizable model, which can then be used to accurately predict the outcomes of future events. The search for such a model is a balance between minimizing the training error (or empirical risk) and regulating the capacity of the hypothesis space. Since the SVM machinery is linear we consider the hypothesis space of all $d-1$ dimensional hyperplanes. The 'kernel trick' may be applied to convert this or any linear machine into a non-linear one through the use of an appropriately chosen kernel function.

In binary SVM classification (SVMC), each input point is assigned one of two annotations $\mathcal{Y} = \{+1, -1\}$. The training set is *separable* if a hyperplane can divide $\mathbb{R}^d$ into two half-spaces corresponding to the positive and negative classes. The hyperplane that maximizes the margin (minimal distance between the positive and negative examples) is then selected as the unique SVM hypothesis. If the training set is not separable, then a further criterion is optimized, namely the empirical classification error. In SVM regression (SVMR), the margin boundaries are fixed in advance at a value $\epsilon \geq 0$ above and below the potential regression function; those training points that are within this ϵ-tube incur no loss in contrast to those outside it. Different configurations of the potential hypothesis, which is again taken to be a hyperplane, lead to different values for the loss which is minimized to find the solution.

The thesis is organized as follows; in Chapter 2 we consider modeling *non-linear* causal links by using kernel functions that implicitly transform the observed inputs into feature vectors $\vec{x} \rightarrow \phi(\vec{x})$ in a high-dimensional feature (flattening) space $\phi(\vec{x}) \in \mathcal{F}$ where *linear* classification/regression SVM techniques can then be applied. An information theoretic analysis of learning is considered in Chapter 3 where the hypothesis space is restricted $\mathcal{F} \subset \mathcal{Y}^X$ on the basis of the amount of training data that is available. Computational considerations for *linear* SVMC and *linear* SVMR are given separately in chapters 4 and 5 respectively; the solution in both instances is determined by solving a quadratic optimization problem with linear inequality constraints.

2

Kernel Methods

All kernel methods make use of a *kernel function* that provides an implicit mapping or projection of a training data set into a *feature space* $\mathcal{F}$ where discriminative classification or regression is performed. Implicitly a kernel function can be seen as an inner product between a pair of data points in the feature space, explicitly however it is simply a function evaluation for the same pair of data points in the input space $\mathcal{X}$ before any mapping has been applied. We will introduce the basic mathematical properties and associated function spaces of kernel functions in the next section and then consider an example known as the Fisher kernel.

2.1 Explicit Mapping Of Observations To Features

The complexity of a training data set, which is sampled from the observation space, affects the performance of any learning algorithms that might make use of it; in extreme cases certain classes of learning algorithms might not be able to learn an appropriate prediction function for a given training data set. In such an instance we have no choice but to manipulate the data so that learning is possible; for example in figure 2.1 we see that if we consider empirical target functions from the hypothesis class of discriminative hyperplanes then a quadratic map must first be applied.

In other instances the training data might not be in a format that the learning algorithm accepts and so again a manipulation or mapping of the data is required. For example the data may be nucleotide sequences of which a numerical representation is required and hence preprocessing steps must be taken.

As we will see later, the most important reason for transforming the training data is that the feature space is often endowed with a structure (definition

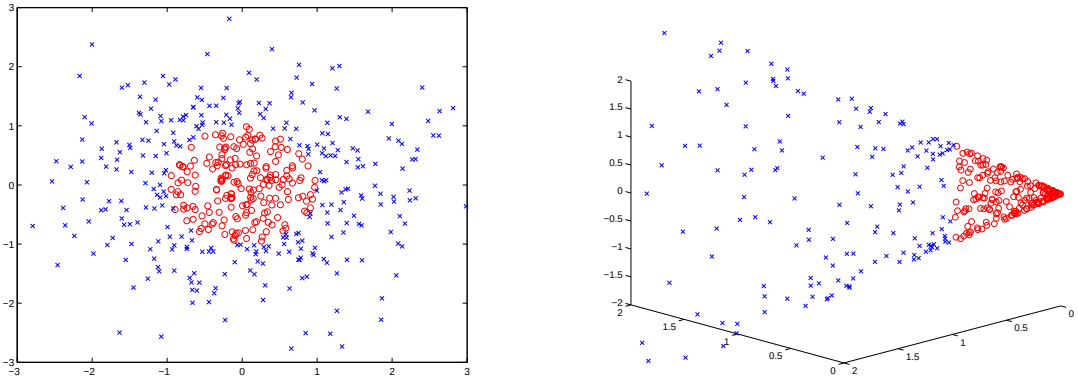

Figure 2–1: [left] Circular decision boundary in $\mathbb{R}^2$: $x_1^2 + x_2^2 = 1$. [right] Data is replotted in a $\mathbb{R}^3$ feature space using a quadratic map: $\Phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ and is then linearly separable.

2.3.7, theorem 2.3.2) that may be exploited (section 2.5, theorem 2.3.3) by the learning algorithm.

Now that we have established that a mapping is necessary, we must decide how to represent the mapped data and then define a corresponding mapping function. The simplest representation [SS01] results from defining a (often non-linear) mapping function $\Phi(\cdot) \in \mathcal{H}$ over the inputs $\vec{x}_i \in \mathcal{X}$ in our training set;

$$\mathcal{S} = \{\vec{x}_i, y_i\}_{i=1}^n \quad \vec{x}_i \in \mathcal{X},\ y_i \in \mathcal{Y}$$

and then representing the data as the set of mapped data

$$\{\Phi(\vec{x}_i)\ , y_i\}_{i=1}^n \quad \Phi(x_i) \in \mathcal{H},\ y_i \in \mathcal{Y}$$

There are several problems that arise from representing the data individually by applying the mapping to each input example; the most common of which is computational since Φ may map elements into a feature space of infinite dimension.

2.2 Finite Kernel Induced Feature Space

We now consider a different approach to the issue of data representation; instead of mapping each training example x_i individually into features $\Phi(x_i)$ using the map $\Phi : \mathcal{X} \to \mathcal{F}$, kernel methods represent the data as a set of *pairwise computations*

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{2.1}$$

Such a kernel function K is defined over a possibly infinite space $\mathcal{X}$; we restrict its domain to observations in the training set $\mathcal{S}$ and thereby define a *finite kernel*:

$$K_{\mathcal{S}} : \vec{x}_i \times \vec{x}_j \to \mathbb{R} \quad \forall i : 1 \leq i \leq n$$

Finite kernels can be represented as square $n \times n$ matrices where $k_{ij} = K_{\mathcal{S}}(x_i, x_j) \in \mathbb{R}$

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix} \tag{2.2}$$

Although the kernel representation may seem unintuitive at first, it has many benefits over the explicit use of a mapping function Φ; later in Theorem 2.3.3 we will see that both these approaches are in fact equivalent and there exists an implicit mapping (2.53, 2.54) and an associated feature space (2.41, 2.44) for every kernel function that is positive-definite. The class of comparison functions is clearly limited by considering only positive-definite kernels but this restriction is applied so that we can make use of an essential 'trick' (section 2.5) that simplifies the objective function of the quadratic optimization that gives rise to the final solution; this trick is possible due to Mercer's Theorem (2.3.3) for which positive definiteness is a necessary and sufficient condition.

Furthermore, depending on the nature of the data to be analyzed it might be significantly more complicated [SS01] to find individual representations of the observations than to consider pairwise comparisons between them. For example, representing a set of protein or DNA sequences as pairwise comparisons between members of the set is easier and potentially more relevant than using vectors for each attribute individually.

The most significant advantage that kernel functions have over the use of an explicit mapping function is that it generalizes the representation of the input data so that an absolute modularity exists between the preprocessing of input data and the training algorithm. For example, given inputs $\{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n\} \in \mathcal{X}$ we could define two mapping functions to extract different features $\phi_p \in \mathbb{R}^p$ and $\phi_q \in \mathbb{R}^q$; now if the dimension of the feature spaces are not equal $p \neq q$ then we have sets of vectors of different lengths $\{\phi_p(\vec{x}_1), \phi_p(\vec{x}_2), \cdots, \phi_p(\vec{x}_n)\}$ and $\{\phi_q(\vec{x}_1), \phi_q(\vec{x}_2), \cdots, \phi_q(\vec{x}_n)\}$ and so the training algorithm must be modified to accept these two different types of input

data. However, regardless of the kernel function used but more significantly regardless of the dimension of the feature space, the resulting kernel matrix is square with dimensions $n \times n$ since we consider only pairwise comparisons between the inputs; the only drawback is that there is less control over the process of extracting features since we relinquish some control of choice of the resulting feature space.

Provided the inputs are defined in an inner product space, we can build a *linear* comparison function by taking the inner product

$$K(\vec{x}_i, \vec{x}_j) = \langle \vec{x}_i \cdot \vec{x}_j \rangle_{\mathcal{X}} \tag{2.3}$$

or dot product if $\mathcal{X}$ is a real vector space:

$$K(x_i, x_j) = (\vec{x}_i \cdot \vec{x}_j) \tag{2.4}$$

Geometrically, the dot product calculates the angle between the vectors $\vec{x_i}$ and $\vec{x_j}$ assuming they are normalized (section 4.1) such that $\|\vec{x}_i\| = \sqrt{\langle \vec{x}_i \cdot \vec{x}_i \rangle} = 1$ and $\|\vec{x}_i\| = 1$.

If inner products are not well-defined in the input space $\mathcal{X}$ then we must explicitly apply a map Φ first, projecting the inputs into an inner product space. We can then construct the following comparison function;

$$K(x_i, x_j) \equiv \langle \Phi(\vec{x}_i) \, , \Phi(\vec{x}_j) \rangle_{\mathcal{H}} \tag{2.5}$$

An obvious question one could ask is does the simple construction define the entire class of positive-definite kernel functions? More specifically, can every positive-definite kernel be decomposed into an inner product in some space? We will prove this in the affirmative and also characterize the corresponding inner product space in the following sections.

2.3 Functional view of the Kernel induced Feature Space

So far we have seen a geometrical interpretation of finite kernels as implicit/explicit projections into a feature space; the associated linear algebra using finite kernel matrices over $\mathcal{S} \times \mathcal{S}$, was realized in a finite dimensional vector space. Now we consider an alternative analysis using *kernel functions*

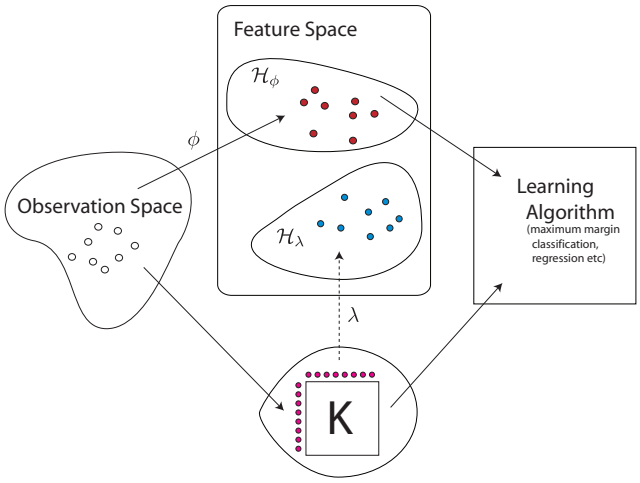

Figure 2–2: Explicit (ϕ) and implicit (λ) mapping of inputs to features.

defined over a dense space (no longer restricted to a finite, discrete space $\mathcal{S}\times\mathcal{S}$) and integral operator theory in an *infinite dimensional function space* which serves as the hypothesis space; a hypothesis being a function from $\mathcal{X}\to\mathcal{Y}$.

If we are to predict in a classification/regression task, then any potential hypothesis function will need to be evaluated at a test data point and hence we will require that they are point-wise defined so that all function evaluations exist within the space of annotations $\mathcal{Y}$. We will denote the space of all real-valued, point-wise defined functions on the domain $\mathcal{X}$ by $\mathbb{R}^{\mathcal{X}}$. Finally, convergent sequences of functions in the hypothesis space should also be point-wise convergent; this is shown to hold in Reproducing Kernel Hilbert spaces (2.38) whereas it does not hold in general for Hilbert spaces, in particular for L^2.

$$\|f_n - f\|_{\mathcal{H}} \to 0 \implies \lim_{n\to\infty} f_n(\vec{x}) - f(\vec{x}) = 0,\ \forall \vec{x}\in\mathcal{X} \tag{2.6}$$

Furthermore, we will show that point-wise convergence in $\mathcal{H}$ implies the continuity of evaluation functionals (2.11) on $\mathcal{H}$. In fact, in the following chapter we will see that an even stronger convergence criterion, that of uniform convergence, is necessary for learning.

In this chapter we show how a certain class of *kernel functions exist in all (and in some sense generate) Hilbert spaces of real valued functions* under a few simple conditions. The material for this section was referenced from [CS02], Chapter 2 of [BTA04], [Zho02], [Zho03], [Gir97], Chapter 3 of [Muk07], [LV07], [Qui01], [CMR02], [HN01], [SSM98], [SHS01], [STB98], [SS05] and [Rud91].

2.3.1 Hilbert Spaces

A Hilbert space is a *complete inner product space* and so distances[1] and angles[2] are well defined. Formally a Hilbert space is a function space $\mathcal{H}$ along with an inner product $\langle h, g\rangle$ defined for all $h, g \in \mathcal{H}$ such that the norm defined using the inner product $\|h\|_{\mathcal{H}} = \langle h, h\rangle_{\mathcal{H}}^{1/2}$ completes the space; this is possible if and only if every sequence $\{h_i\}_{i=1}^{\infty}$ with $h_i \in \mathcal{H}$ satisfying the Cauchy criteria;

$$\forall \epsilon \; \exists N(\epsilon) \in \mathbb{N} \text{ such that } \forall n, m > N(\epsilon) \quad : \quad \|h_n - h_m\|_{\mathcal{H}} < \epsilon$$

converges to a limit contained *within* the space;

$$\lim_{i \to \infty} h_i \in \mathcal{H}$$

Given either an open or closed subset N of a Hilbert space $\mathcal{H}$, we define its orthogonal complement as the space:

$$N^{\perp} = \{l \in \mathcal{H} : \langle l, g\rangle = 0, \; \forall g \in \bar{N}\}$$

noting that the only instance when $\langle g, g\rangle = 0$ is if g is identically zero which implies that $\bar{N} \cap N^{\perp} = \{0\}$. The *direct sum* of these two complementary spaces [3] equals $\mathcal{H}$:

$$\mathcal{H} = \bar{N} \oplus N^{\perp} = \{g + l : g \in \bar{N} \text{ and } l \in N^{\perp}\} \tag{2.7}$$

although the union of these same subspaces need not cover $\mathcal{H}$:

$$\bar{N} \cup N^{\perp} \subseteq \mathcal{H} \tag{2.8}$$

So any function $h \in \mathcal{H}$ can be represented as the sum of two other functions;

$$h = g + l \tag{2.9}$$

---

[1] Every inner product space is a normed space which in turn is a metric space, $d(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\| = \sqrt{\langle \vec{x}, \vec{y}\rangle}$

[2] Orthogonality in particular; determined by the inner product

[3] The closure of N and its orthogonal complement $N^{\perp}$, both of which are Hilbert spaces themselves

where $g \in N^{\perp}$ and $l \in \bar{N}$. Therefore every Hilbert space $\mathcal{H}$ can be decomposed into two distinct (except for the zero vector) closed subspaces; however this decomposition need not be limited to only two mutually orthogonal subspaces.

Infinite-dimensional Hilbert spaces are similar to finite-dimensional spaces in that they must have (proof using Zorn's Lemma combined with the Gram-Schmidt orthogonalization process) an orthonormal basis $\{h_1, h_2, \cdots : h_i \in \mathcal{H}\}$ satisfying

- Normalization: $\|h_i\| = 1 \ \ \forall i$

- Orthogonality: $\langle h_i, h_j \rangle = 0$ if $i \neq j$

so that every function in $\mathcal{H}$ can be represented uniquely as an unconditionally convergent, linear combination of these fixed elements

- Completeness: $\forall h \in \mathcal{H}, \ \exists \{\alpha_1, \alpha_2, \cdots : \alpha_i \in \mathbb{R}\}$ such that $h = \sum_{i=1}^{\infty} \alpha_i h_i$

Note that an orthonormal basis is the maximal subset of $\mathcal{H}$ that satisfies the above three criteria. It is of infinite cardinality for infinite-dimensional spaces. Let N_i be the space spanned by h_i then:

$$\mathcal{H} = N_1 \oplus N_2 \oplus \cdots \oplus N_i \oplus \cdots$$

although as before

$$N_1 \cup N_2 \cup \cdots \cup N_i \cup \cdots \subseteq \mathcal{H}$$

Finally, when the Hilbert space is infinite dimensional, the span of the orthonormal basis need not be equal to the entire space but instead must be dense in it; for this reason it is not possible to express *every* element in the space as a linear combination of select elements in the orthonormal basis. We will assume henceforth that Hilbert spaces have a countable orthonormal basis. Such a space is *separable* so it contains a countable everywhere, dense subset whose closure is the entire space. When the Hilbert space is a finite-dimensional function space then there exists a finite orthogonal basis so that every function in the space and every linear operator acting upon these functions can be represented in matrix form.

2.3.2 Linear Functionals

A *functional* $\mathcal{F}$ is a real-valued function whose arguments are also functions (specifically the hypothesis function $f : \mathcal{X} \to \mathcal{Y}$) taken from some space $\mathcal{H}$:

$$\mathcal{F} : \mathcal{H}(\mathcal{X} \to \mathcal{Y}) \to \mathbb{R}$$

An *evaluation functional* $\mathcal{E}_{\vec{x}}[f] : \mathcal{H}(\mathcal{X}) \to \mathcal{Y}$ simply evaluates a hypothesis function $f \in \mathcal{H}$ at some fixed point $\vec{x} \in \mathcal{X}$ in the domain:

$$\mathcal{E}_{\vec{x}}[f] = f(\vec{x}) \tag{2.10}$$

Point-wise convergence in the hypothesis space ensures the continuity of the evaluation functional:

$$f_n(\vec{x}) \to f(\vec{x}),\ \forall \vec{x} \implies \mathcal{E}_{\vec{x}}[f_n] \to \mathcal{E}_{\vec{x}}[f],\ \forall \vec{x} \tag{2.11}$$

Linear functionals are defined over a linear (vector) space whose elements can be added and scaled under the functional:

$$\mathcal{F}(\alpha_1 h_1 + \alpha_2 h_2) = \alpha_1 \mathcal{F}(h_1) + \alpha_2 \mathcal{F}(h_2),\ \ \forall h_1, h_2 \in \mathcal{H}$$

The set of functionals themselves form a vector space $\mathcal{J}$ if they can be added and scaled:

$$\mathcal{F}_1(\alpha_1 h) + \mathcal{F}_2(\alpha_2 h) = (\alpha_1 \mathcal{F}_1 + \alpha_2 \mathcal{F}_2)(h),\ \ \forall \mathcal{F}_1, \mathcal{F}_2 \in \mathcal{J}, \forall h \in \mathcal{H}$$

The null space and image (range) space of the functional $\mathcal{F}$ are defined as:

$$\texttt{null}_{\mathcal{F}} \equiv \{h \in \mathcal{H} : \mathcal{F}(h) = 0\}$$

$$\texttt{img}_{\mathcal{F}} \equiv \{\mathcal{F}(h) : h \in \mathcal{H}\}$$

and are subspaces of the domain $\mathcal{H}$ and co-domain $\mathbb{R}$ respectively. The Rank-Nullity Theorem [Rud91] for finite-dimensional spaces states that the dimension of the domain is the sum of the dimensions of the null and image subspaces:

$$\texttt{dim}(\mathcal{H}) = \texttt{dim}(\texttt{null}_{\mathcal{F}}) + \texttt{dim}(\texttt{img}_{\mathcal{F}})$$

A *linear functional* is bounded if for some constant α the following is satisfied

$$|\mathcal{F}(h)| \leq \alpha \|h\|_{\mathcal{H}}\ \ \forall h \in \mathcal{H}$$

Furthermore, boundedness implies continuity of the linear functional. To see this, let us assume we have a sequence of functions in a Hilbert space that converge to some fixed function $h_i \longrightarrow h$ so that $\|h_i - h\|_{\mathcal{H}} \longrightarrow 0$. Then the continuity criteria for the linear bounded functional $\mathcal{F}$ is satisfied:

$$\forall \epsilon > 0,\ \exists N \in \mathbb{N},\ \text{such that } \forall i > N \tag{2.12}$$
$$|\mathcal{F}(h_i) - \mathcal{F}(h)| = |\mathcal{F}(h_i - h)| \leq \alpha \|h_i - h\|_{\mathcal{H}} \longrightarrow 0$$

Let $\{h_1, h_2, \cdots : h_i \in \mathcal{H}\}$ be an orthonormal basis for a Hilbert space which in a linear combination can be used to express any vector $h \in \mathcal{H}$

$$h = \sum_{i=1}^{\infty} \alpha_i h_i \ = \ \sum_{i=1}^{\infty} \langle h, h_i \rangle h_i$$

where the second equality follows from:

$$\langle h, h_j \rangle = \left\langle \sum_{i=1}^{\infty} \alpha_i h_i, h_j \right\rangle = \sum_{i=1}^{\infty} \alpha_i \langle h_i, h_j \rangle = \alpha_j$$

where the second equality follows from the linearity and continuity (which is necessary since we have an infinite sum) of the inner product and the third equality follows from the orthogonality of the basis. So any *linear and continuous* functional over an infinite-dimensional Hilbert space can be decomposed into a linear combination of linear functionals applied to the orthonormal basis using the same coefficients as above:

$$\mathcal{F}(h) = \sum_{i=1}^{\infty} \langle h, h_i \rangle \mathcal{F}(h_i) = \sum_{i=1}^{\infty} \langle h, h_i \mathcal{F}(h_i) \rangle \tag{2.13}$$

Definition 2.3.1 (Projection Operator) *A projection $P : \mathcal{H} \to L$ over a (vector) space $\mathcal{H} = G \oplus L$ is a linear operator that maps points from* $\mathcal{H}$ along *the subspace G* onto *the subspace L; these two subspaces are complementary, the elements in the latter are mapped by P to themselves (image of P) while those in the former are mapped by P to zero (nullity of P).*

Application of the projection twice is equivalent to applying it a single time, the operator is therefore idempotent:

$$P = P^2$$

The operator $(I - P)$ is then the *complimentary projection* of $\mathcal{H}$ *along L onto G*. A projection is called *orthogonal* if its associated image space and null space are orthogonal complements in which case P is necessarily self-adjoint.

When the space $\mathcal{H}$ over which P is defined is finite-dimensional, i.e. $\texttt{dim}(\mathcal{H}) = n$, the projection P is a finite-dimensional $n \times n$ matrix whose entries are a function of the basis vectors of L. In figure 4–1 we see an orthogonal projection of $\vec{x}$ onto $\vec{w}$, in which case the projection matrix is given by:

$$P_{\vec{w}} = \frac{\vec{w}}{\|\vec{w}\|} \frac{\vec{w}^\top}{\|\vec{w}\|}$$

so that any vector orthogonal to $\vec{w}$ (parallel to the hyperplane $\mathfrak{H}$ which we will assume intersects the origin so that the bias term $b = 0$) is mapped to zero. The orthogonal projection is then given by the vector:

$$P_{\vec{w}}\vec{x} = \left(\frac{\vec{w}}{\|\vec{w}\|} \frac{\vec{w}^\top}{\|\vec{w}\|} \right) \vec{x}$$

which is equivalent to the vector resolute defined in (4.7).

More generally, let us consider the subspace $L \subset \mathcal{H}$ with an orthonormal basis $\{l_1, l_2, \cdots, l_t\}$. The projection matrix is then given by the square of the matrix L_p whose columns are the vectors that form the orthonormal basis:

$$P_L = L_p L_p^\top = [l_1|l_2|\cdots|l_t] \ [l_1|l_2|\cdots|l_t]^\top$$

If the vectors do not form an orthonormal basis then the projection matrix is given by 'normalizing' the above projection:

$$P_L = L_p (L_p^\top L_p)^{-1} L_p^\top$$

Note the similarity to the normal equations used in linear regression.

2.3.3 Inner Product Dual Spaces

If $\mathcal{H}$ is a Hilbert space then the associated inner product[4] can be used to define a linear (bounded) functional:

$$\mathcal{F}_g(\cdot) = \langle g, \cdot \rangle_{\mathcal{H}} \in \mathcal{H}^*$$

[4] which can be shown [HN01] to be a bounded mapping and hence by (2.12) must be continuous

The functional defined in terms of a kernel (2.1) function $K_{\vec{x}} = K(\vec{x}, \cdot) \in \mathcal{H}$, is given by:

$$\mathcal{F}_{Kx}(\cdot) = \langle K_{\vec{x}}, \cdot \rangle_{\mathcal{H}} \in \mathcal{H}^*$$

for some input vector $\vec{x} \in \mathcal{X}$. So essentially every element $g \in \mathcal{H}$ (or $K(\vec{x}, \cdot) \in \mathcal{H}$) has a corresponding linear bounded functional in a dual space $\mathcal{H}^*$:

$$g \longmapsto \mathcal{F}_g(\cdot) = \langle g, \cdot \rangle_{\mathcal{H}} \in \mathcal{H}^*$$

The dual space $\mathcal{H}^*$ of all linear bounded functionals on a Hilbert space $\mathcal{H}$ is also Hilbertian [HN01] and has a *dual basis* that is a function of the orthonormal basis of the original space. The spaces $\mathcal{H}$ and its dual $\mathcal{H}^*$ are isomorphic so that each element (function) in the former has a corresponding element (functional) in the latter and vice versa. The null space of the functional fixed at a basis vector g is then given by

$$\texttt{null}_{\mathcal{F}_g} \equiv \{h \in \mathcal{H} : \mathcal{F}_g(h) = \langle h, g \rangle_{\mathcal{H}} = 0\} \tag{2.14}$$

and consists of all the vectors (including the zero vector) in $\mathcal{H}$ that are orthogonal to g. The null space therefore has dimension one less than the dimension of $\mathcal{H}$ since g is orthogonal to all the basis vectors except itself. Hence the dimension of the space orthogonal to the null space is one by the Rank-Nullity Theorem:

$$\texttt{dim}((\texttt{null}_{\mathcal{F}})^{\perp}) = 1$$

We now state an important theorem that will help establish a subsequent result:

Theorem 2.3.1 (Riesz Representation Theorem) *Every bounded (continuous) linear functional $\mathcal{F}$ over a Hilbert space $\mathcal{H}$ can be represented as an inner product with a fixed, unique, non-zero vector $r_{\mathcal{F}} \in \mathcal{H}$ called the representer for $\mathcal{F}$:*

$$\exists r_{\mathcal{F}} \in \mathcal{H} \; (\exists \mathcal{G}_{r_{\mathcal{F}}} \in \mathcal{H}^*) \; : \; \mathcal{F}(h) = \langle r_{\mathcal{F}}, h \rangle_{\mathcal{H}} = \mathcal{G}_{r_{\mathcal{F}}}(h), \;\; \forall h \in \mathcal{H} \tag{2.15}$$

For an evaluation functional we therefore have:

$$\forall \vec{x} \in \mathcal{X}, \; \exists r_{\mathcal{E}x} \in \mathcal{H} \; : \; f(\vec{x}) = \mathcal{E}_{\vec{x}}[f] = \langle r_{\mathcal{E}x}, f \rangle_{\mathcal{H}} = \mathcal{G}_{r_{\mathcal{E}x}}(f), \;\; \forall f \in \mathcal{H} \tag{2.16}$$

Proof When $\mathcal{H}$ is finite dimensional the proof is trivial and follows from (2.13) since the finite summation can be taken inside the dot product so that the representer is a function of the finite basis of the space: $r_{\mathcal{F}} = \sum_{i=1}^{n} \mathcal{F}(h_i)h_i$.

We now consider the case where $\mathcal{H}$ is infinite-dimensional; in subsection (2.3.2) we saw that a bounded linear functional $\mathcal{F}$ must also be continuous which in turn implies that $\texttt{null}_{\mathcal{F}}$ is a *closed* linear subspace of $\mathcal{H}$. Hence by the Projection Theorem there must exist a non-zero vector $z \in \mathcal{H}$ that is orthogonal to the null space of $\mathcal{F}$:

$$z \perp \texttt{null}_{\mathcal{F}}$$

In fact, the basis vector that is orthogonal to the null space is unique so that the number of linearly independent elements in the subspace orthogonal to the null space of $\mathcal{F}$ is one:

$$\texttt{dim}((\texttt{null}_{\mathcal{F}})^{\perp}) = 1$$

This implies that any vector in $(\texttt{null}_{\mathcal{F}})^{\perp}$ can be expressed as a multiple of a single basis vector $g \in (\texttt{null}_{\mathcal{F}})^{\perp} \subset \mathcal{H}$. Using this single basis vector and a scalar value α_h we can decompose *any* vector $h \in \mathcal{H}$ as

$$h = \alpha_h g + l \tag{2.17}$$

where $\alpha_h g \in (\texttt{null}_{\mathcal{F}})^{\perp}$ and $l \in \texttt{null}_{\mathcal{F}}$ which after application of the functional gives:

$$\mathcal{F}(h) = \mathcal{F}(\alpha_h g) + \mathcal{F}(l) = \alpha_h \mathcal{F}(g) \tag{2.18}$$

from the linearity of the functional and the definition of the null space. If we take the inner product of (2.17) with g while assuming that $\|g\|_{\mathcal{H}} = 1$, we have:

$$\begin{aligned}
\langle h, g \rangle &= \langle \alpha_h g, g \rangle + \langle l, g \rangle \\
&= \alpha_h \langle g, g \rangle + 0 && (2.19) \\
&= \alpha_h \|g\|^2_{\mathcal{H}} && (2.20) \\
&= \alpha_h && (2.21) \\
&= \mathcal{F}(h)/\mathcal{F}(g) && (2.22)
\end{aligned}$$

where (2.19) follows from the orthogonality of l and g, (2.20) follows from the definition of the norm, (2.21) follows from our assumption that the vectors g be normalized and (2.22) follows from (2.18). Rearranging gives the functional

in terms of a dot product:

$$\mathcal{F}(h) = \langle h, g\mathcal{F}(g)\rangle \tag{2.23}$$

from which we see that the representer for $\mathcal{F}$ has the form:

$$r_{\mathcal{F}} = g\mathcal{F}(g) \tag{2.24}$$

□

2.3.4 Square Integrable Function Spaces

As an example let us consider the infinite-dimensional space $L^2(\mathcal{Z})$ of all real-valued, square integrable, Lebesgue measurable functions on the measure space $(\mathcal{Z}, \Sigma, \mu)$ where Σ is a σ-algebra (closed under complementation and countable unions) of subsets of $\mathcal{Z}$ and μ is a measure on Σ so that two distinct functions are considered equivalent if they differ only on a set of measure zero. We could take the domain $\mathcal{Z}$ to be either the closed $\mathcal{Z} = [a, b]$ or open $\mathcal{Z} = (a, b)$ intervals both of which have the same Lebesgue measure $\mu(\mathcal{Z}) = b - a$ since the closure of the open set has measure zero.

More generally, any closed or open subset of a finite-dimensional real space $\mathcal{Z} = \mathbb{R}^n$ is Lebesgue measurable in which case the space $L^2(\mathbb{R}^n)$ is infinite-dimensional (if the σ-algebra Σ has an infinite number of elements then the resulting $L^2(\mathcal{Z})$ space is infinite-dimensional). When we consider an infinite-dimensional measure space $(\mathcal{Z}, \Sigma)$ then the Lebesgue measure is not well defined as it fails to be both locally finite and translation-invariant. An inner product in terms of the Lebesgue integral is then given as:

$$\langle f, g\rangle_{L^2} = \int_{\mathcal{Z}} f(\vec{z})g(\vec{z})d\mu(\vec{z}) \tag{2.25}$$

Moreover, we define the norm (that completes the space) as

$$\|f\|_{L^2} = \sqrt{\langle f, f\rangle_{L^2}} \tag{2.26}$$

The space $L^2(\mathcal{Z})$ contains all functions that are *square-integrable* on $\mathcal{Z}$:

$$L^2(\mathcal{Z}) = \left\{ f \in \mathbb{R}^{\mathcal{Z}} : \|f\|_{L^2} = \sqrt{\langle f, f\rangle_{L^2}} = \left(\int_{\mathcal{Z}} f(\vec{z})^2 d\mu(\vec{z}) \right)^{1/2} < \infty \right\} \tag{2.27}$$

The function space $L^2(\mathcal{Z})$ is a Hilbert space since it is an inner product space that is closed under addition:

$$f, g \in L^2(\mathcal{Z}) \Longrightarrow f + g \in L^2(\mathcal{Z})$$

and is Cauchy complete (Riesz-Fischer Theorem). Hence, if we take a Cauchy sequence of square-integrable functions $\{h_1, h_2, \cdots : h_i \in \mathcal{H}\}$ satisfying:

$$\lim_{i,j\to\infty} \|h_i - h_j\|_{L^2} = \lim_{i,j\to\infty} \left(\int_{\mathcal{Z}} \left(h_i(\vec{z}) - h_j(\vec{z})\right)^2 d\mu(\vec{z}) \right)^{1/2} = 0$$

then there exists some square-integrable function $h \in \mathcal{H}$ that is the mean limit of the above Cauchy sequence:

$$\lim_{i\to\infty} \left(\int_{\mathcal{Z}} \left(h_i(\vec{z}) - h(\vec{z})\right)^2 d\mu(\vec{z}) \right)^{1/2} = 0$$

From the Reisz representation theorem it follows that every bounded, real-valued, linear functional on the Hilbert space L^2 is of the form:

$$\mathcal{F}(g) = \langle r_{\mathcal{F}}, g \rangle_{L^2} = \int_{\mathcal{Z}} r_{\mathcal{F}}(z) g(z) d\mu(z) = \mathcal{G}_{r_{\mathcal{F}}}(g) \tag{2.28}$$

We can generalize the $L^2(\mathcal{Z})$ function space as follows:

$$L^p(\mathcal{Z}) = \left\{ f \in \mathbb{R}^{\mathcal{Z}} : \|f\|^p = \left(\int_{\mathcal{Z}} |f|^p d\mu(\vec{z}) \right)^{1/p} < \infty \right\} \tag{2.29}$$

It is important to note that only in the case that $p = 2$ the resulting space is Hilbertian. When $p = 1$ then the space $L^1(\mathcal{Z})$ contains all functions that are *absolutely integrable* on $\mathcal{Z}$:

$$L^1(\mathcal{Z}) = \left\{ f \in \mathbb{R}^{\mathcal{Z}} : \|f\|_{L_1} = \|f\| = \int_{\mathcal{Z}} |f(\vec{z})| d\mu(\vec{z}) < \infty \right\}$$

When $p = \infty$ we use the *uniform norm* defined using the supremum operator instead of a dot product and obtain the space of bounded functions:

$$L^\infty(\mathcal{Z}) = \left\{ f \in \mathbb{R}^{\mathcal{Z}} : \|f\|_{L_\infty} = \sup_{\vec{z}\in\mathcal{Z}} |f(\vec{z})| < \infty \right\} \tag{2.30}$$

Convergent sequences of functions in L^∞ are uniformly convergent. Elements of the L^p spaces need not be continuous; discontinuous functions over domains of compact support are Lebesgue integrable as long as their discontinuities

have measure zero. In other words, when the discontinuous function is equivalent to a continuous one (which is Riemann integrable) almost everywhere (i.e. on a set of measure one) then their Lebesgue integrals are equal. These unmeasurable irregularities imply ([CMR02]) that functions in L^p are not point-wise well defined.

Since L^2 is a Hilbert space, it must have a countable orthonormal basis and hence is separable (has a countable everywhere dense subset) which implies that there exist square (Lebesgue) integrable functions almost everywhere. Furthermore, continuous functions are also dense in L^2 (as long as the domain has compact support); so any function in L^2 can be approximated infinitely accurately by a continuous function. Essentially, L^2 is the Cauchy completion of the space of continuous functions C^0 with respect to the norm (2.26) and includes those functions which although discontinuous, are almost everywhere equal to elements in C^0.

2.3.5 Space Of Continuous Functions

The space of all real-valued, continuous functions on the domain $\mathcal{X}$ that are differentiable up to k times is denoted by $C^k\left(\mathbb{R}^{\mathcal{X}}\right)$. Most frequently we will consider: the space C^0 of continuous functions, the space C^1 of continuous functions whose derivative is also continuous, the space C^2 of twice differentiable functions and the space of *smooth* functions C^∞ that are infinitely differentiable. One essential difference between L^2 and C^0 is that the latter is not Cauchy complete and is therefore not a Hilbert space. In fact, as mentioned previously, L^2 is the Cauchy completion of the function space C^0 or in other words, continuous functions on $\mathcal{X}$ are dense in $L^2(\mathcal{X})$.

2.3.6 Normed Sequence Spaces

We consider a special case of the L^p spaces where the measure μ is taken to be the counting measure and a summation is taken instead of an integral. Essentially we have a function from the natural numbers to the real line represented as a vector $\vec{z}$ of countably infinite length. The norm is then given by:

$$\|\vec{z}\|_{\ell_p} = \left(\sum_{i=1}^{\infty} |z_i|^p\right)^{1/p}$$

Convergence of the above series depends on the vector $\vec{z}$; so the space ℓ^p is taken as the set of all vectors $\vec{z}$ of infinite length that have a finite ℓ^p-norm:

$$\ell^p(\mathcal{Z}) = \{\vec{z} \in \mathcal{Z} : \|\vec{z}\|_{\ell^p} < \infty\}$$

It is important to note that the size of the ℓ^p space increases with p. For example ℓ^∞ is the space of all bounded sequences and is a superset of all other ℓ^p spaces: ℓ^1 is the space of all absolutely convergent sequences, ℓ^2 is the space of all square convergent sequences and ℓ^0 is the space of all null sequences (converges to zero). Of these only ℓ^2 is a Hilbert space and in fact, as we will see later, a reproducing kernel Hilbert space (RKHS).

2.3.7 Compact and Self Adjoint Operators

The linear algebra of compact operators acting on infinite-dimensional spaces closely resembles that of regular operators on finite-dimensional spaces.

Definition 2.3.2 (Compact Operator) *A bounded (continuous) linear operator T is compact if, when applied to the elements of* any *bounded subset of the domain, the resulting image space is precompact (totally bounded) or equivalently, if the closure of the resulting image space is compact (complete and totally bounded).*

Note however that the entire domain itself might be unbounded but an operator acting on it may still be compact. If the domain is bounded and an operator acting upon it is compact then the entire image space is precompact.

So a bounded (continuous) linear operator from one Hilbert space to another,

$$T : L^2(\mathbb{R}^{\mathcal{X}}) \to L^2(\mathbb{R}^{\mathcal{X}})$$

is compact if for *every* bounded subset S of the domain $L^2(\mathbb{R}^{\mathcal{X}})$, the closure of the image space

$$\overline{\{(Tf) : f \in S\}} \subset L^2(\mathbb{R}^{\mathcal{X}})$$

is compact.

Definition 2.3.3 (Self-Adjoint Operators) *A linear operator T is said to be self-adjoint if it is equal to its Hermitian adjoint T^* which satisfies the following:*

$$\langle Th, g\rangle = \langle h, T^* g\rangle$$

All the eigenvalues of a self-adjoint operator are real. In the finite dimensional case, a self-adjoint operator (matrix) T is conjugate symmetric.

By the Reisz Representation Theorem we can show the existence of the adjoint for every operator T that defines a bounded (continuous) linear functional $\mathcal{F} : h \mapsto \langle g, Th\rangle,\ \forall h, g \in \mathcal{H}$:

$$\exists r_{\mathcal{F}} \in \mathcal{H}\ :\ \mathcal{F}(h) = \langle g, Th\rangle = \langle r_{\mathcal{F}}, h\rangle,\ \forall h \in \mathcal{H}$$

so we can define the adjoint as $T^*g = r_{\mathcal{F}}$. We will now characterize and show the existence of the basis of the image space of a compact, self-adjoint operator.

THEOREM 2.3.2 (THE SPECTRAL THEOREM) *Every compact, self-adjoint operator $T : \mathcal{H}_D \to \mathcal{H}_R$ when applied to a function in a Hilbert space $f \in \mathcal{H}$ has the following decomposition:*

$$Tf = \sum_{i=1}^{\infty} \alpha_i P_{\mathcal{H}_i}[f] \in \mathcal{H} \tag{2.31}$$

where each α_i is a complex number and each $\mathcal{H}_i$ is a closed subspace of $\mathcal{H}_D$ such that $P_{\mathcal{H}_i}[f]$ is the orthogonal projection of f onto $\mathcal{H}_i$.

The direct sum of these complementary (orthogonal) subspaces (excluding the null space or zero eigenspace $\mathcal{H}_0$ of the domain) equals the image space of the operator:

$$\mathcal{H}_R = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \mathcal{H}_3 \oplus \cdots$$

When the operator T induces the following decomposition:

$$T\varsigma_i = \upsilon_i \varsigma_i \tag{2.32}$$

we call ς_i an eigenfunction and υ_i an eigenvalue of the operator. The eigenfunctions of T form a complete, countable orthonormal basis of the image space: hence each $\mathcal{H}_i$ has a basis of eigenfunctions all with the same eigenvalue; so we can rewrite the decomposition as follows:

$$Tf = \sum_{j=1}^{\infty} \upsilon_j P_{\varsigma_j}[f] \tag{2.33}$$

where $P_{\varsigma_j}[f]$ is now the projection of f onto the (normalized) eigenfunction ς_j. Different subspaces have different eigenvalues whose associated eigenfunctions

are orthogonal:

$$\mathcal{H}_i \neq \mathcal{H}_j \implies \upsilon_i \neq \upsilon_j \implies \langle \varsigma_i , \varsigma_j \rangle_{\mathcal{H}} = 0$$

The reverse is however not true; two orthogonal eigenfunctions may have the same eigenvalue and be basis vectors for the same subspace. When the domain of the operator $\mathcal{H}$ is a finite n-dimensional space then there are n eigenfunctions and associated eigenvalues. When the operator is positive then [Rud91] the eigenvalues are positive and absolutely convergent (elements of ℓ^1 so that they decrease to zero).

As an example let us consider a single function in the domain $f \in L^2(\mathcal{X})$ and take a bounded subspace $\mathcal{B}$ around it, for example the ball of unit length:

$$\mathcal{B} = \{g \in L^2(\mathcal{X}) : \|f - g\|_{L^2} \leq 1\}$$

Then application of the *compact* operator T to elements in this bounded subspace $\mathcal{B}$ yields an image space whose closure is compact and hence finite-dimensional. So applying T to any function in $\mathcal{B}$ yields a function which can be decomposed into a *finite* linear combination of orthogonal basis vectors in the form (2.31) or (2.33).

2.3.8 Integral Operators

Essentially, what we would like to achieve is the transformation of a function from a space where it is difficult to manipulate to a space where it can be represented as a sum of simple functions which are easier to manipulate. An associated inverse transform, if it exists, can then transform the function back into its original space. We begin by defining this transformation operator and its associated kernel:

Definition 2.3.4 (Integral Operator) *A linear operator $T_K : L^2(\mathcal{X}) \to L^2(\mathcal{X})$ is integral if for a given kernel function $K \in L_\infty(\mathcal{X} \times \mathcal{X})$ the following transformation of one function space into another holds almost everywhere for all $f \in L^2(\mathcal{X})$:*

$$(T_K f)(\cdot) = \int_{\mathcal{X}} K(\cdot, \vec{x}) f(\vec{x}) d\mu(\vec{x}) \tag{2.34}$$

where μ is the Lebesgue measure.

When the image space is finite-dimensional, the integral transformation T_K changes the representation of the input function f to an output function

$(T_K f)$ expressed as a linear combination of a finite set of orthogonal basis functions:

$$(T_K f) = \sum_{i=1}^{b} \alpha_i f_i \text{ such that } \langle f_i, f_j \rangle = 0 \ \ \forall i, j < b \tag{2.35}$$

Definition 2.3.5 (Positive Kernel) *A function $K \in L_\infty(\mathcal{X} \times \mathcal{X})$ such that any quadratic form over it is positive:*

$$\int_{\mathcal{X}} \int_{\mathcal{X}} K(\vec{x}, \vec{y}) \, \varsigma(\vec{x}) \, \varsigma(\vec{y}) \, d\mu(\vec{x}) \, d\mu(\vec{y}) > 0 \quad \forall \varsigma \in L^2(\mathcal{X})$$

is called a positive kernel.

It is easy to see that when a finite kernel is positive-definite over all possible finite sets of vectors in the space $\mathcal{X} \times \mathcal{X}$ then the kernel is positive; furthermore if all functions in the domain are positive ($f > 0$) then the integral operator is also positive $Tf > 0$ and vice versa.

Definition 2.3.6 (Continuous Kernel) *A function $K \in C^0(\mathcal{X} \times \mathcal{X})$ is continuous at a point $(\vec{b}, \vec{c}) \in \mathcal{X} \times \mathcal{X}$ if it satisfies:*

$$\begin{aligned} &\forall \epsilon > 0, \ \exists \delta > 0, \\ &\forall \vec{x}, \vec{s} \in \mathcal{X}, \ \vec{b} - \delta < \vec{x} < \vec{b} + \delta, \ \vec{c} - \delta < \vec{s} < \vec{c} + \delta \\ &\implies K(\vec{b}, \vec{c}) - \epsilon < K(\vec{x}, \vec{s}) < K(\vec{b}, \vec{c}) + \epsilon \end{aligned} \tag{2.36}$$

If the kernel K is symmetric, then the integral operator T_K (2.34) must be self-adjoint. To see this, consider two hypothesis functions $f, g \in \mathcal{H}$:

$$\begin{aligned} \langle (T_K f), g \rangle_{L^2} &= \int_{\mathcal{X}} g(\vec{y}) \left(\int_{\mathcal{X}} K(\vec{y}, \vec{x}) f(\vec{x}) d\mu(\vec{x}) \right) d\mu(\vec{y}) \\ &= \int_{\mathcal{X}} \int_{\mathcal{X}} g(\vec{y}) K(\vec{y}, \vec{x}) f(\vec{x}) d\mu(\vec{x}) d\mu(\vec{y}) \\ &= \int_{\mathcal{X}} \int_{\mathcal{X}} g(\vec{y}) K(\vec{y}, \vec{x}) f(\vec{x}) d\mu(\vec{y}) d\mu(\vec{x}) \\ &= \int_{\mathcal{X}} f(\vec{x}) \left(\int_{\mathcal{X}} K(\vec{x}, \vec{y}) g(\vec{y}) d\mu(\vec{y}) \right) d\mu(\vec{x}) \\ &= \langle f, (T_K g) \rangle_{L^2} \end{aligned}$$

where the third equality (switching the order of integration) follows from applying Fubini's Theorem. Assume further that the kernel K is continuous

$K \in C^0(\mathcal{X} \times \mathcal{X})$:

$$\int_{\mathcal{X}\times\mathcal{X}} K(\vec{x}, \vec{y})^2 d\mu(\vec{x}) d\mu(\vec{y}) < \infty$$

Now for any bounded subspace of the domain $\mathcal{X} \times \mathcal{X}$ one can show that the image space under the operator T_K is precompact in $L^2(\mathcal{X})$ and hence that the integral operator T_K defined in (2.34) is compact.

So when the kernel K is positive, symmetric and square integrable the resulting integral operator T_K is positive, self-adjoint and compact. It therefore follows from the Spectral Decomposition Theorem that T_K must have a countable set of non-negative eigenvalues; furthermore, the corresponding eigenfunctions $\{\varsigma_1, \varsigma_2, \cdots\}$ must form an orthonormal basis[5] for $L^2(\mathcal{X})$ assuming they have been normalized $\|\varsigma_i\|_{L^2} = 1$.

THEOREM 2.3.3 (MERCER'S THEOREM) *For all positive (2.3.5), symmetric and continuous (2.37) kernel functions $K \in L^2(\mathcal{X}\times\mathcal{X})$ over a compact domain $\mathcal{X} \times \mathcal{X}$, defining a positive, self-adjoint and compact integral operator T_K with an eigen-decomposition (2.32) the following five conditions are satisfied:*

1. $\{\upsilon_1, \upsilon_2, \cdots\} \in l_1$ *: the sequence of eigenvalues are absolutely convergent*
2. $\upsilon_i > 0$, $\forall i$ *: the eigenvalues are strictly positive*
3. $\varsigma_i \in L_\infty(\mathcal{X})$ *: the individual eigenfunctions $\varsigma_i : \mathcal{X} \to \mathbb{R}$ are bounded.*
4. $\sup_i \|\varsigma_i\|_{L_\infty} < \infty$ *: the set of all eigenfunctions is also bounded*
5. $\forall \vec{s}, \vec{x} \in \mathcal{X}$: $K(\vec{s}, \vec{x}) = \sum_{i=1}^{\infty} \upsilon_i \, \varsigma_i(\vec{s}) \, \varsigma_i(\vec{x}) = \langle \Phi(\vec{s}), \Phi(\vec{x}) \rangle_{L^2}$

where (5) converges absolutely for each $(\vec{x}, \vec{y}) \in \mathcal{X} \times \mathcal{X}$ and therefore converges uniformly for almost all $(\vec{x}, \vec{y}) \in \mathcal{X} \times \mathcal{X}$.

Proof Since T_K is a compact operator we can apply the Spectral Decomposition Theorem which guarantees the existence of an orthonormal basis (eigen-decomposition) in terms of eigenfunctions and eigenvalues:

$$T\varsigma_i(\vec{s}) = \int_{\mathcal{X}} K(\vec{t}, \vec{s}) \varsigma_i(\vec{t}) d\mu(\vec{t}) = \upsilon_i \varsigma_i(\vec{s})$$

[5] Strictly speaking, the eigenfunctions span a dense subset of $L^2(\mathcal{X})$.

Since the eigenfunctions form an orthonormal basis for $L^2(\mathcal{X})$, it follows that

$$\|\varsigma_i\|_{L^2} = \int_{\mathcal{X}} \varsigma_i(\vec{x})^2 d\mu(\vec{x}) = 1$$

(1) easily follows from (5) and the boundedness (which is implied by continuity over a compact domain) of the kernel function $K \in L_\infty(\mathcal{X} \times \mathcal{X})$; integrating both sides of the kernel expansion in (5) and taking $\vec{s} = \vec{x}$ gives:

$$\sum_{i=1}^{\infty} \upsilon_i \int_{\mathcal{X}} \varsigma_i(\vec{x})^2 d\mu(\vec{x}) = \sum_{i=1}^{\infty} \upsilon_i = \int_{\mathcal{X}} K(\vec{x}, \vec{x}) d\mu(\vec{x}) < \infty$$

(2) follows from the positivity of the integral operator T_K which is implied by the positivity of the kernel function.

(3) and (4) follow from the continuity of the kernel and the eigenfunctions over a compact domain; if $\upsilon_i \neq 0$ then its associated eigenfunctions are continuous on $\mathcal{X}$ since:

$$\begin{aligned}
\forall \epsilon > 0,\ \exists \delta > 0,\ :\ |\vec{x} - \vec{y}| &< \delta \implies \qquad (2.37)\\
|\varsigma_i(\vec{x}) - \varsigma_i(\vec{y})| &= \frac{1}{|\upsilon_i|} \left| \int_{\mathcal{X}} \left(K(\vec{s}, \vec{x}) - K(\vec{s}, \vec{y})\right) \varsigma_i(\vec{s}) d\mu(\vec{s}) \right| \\
&\leq \frac{1}{|\upsilon_i|} \int_{\mathcal{X}} |K(\vec{s}, \vec{x}) - K(\vec{s}, \vec{y})|\, |\varsigma_i(\vec{s})| d\mu(\vec{s}) \\
&\leq \frac{\sup_i \|\varsigma_i\|_{L_\infty}}{|\upsilon_i|} \int_{\mathcal{X}} |K(\vec{s}, \vec{x}) - K(\vec{s}, \vec{y})|\, d\mu(\vec{s}) \\
&\leq \epsilon
\end{aligned}$$

where the last inequality follows from the continuity of K so that the difference $|K(\vec{s}, \vec{x}) - K(\vec{s}, \vec{y})|$ can be made arbitrarily small.

We can bound the following infinite sum, a proof of which is found in [Hoc73], which implies the absolute convergence in (5):

$$\sum_{i=1}^{\infty} \upsilon_i |\varsigma_i(t)\varsigma_i(s)| = \sum_{i=1}^{\infty} \frac{1}{|\upsilon_i|} \left| \int_{\mathcal{X}} K(\vec{x}, \vec{t}) \varsigma_i(\vec{x}) d\mu(\vec{x}) \int_{\mathcal{X}} K(\vec{x}, \vec{s}) \varsigma_i(\vec{x}) d\mu(\vec{x}) \right|$$

□

2.3.9 Reproducing Kernel Hilbert Spaces

A Reproducing Kernel Hilbert Space (RKHS) is the 'working' hypothesis (function) space for Support Vector Machine algorithms; elements from the observation space are mapped into a RKHS, in which the structure necessary

to define (and then solve) a given discriminative or regression problem already exists. Any observations can be transformed into features in a RKHS and hence there exists a universal representational space for any given set from the observation space. The explicit form the features take are as a kernelized distance metric between any two observations which implicitly can be expressed as an inner product; essentially a RKHS combines a (restricted) Hilbert Space with an associated positive kernel function (definition 2.3.5).

Definition 2.3.7 (Reproducing Kernel Hilbert Space) *A Hilbert space* $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ *that is point-wise defined (on* $\mathbb{R}^{\mathcal{X}}$*) and where every evaluation functional* $\mathcal{E}_t[f] : \mathcal{H}(\mathcal{X}) \to \mathbb{R}$ *is continuous is a Reproducing Kernel Hilbert Space (RKHS).*

Hence all point-wise evaluations are bounded and then by the Reisz Representer Theorem (2.3.1) every function evaluation at some fixed point $\vec{x} \in \mathcal{X}$ has a fixed representer function $r_{\mathcal{E}x} \in \mathcal{H}_K$ essentially satisfying (2.16).

It is easy to show that norm convergence in a RKHS always implies point-wise convergence and vice versa:

$$\|f_n - f\|_{\mathcal{H}} \to 0 \iff \lim_{n\to\infty} f_n(\vec{x}) = \lim_{n\to\infty} \mathcal{E}_{\vec{x}}(f_n) = \lim_{n\to\infty} \mathcal{E}_{\vec{x}}(f) = f(\vec{x}),\ \forall \vec{x} \in \mathcal{X} \tag{2.38}$$

where the second equality on the right follows from the continuity of the evaluation functional and the assumption that f_n converges to f in norm. Recall that point-wise convergence (2.6) was the second of two restrictions deemed necessary for all functions in the hypothesis space.

Definition 2.3.8 (Reproducing Kernel) *A kernel function* K *of a Hilbert space* $L^2(\mathcal{X} \times \mathcal{X})$ *that satisfies the following for all* $\vec{x} \in \mathcal{X}$*:*

1. $K_{\vec{x}} \in \mathcal{H}$ *: the kernel fixed at some point* $\vec{x} \in \mathcal{X}$ *is a function over a Hilbert space*
2. $\forall f \in \mathcal{H}$ *the reproducing property is satisfied*

$$\langle f, K_{\vec{x}} \rangle = f(\vec{x})$$

and in particular when $f = K_{\vec{s}}$ *:*

$$\langle K_{\vec{s}}, K_{\vec{x}} \rangle = K_{\vec{s}}(\vec{x}) = K_{\vec{x}}(\vec{s}) = K(\vec{s}, \vec{x})$$

So by definition the reproducing kernel is such that for all vectors in the input space $\vec{x} \in \mathcal{X}$, the function $K_{\vec{x}}$ is the unique representer for the evaluation functional $\mathcal{E}_{\vec{x}}(f)$.

$$\forall \vec{x} \in \mathcal{X},\ \exists K_{\vec{x}} \in \mathcal{H}_K\ :\ f(\vec{x}) = \mathcal{E}_{\vec{x}}[f] = \langle K_{\vec{x}}, f \rangle_{\mathcal{H}_K} = \mathcal{G}_{r_{\mathcal{E}}}(f),\ \ \forall f \in \mathcal{H} \quad (2.39)$$

The only difference between (2.16) and (2.39) is that the latter requires the representer to have the form of a kernel function $r_{\mathcal{E}x} = K_{\vec{x}} = K(\vec{x}, \cdot)$ fixed in its first argument at some point in the input space. Therefore it follows that every function in a RKHS can be represented point-wise as an inner product whose first argument is always taken from the same set $\{K_{\vec{x}_1}, K_{\vec{x}_2}, K_{\vec{x}_3}, \cdots\}$ of distinct (representer) kernel functions and whose second argument is the function itself.

Theorem 2.3.4 (Moore-Aronszajn Theorem) *Every positive-definite kernel $K(\cdot, \cdot)$ on $\mathcal{X} \times \mathcal{X}$ is a reproducing kernel for some unique RKHS of functions on $\mathcal{X}$. Conversely, every RKHS has an associated unique positive-definite kernel whose span is dense in it. In short, there exists a bijection between the set of all reproducing kernel Hilbert spaces and the set of all positive kernel functions.*

Proof Given a RKHS $\mathcal{H}_K$, by the Reisz Representation Theorem there exists a representer in $\mathcal{H}_K$ for all evaluation functionals (which are continuous by definition of a RKHS) over $\mathcal{H}_K$; the representer is given by $K_{\vec{x}}$ (see 2.42 or 2.46) and the reproducing kernel (which can be shown to be positive and unique) is therefore given by

$$K(\vec{x}, \vec{s}) = \langle K_{\vec{x}}, K_{\vec{s}} \rangle_{\mathcal{H}_K},\ \forall \vec{s} \in \mathcal{X} \quad (2.40)$$

Conversely, given a positive kernel K we define a set of functions $\{K_{\vec{x}_1}, K_{\vec{x}_2}, \cdots\}$ for each $\vec{x}_i \in \mathcal{X}$ and then define the elements of the RKHS as the point-wise defined functions in (the completion of) the space spanned by this set:

$$\mathcal{H}_K = \left\{ f \in \mathbb{R}^{\mathcal{X}}\ :\ f = \sum_{\vec{x}_i \in \mathcal{X}} \alpha_i K_{\vec{x}_i},\ \|f\|_{\mathcal{H}_K} < \infty,\ \alpha_i \in \mathbb{R} \right\} \quad (2.41)$$

The reproducing property is satisfied in this space:

$$\begin{aligned} \langle K_{\vec{s}}, f\rangle_{\mathcal{H}_K} &= \left\langle K_{\vec{s}}\,,\ \sum_j \beta_j K_{\vec{t}_j} \right\rangle_{\mathcal{H}_K} \\ &= \sum_j \beta_j \langle K_{\vec{s}}, K_{\vec{t}_j}\rangle_{\mathcal{H}_K} \\ &= \sum_j \beta_j K(\vec{s}, \vec{t}_j) \\ &= f(\vec{s}) \end{aligned} \tag{2.42}$$

so that $K_{\vec{s}}$ is in fact the representer of the evaluation functional $\mathcal{E}_{\vec{s}}(\cdot)$. Evaluation functionals in this space are necessarily bounded and therefore continuous:

$$|\mathcal{E}_{\vec{x}}(f)| = |f(\vec{x})| = |\langle K_{\vec{x}}, f\rangle| \leq \|K_{\vec{x}}\|_{\mathcal{H}}\|f\|_{\mathcal{H}} = \alpha\|f\|_{\mathcal{H}}$$

where the second equality is due to the reproducing property of the kernel and the third inequality is due to the Cauchy-Schwarz Inequality. Norms in this space $\|\cdot\|_{\mathcal{H}_K}$ are induced by the inner (dot) product which is defined as follows:

$$\begin{aligned} \langle f, g\rangle_{\mathcal{H}_K} &= \left\langle \sum_i \alpha_i K_{\vec{x}_i},\ \sum_j \beta_j K_{\vec{x}_j} \right\rangle_{\mathcal{H}_K} \\ &\equiv \sum_i \sum_j \alpha_i \beta_j K(\vec{x}_i, \vec{x}_j) \end{aligned} \tag{2.43}$$

which can easily be shown to be symmetric and linear when the kernel is positive.

We complete the space spanned by the kernel function K by adding to it the limit functions of all Cauchy sequences of functions, if they are not already within the space. The limit functions that must be added (and which can therefore not be expressed as a linear combination of the kernel basis functions, i.e. the span of the kernel is dense in the space) must be point-wise well defined. However we have already seen that in a RKHS, norm convergence (and in particular Cauchy convergence) implies point-wise convergence so that the limit function is always point-wise well defined; so all Cauchy sequences converge point-wise to limit functions whose addition to the space completes it. □

So given any positive-definite kernel function we can construct its associated unique reproducing kernel Hilbert space and vice versa. As an example

let us consider the Hilbert space L^2 that contains functions that have discontinuities (evaluation functionals are therefore not bounded and hence not continuous and so it is not a RKHS) of measure zero and are therefore not smooth, as are all the elements of C^∞ which is however not a Hilbert space; hence we seek to restrict the Hilbert space L^2, removing all functions that are not smooth as well as some that are, ensuring that the resulting space is still Hilbertian. Define L^2_K as the subspace of L^2 that includes the span of the functions $K_{\vec{x}},\ \vec{x} \in \mathcal{X}$ as well as their point-wise limits. The resulting space is Hilbertian. If the kernel reproduces in the space and is bounded then L^2_K is a reproducing kernel Hilbert space.

Alternatively, we can construct a RKHS by using Mercer's Decomposition (Condition 5 of 2.3.3); consider the space spanned by the eigenfunctions (which have non-zero eigenvalues) of the eigendecomposition of the integral operator defined using some kernel K:

$$\mathcal{H}_K = \left\{ f \in \mathbb{R}^{\mathcal{X}} \ : \ f = \sum_{i=1}^{\infty} \alpha_i \varsigma_i,\ \|f\|_{\mathcal{H}_K} < \infty,\ \alpha_i \in \mathbb{R}, \varsigma_i \in L_\infty(\mathcal{X}) \right\} \quad (2.44)$$

so that the dimension of the space $\mathcal{H}_K$ is equal to the number of non-zero eigenvalues of the integral operator. Then define the norm on this RKHS in terms of an inner product:

$$\begin{aligned} \langle f, g \rangle_{\mathcal{H}_K} &= \left\langle \sum_{i=1}^{\infty} \alpha_i \varsigma_i, \sum_{i=1}^{\infty} \beta_i \varsigma_i \right\rangle_{\mathcal{H}_K} \\ &\equiv \sum_{i=1}^{\infty} \frac{\alpha_i \beta_i}{\upsilon_i} \end{aligned} \quad (2.45)$$

It then follows from Mercer's Theorem that the function $K_{\vec{x}}$ is a representer of the evaluation functional $\mathcal{E}_{\vec{x}}$ and therefore reproduces in the RKHS $\mathcal{H}_K$:

$$\begin{aligned} \langle f(\cdot), K_{\vec{x}}(\cdot) \rangle_{\mathcal{H}_K} &= \left\langle \sum_{i=1}^{\infty} \alpha_i \varsigma_i(\cdot), \sum_{i=1}^{\infty} \upsilon_i\ \varsigma_i(\vec{x})\ \varsigma_i(\cdot) \right\rangle_{\mathcal{H}_K} \\ &\equiv \sum_{i=1}^{\infty} \frac{\alpha_i \upsilon_i \varsigma_i(\vec{x})}{\upsilon_i} \\ &= \sum_{i=1}^{\infty} \alpha_i \varsigma_i(\vec{x}) \\ &= f(\vec{x}) \end{aligned} \quad (2.46)$$

So instead of minimizing the *regularized risk functional* over all functions in the hypothesis space:

$$f^* = \arg\inf_{f \in \mathcal{H}} \left\{ \sum_{i=1}^{n} \ell(f, \{\vec{x}_i, \vec{y}_i\}) + \lambda \|f\|^2_{\mathcal{H}_K} \right\} \tag{2.47}$$

we can minimize the following functional over all sequences of expansion coefficients $\{\alpha_1, \alpha_2, \cdots\}$:

$$f^* = \arg\inf_{\{\alpha_1, \alpha_2, \cdots\}} \left\{ \sum_{i=1}^{n} \ell \left(\sum_{j=1}^{\infty} \alpha_j \varsigma_j(\cdot), \{\vec{x}_i, \vec{y}_i\} \right) + \lambda \sum_j \frac{\alpha_j^2}{\upsilon_j} \right\} \tag{2.48}$$

which follows from (2.44) and (2.45). The number of expansion coefficients is equal to the number of non-zero eigenvalues which is also the dimension of the RKHS constructed in (2.44); since this number is possibly infinite the above optimization is possibly infeasible.

More generally we can construct a RKHS by completing the span of any basis set. The RKHS constructions (2.41) and (2.44) are equivalent (see [CS02] for a proof). The inner products defined in (2.45) and (2.43) can also be shown to be equivalent.

2.4 RKHS and Function Regularity

Now that we have introduced the RKHS family of hypothesis spaces we introduce some further restrictions and discuss why they are necessary. The hypothesis that the learning algorithm selects will need to conform to three basic criteria:

Definition 2.4.1 (Well-Posed Optimization) *An optimization Ψ is well-posed provided the solution $f^* : \mathcal{X} \to \mathcal{Y}$:*

1. *Exists: if the hypothesis space is too small then the solution may not exist.*

$$\exists \hat{f}^* \in \mathcal{H} \ : \ f^* = \arg\inf_{f \in \mathcal{H}} \Psi$$

2. *is Unique: if the hypothesis space is too large or the training set is too small then the solution may not be unique.*

$$\forall \hat{f}_1^*, \hat{f}_2^* \in \mathcal{H} \ : \ \hat{f}_1^*, \hat{f}_2^* = \arg\inf_{f \in \mathcal{H}} \Psi \implies \hat{f}_1^* = \hat{f}_2^*$$

3. *is Stable: f^* depends continuously on the training set, so that slight perturbations in the training set do not affect the resulting solution, especially as the number of training examples gets larger.*

As we will see in the following chapter, the prediction function output by the learning algorithm must be generalizable *and* well-posed. The third criterion above is especially important as it relates to the generalization ability of a hypothesis: a stable transform is less likely to overfit the training set.

The ERM principle guarantees the existence of a solution assuming $\mathcal{H}$ is compact and the loss function ℓ (and hence the empirical risk $\hat{R}_n$) is continuous; in general neither of these conditions are satisfied. ERM does not however guarantee the uniqueness (all functions that achieve the minimum empirical risk are in the same equivalence class but there is only one amongst this class that generalizes well) or the stability (removing a single example from the training set will give rise to a new prediction function that is fundamentally different) of the solution; the method is therefore ill-posed.

We must resort to using prior information to determine which solution from within the equivalence class of functions of minimal empirical risk is best suited for prediction. This can be done for example by constraining the capacity of the hypothesis space. We will consider two regularization methods that attempt to do this, thereby ensuring the uniqueness and stability of the solution. The question of how to constrain the hypothesis space is answered by Occam's Razor which essentially states that the simplest solution is often the best, given that all other variables (i.e. the empirical risk) remain constant.

So in a nutshell, regularization attempts to provide well-posed solutions to a learning task, specifically ERM, by constraining the capacity of the hypothesis space through the elimination of complex functions that are unlikely to generalize, thereby isolating a unique and stable solution.

We can explicitly constrain the capacity of the hypothesis space (Ivanov Regularization) or implicitly optimize a parameter (Tikhonov Regularization)

that regulates the capacity of the hypothesis space. Both methods are equivalent[6] and make use of a measure of the "smoothness"[7] of a function to regulate the hypothesis space. It is easy to show that the norm functional serves as an appropriate measure of smoothness given that the associated kernel serves as an appropriate measure of similarity.

DEFINITION 2.4.2 (LIPSCHITZ CONTINUITY) *A map $f : \mathcal{X} \to \mathcal{Y}$ is Lipschitz continuous if it satisfies:*

$$|f(\vec{x}_1) - f(\vec{x}_2)| \leq M|\vec{x}_1 - \vec{x}_2|$$

The smallest $M \geq 0$ that satisfies the above inequality for all $\vec{x}_1, \vec{x}_2 \in \mathcal{X}$ is called the Lipschitz constant of the function. Every Lipschitz continuous map is uniformly continuous which is a stronger condition than simple continuity.

Functions in a RKHS are Lipschitz continuous; take two points in the domain $\vec{x}_1, \vec{x}_2 \in \mathcal{X}$ then from the Reisz Representation Theorem it follows that:

$$\begin{aligned} |f(\vec{x}_1) - f(\vec{x}_2)| &= |\langle f, K_{\vec{x}_1}\rangle_{\mathcal{H}_\mathcal{K}} - \langle f, K_{\vec{x}_2}\rangle_{\mathcal{H}_\mathcal{K}}| \\ &= |\langle f, K_{\vec{x}_1} - K_{\vec{x}_2}\rangle_{\mathcal{H}_\mathcal{K}}| \\ &\leq \|f\|_{\mathcal{H}_\mathcal{K}}(K_{\vec{x}_1} - K_{\vec{x}_2})^2 \end{aligned} \tag{2.49}$$

where the Lipschitz constant is given by the norm of the function $M = \|f\|_{\mathcal{H}_\mathcal{K}}$ and the distance between two elements in the domain is given by the square of the difference of their kernelized positions. As the Lipschitz constant (in this case the norm of the function) decreases, the function varies less in the image space for similar (as measured by the kernel) points in the domain. This justifies the use of the norm in the regularized risk functional defined in (2.47) and now used in the following regularization methods.

---

[6] The Lagrange multiplier technique (5.1) reduces an Ivanov Regularization with constraints to a Tikhonov Regularization without constraints

[7] Intuitively, a function is smooth when the variance in the image space is *slow* for points in the domain that are similar. The similarity of points in a RKHS can naturally be measured by the associated kernel function (2.49).

2.4.1 Ivanov Regularization

Ivanov Regularization requires that all functions in the hypothesis space $f \in \mathcal{H}_{\mathcal{T}}$, of which there might be an infinite number, exist in a $\mathcal{T}$-bounded subset of a RKHS $\mathcal{H}_K$:

$$\hat{f}^* = \arg\inf_{f \in \mathcal{H}} \hat{R}_n[f] \text{ subject to } \|\mathcal{H}\|_{\mathcal{H}_K} \leq \mathcal{T} \tag{2.50}$$

Another way to see why this works is to consider functions from two hypothesis spaces, one significantly less complex (functions are smoother) than the other;

$$\mathcal{H}_{\mathcal{T}_i} = \{f : f \in \mathcal{H}_K \text{ and } \|f\|^2_{\mathcal{H}_K} \leq \mathcal{T}_i\}, \;\; i \in \{1, 2\}, \; \mathcal{T}_1 \ll \mathcal{T}_2$$

Small perturbations in the training data cause prediction functions from the more complex class $\mathcal{H}_{\mathcal{T}_2}$ to fluctuate more whereas functions from the smoother class $\mathcal{H}_{\mathcal{T}_1}$ remain relatively *stable*. In [Rak06] we also see that for ERM in particular, stability and consistency (3.13) are in fact equivalent. Furthermore, a *bounded, finite-dimensional* RKHS $\mathcal{H}_{\mathcal{T}_i}$ is a totally bounded space and hence must have a finite epsilon-net (definition 3.4.1) which implies the covering number (definition 3.4.3) of $\mathcal{H}_{\mathcal{T}_i}$ may be used in deriving generalization bounds. Yet there is no specified methodology for choosing the value of $\mathcal{T}$ and so we must resort to using another related regularization technique.

2.4.2 Tikhonov Regularization

The Tikhonov Regularization differs in that it penalizes the complexity and instability of the hypothesis space in the objective function of the optimization instead of explicitly bounding it by some constant;

$$\hat{f}^* = \arg\inf_{f \in \mathcal{H}, \lambda} \left\{ \hat{R}_n[f] + \lambda \|f\|^2_{\mathcal{H}_K} \right\} \tag{2.51}$$

where λ is a *regularization parameter* that must also be optimized to ensure optimal generalization performance as well as the stability and uniqueness of the solution [Rak06]. In the following theorem we see that although the hypothesis space is potentially an infinite dimensional Hilbert function space, the solution of the Tikhonov optimization has the form of a finite basis expansion.

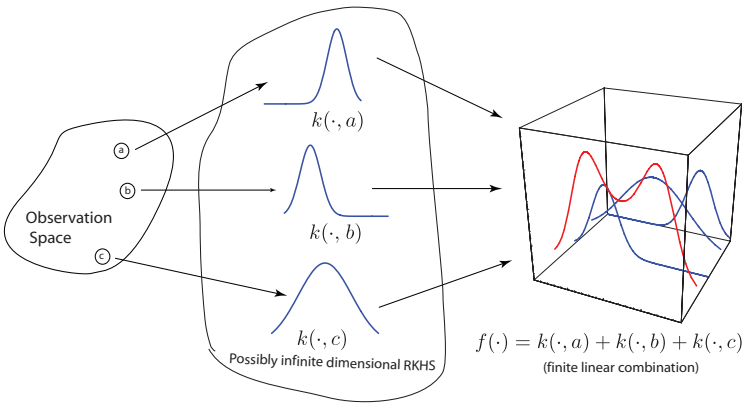

Figure 2–3: Each training data point is mapped to a basis function (in blue) which can then be used to define the solution (in red) as a linear combination of the basis functions.

THEOREM 2.4.1 (REPRESENTER THEOREM) *Consider the objective function of the Tikhonov Regularization Method that optimizes the sum of a loss function and a regularization term:*

$$f^* = \arg\inf_{f \in \mathcal{H}} \left\{ \sum_{i=1}^{n} \ell(f, \{\vec{x}_i, \vec{y}_i\}) + \Upsilon(\|f\|^2_{\mathcal{H}}) \right\}$$

Then if ℓ is a point-wise defined loss function (i.e. $\forall\{\vec{x}_i, y_i\} \in \mathcal{S} : \ell(f, \{\vec{x}_i, \vec{y}_i\}) \in \mathbb{R}$) and Υ is monotonically increasing then the solution to the optimization exists and can be written as a linear combination of a finite set of functions defined over the training data;

$$f^* = \sum_{j=1}^{n} \alpha_j K_{\vec{x}_j}$$

where $K_{\vec{x}_j}$ is the representer of the (bounded) evaluation functional $\mathcal{E}_{\vec{x}_j}(f) = f(\vec{x}_j)$ for all $f \in \mathcal{H}$.

Proof The functions $K_{\vec{x}_i}$, $\forall \vec{x}_i \in \mathcal{S}$ span a subspace of $\mathcal{H}$:

$$\mathcal{U} = \mathtt{span}\{K_{\vec{x}_i} : 1 \leq i \leq n\} = \left\{ f \in \mathcal{H} : f = \sum_{i=1}^{n} \alpha_i K_{\vec{x}_i} \right\}$$

Denote by $P_{\mathcal{U}}$ the projection that maps functions from $\mathcal{H}_K$ onto $\mathcal{U}$, then any function $P_{\mathcal{U}}[f]$ can be represented as a finite linear combination:

$$\forall P_{\mathcal{U}}[f] \in \mathcal{U} : P_{\mathcal{U}}[f] = \sum_{i=1}^{n} \alpha_i K_{\vec{x}_i}$$

Hence any function $f \in \mathcal{H}$ can be represented as:

$$f = P_{\mathcal{U}}[f] + (I - P_{\mathcal{U}})[f] = \sum_{i=1}^{n} \alpha_i K_{\vec{x}_i} + (I - P_{\mathcal{U}})[f]$$

where $(I - P_{\mathcal{U}})$ is the projection of functions in $\mathcal{H}$ onto $\mathcal{U}^\top$ whose elements are orthogonal to those in $\mathcal{U}$. Now applying the reproducing property of a RKHS and noting that the function $K_{\vec{x}_j}$ is orthogonal to all vectors in $\mathcal{U}^\top$:

$$\begin{aligned}
f(\vec{x}_j) &= \langle f, K_{\vec{x}_j} \rangle_{\mathcal{H}} \\
&= \left\langle \sum_{i=1}^{n} \alpha_i K_{\vec{x}_i} + (I - P_{\mathcal{U}})[f], K_{\vec{x}_j} \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} \alpha_i \left\langle K_{\vec{x}_i}, K_{\vec{x}_j} \right\rangle_{\mathcal{H}} + \left\langle (I - P_{\mathcal{U}})[f], K_{\vec{x}_j} \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} \alpha_i \left\langle K_{\vec{x}_i}, K_{\vec{x}_j} \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} \alpha_i K(\vec{x}_i, \vec{x}_j)
\end{aligned}$$

so that the evaluation of functions in the hypothesis space is not dependent on corresponding components in the subspace $\mathcal{U}^\top$ but is dependent on the coefficients $\{\alpha_i, i = 1, \cdots, n\}$ which must be determined. Now since the loss function needs only to be evaluated point-wise over the training set, we can group all functions that have the same point-wise evaluation over $\mathcal{S}$ (and hence the same risk) into an equivalence class:

$$\begin{aligned}
f = g &\iff f(\vec{x}_i) = g(\vec{x}_i),\ \forall \vec{x}_i \in \mathcal{S} \\
&\iff f(\vec{x}_i) = \sum_{j=1}^{n} \alpha_j k(\vec{x}_i, \vec{x}_j) = \sum_{j=1}^{n} \beta_j k(\vec{x}_i, \vec{x}_j) = g(\vec{x}_i),\ \forall \vec{x}_i \in \mathcal{S} \\
&\implies \ell(f, \mathcal{S}) = \ell(g, \mathcal{S}) \\
&\implies \hat{R}_n[f] = \hat{R}_n[g]
\end{aligned}$$

Now for $g \in \mathcal{U}$ and $l \in \mathcal{U}^\top$ such that $f = g + l$ we have:

$$\Upsilon(\|f\|_{\mathcal{H}}^2) = \Upsilon(\|g\|_{\mathcal{H}}^2 + \|l\|_{\mathcal{H}}^2)$$

it then follows that the optimal function within the equivalence class of minimum risk must have $\|l\|_{\mathcal{H}} = 0$ since otherwise it increases $\|f\|_{\mathcal{H}}^2$ (and hence

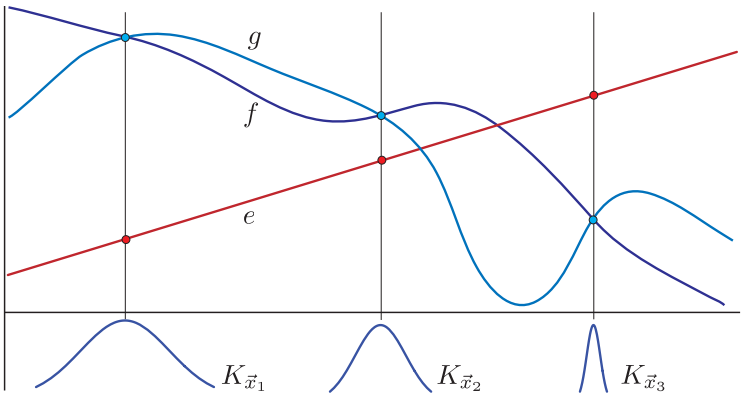

Figure 2–4: Each function $e, f, g \in \mathcal{H}$ has a distinct set of expansion coefficients. However f and g are equivalent in the sense that their function evaluations over the training set are equal: $g(\vec{x}_i) = \sum_{j=1}^{n} \beta_j k(\vec{x}_i, \vec{x}_j) = \sum_{j=1}^{n} \alpha_j k(\vec{x}_i, \vec{x}_j) = f(\vec{x}_i)$.

increases the evaluation of the monotonically increasing function Υ) but leaves the loss unaltered. We can therefore rewrite the objective function as:

$$f^* = \underset{f \in \mathcal{H}, g = P_{\mathcal{U}}[f]}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \ell(g, \{\vec{x}_i, \vec{y}_i\}) + \Upsilon(\|g\|_{\mathcal{H}}^2) \right\}$$

In this way we have linked the search for the global optima in $\mathcal{H}$ with a search for the optimal coefficients $\{\alpha_i, i = 1, \cdots, n\}$ that define a function in the subspace $\mathcal{U}$;

$$f^* = \underset{\{\alpha_1, \alpha_2, \cdots, \alpha_n\}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \ell \left(\sum_{j=1}^{n} \alpha_j K_{\vec{x}_j}, \{\vec{x}_i, \vec{y}_i\} \right) + \Upsilon \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j) \right) \right\} \tag{2.52}$$

In contrast to (2.48), the optimization defined above is feasible as it is performed over a finite number of basis expansion coefficients. So in summary to arrive at a solution in a finite dimensional space $\mathcal{U}$, the optimization first identifies the equivalence class of functions in $\mathcal{H}$ that have minimal risk and then within this class, it identifies the hypothesis whose component in the complementary (orthogonal) subspace $\mathcal{U}^\top$ has a norm equal to zero. □

The solution can also be expressed as a linear combination of a finite number of eigenfunctions as long as they serve as representers for the evaluation functional:

$$f^* = \sum_{j=1}^{m} \beta_j \varsigma_i$$

The solution f^* can then be substituted into the optimization (2.52) so that the values of the expansion coefficients can be numerically calculated; when the loss function is quadratic then this amount to solving a linear system and otherwise a gradient descent algorithm is employed.

So instead of searching through the entire infinite dimensional hypothesis space $\mathcal{H}_K$, as defined in (2.41), we will only consider a finite-dimensional subspace of $\mathcal{U}$ that is spanned by a finite number of basis functions. Within this finite dimensional subspace the solution may still not be unique if we optimized over the loss function alone since there can be several functions that linearly separate (for zero-one (4.1) or hinge loss (4.3) functions) or near-perfectly pass through (for ϵ-insensitive loss (5.1) function) the entire data set to achieve minimal risk; the addition of the regularization term guarantees uniqueness.

2.5 The Kernel Trick

The kernel trick simplifies the quadratic optimizations used in support vector machines by replacing a dot product of feature vectors in the feature space with a kernel evaluation over the input space. Use of the (reproducing) kernel trick can be justified by constructing the explicit map $\Phi : \mathcal{X} \longmapsto \mathbb{R}^{\mathcal{X}}$ in two different ways both of which map a vector $\vec{x} \in \mathcal{X}$ in the input space to a vector in a (feature) reproducing kernel Hilbert space; the first method is derived from the Moore-Aronzajn construction (2.41) of a RKHS and defines the map as:

$$\Phi : \vec{x} \to K_{\vec{x}} \in L^2(\mathcal{X})$$

The reproducing property can then be used to show that the inner product of two functions in the feature (RKHS) space is equivalent to a simple kernel evaluation:

$$\langle \Phi(\vec{x}), \Phi(\vec{s}) \rangle_{\mathcal{H}_K} = \langle K_{\vec{x}}, K_{\vec{s}} \rangle_{\mathcal{H}_K} = K(\vec{x}, \vec{s}) \tag{2.53}$$

The second method is derived from Mercer's Construction (2.44) of a RKHS and defines the map as:

$$\Phi : \vec{x} \to \{\sqrt{\upsilon_1}\varsigma_1(\vec{x}), \sqrt{\upsilon_2}\varsigma_2(\vec{x}), \cdots\} \in \ell^2$$

From condition (5) of Mercer's Theorem it then follows that the L^2 inner product of two functions in the feature space is equivalent to a simple kernel

evaluation:

$$\langle \Phi(\vec{x}), \Phi(\vec{s}) \rangle_{L^2} = \sum \upsilon_i \varsigma_i(\vec{x}) \varsigma_i(\vec{s}) = K(\vec{x}, \vec{s}) \tag{2.54}$$

Mercer's Theorem proves the converse, specifically that a positive, continuous, symmetric kernel can be decomposed into an inner product of infinite-dimensional (implicitly) mapped input vectors.

2.5.1 Kernelizing the Objective Function

As an example let us consider the dual quadratic optimization used in support vector regression (5.16) which includes the inner product $\langle \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \rangle$ in its objective function;

$$\text{maximise} \left\{ \begin{array}{r} -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i - \beta_i)(\alpha_j - \beta_j) \ \langle \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \rangle \\ -\epsilon \sum_{i=1}^{n} (\alpha_i + \beta_i) + \sum_{i=1}^{n} y_i (\alpha_i - \beta_i) \end{array} \right\}$$

$$\text{subject to} \left[\begin{array}{l} \sum_{i=1}^{n} (\alpha_i - \beta_i) = 0 \\ \alpha_i, \beta_i \in [0, \zeta] \end{array} \right.$$

The process of applying the projection or mapping ϕ to each input and then taking inner products between all pairs of inputs is computationally intensive; in cases where the feature space is infinite dimensional it is infeasible; so we substitute a kernel evaluation for this inner product in the objective function of the quadratic program and by Theorem (2.3.3) we see that the inner product is now performed implicitly in the feature space;

$$\text{maximise} \left\{ \begin{array}{r} -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i - \beta_i)(\alpha_j - \beta_j) \ K(\vec{x}_i, \vec{x}_j) \\ -\epsilon \sum_{i=1}^{n} (\alpha_i + \beta_i) + \sum_{i=1}^{n} y_i (\alpha_i - \beta_i) \end{array} \right\}$$

$$\text{subject to} \left[\begin{array}{l} \sum_{i=1}^{n} (\alpha_i - \beta_i) = 0 \\ \alpha_i, \beta_i \in [0, \zeta] \end{array} \right.$$

2.5.2 Kernelizing the Solution

The solution $f(\vec{x}_t)$ to a *kernelized* classification task (4.12) is given in terms of the weight vector $\vec{w}$ (which is orthogonal to the separating hyperplane), which in turn is computed using a constraint derived from the dual form of a quadratic optimization (4.22) and expressed as a linear combination of support vectors (section 4.2.2) which must be mapped (using ϕ) into the feature space:

$$\vec{w} = \sum_{i}^{\#sv} \alpha_i y_i \phi(\vec{x}_i)$$

The hypothesis function can be kernelized (so that prediction is possible even in infinite dimensional spaces) by first mapping the test example $\vec{x}_t$ in its definition using the map ϕ and then substituting a kernel evaluation with the dot-product;

$$\begin{aligned} f(\vec{x}_t) &= \mathtt{sgn}\left(\phi(\vec{x}_t) \cdot \vec{w} + b\right) && (2.55) \\ &= \mathtt{sgn}\left(\phi(\vec{x}_t) \cdot \sum_i \alpha_i y_i \phi(\vec{x}_i) + b\right) \\ &= \mathtt{sgn}\left(\sum_i \alpha_i y_i \ \langle \phi(\vec{x}_t), \phi(\vec{x}_i) \rangle + b\right) && (2.56) \\ &= \mathtt{sgn}\left(\sum_i \alpha_i y_i \ K(\vec{x}_t \cdot \vec{x}_i) + b\right) && (2.57) \end{aligned}$$

We refer to equation (2.55) as the *primal solution*, to equation (2.56) as the *dual solution* and to equation (2.57) as the *kernelized dual solution*. The solution $f(\vec{x}_t)$ to a regression task (5.18) can be kernelized in a similar fashion.

It is important to note that this (2.55 and 2.57) is simply an example that reveals how kernel functions correspond to a specific map into a specific feature space; in general however it is not necessary to know the structure of either the implicit map or feature space associated with a kernel function; so although 'learning' is performed implicitly in a complex non-linear feature space, all computation is performed in the input space; this includes the optimization of all learning parameters as well as the evaluation of the solution.

3

Statistical Learning Theory

In searching for an optimal prediction function the most natural approach is to define an optimization over some measure that gauges the accuracy of admissible prediction functions over the training set $\mathcal{S} = \{\vec{x}_i, y_i\}_{i=1}^n \subset \mathcal{X}$; by applying such a measure or *loss function* $\ell(f, \{\vec{x}, y\})$ to each hypothesis in the hypothesis space $f \in \mathcal{H}$ we get a resulting space of functions known as the *loss class*:

$$\mathcal{L}(\mathcal{H}, \cdot) = \{\ell(f, \cdot) : f \in \mathcal{H}\}$$

Now to test a hypothesis, its performance must be evaluated by some fixed loss function over the entire observation space. However, since the generation of observations is governed by the distribution $P(\vec{x}, y)$, making some observations more likely than others, we will need to integrate with respect to it:

Definition 3.0.1 (The expected risk) *is the average loss or error that a fixed function produces over the observation space $\mathcal{X} \times \mathcal{Y}$, integrated with respect to the distribution of data generation*

$$R_{\mathcal{X}}[f] = \int_{\mathcal{Y}} \int_{\mathcal{X}} \ell(f, \{\vec{x}, y\})\, dP(\vec{x}, y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} \ell(f, \{\vec{x}, y\}) P(\vec{x}, y)\, d\vec{x}\, dy$$

A learning method can now simply minimize the *expected risk* over all measurable functions in the hypothesis space $\mathcal{H}$ for some fixed loss function ℓ:

$$f^* = \arg\inf_{f \in \mathcal{H}} R_{\mathcal{X}}[f] \tag{3.1}$$

to find the function f^* that, in the case of a binary classification task, separates the n positive and negative training examples with minimal expected loss; we

refer to this quantity as the *actual risk* for a given function class:

$$R_A(\mathcal{H}) = \inf_{f \in \mathcal{H}} R_X[f] \tag{3.2}$$

Since $P(\vec{x}, y)$ is unknown and also since annotations are not available for the entire input space (which would make learning quite unnecessary) finding f^* using (3.1) is technically impossible.

The material for this chapter was referenced from [CS02], Chapters 8 and 9 of [Muk07], [Che97], [Zho02], [LV07], [BBL03], [PMRR04], [Rak06], [CST00], [HTH01], [EPP00], [Ama95], [Vap99], [Vap96] and [Vap00].

3.1 Empirical Risk Minimization (ERM)

Since evaluating the expected risk is not possible we can instead try to approximate it; a Bayesian approach attempts to model $P(\vec{x}, y) = P(\vec{x}) \cdot P(y|\vec{x})$ and then estimate it from the training data so that the integration in (3.0.1) is realizable. A frequentist approach uses the mean loss or empirical risk achieved over the training data as an approximation of the expected risk;

$$\hat{R}_n[f] = \frac{1}{n} \sum_{i=1}^{n} \ell(f, \{\vec{x}_i, y_i\}) \tag{3.3}$$

The *Empirical Risk Minimization* (ERM) methodology then minimizes the empirical risk $\hat{R}_n$ in search of a hypothesis, that hopefully has minimized expected risk as well so that it is able to accurately predict the annotations of future test examples that are generated by the same input distribution $P(\vec{x})$ that was used in generating the sample set from which the empirical risk was initially calculated:

$$f_n^* = \arg\inf_{f \in \mathcal{H}} \hat{R}_n[f] \tag{3.4}$$

The remainder of this chapter discusses conditions under which ERM's choice of hypothesis f_n^* is equal to the best possible hypothesis f^*. To begin with we would like to measure the deviation between the expected risk (or test error) of the hypothesis f_n^* that has minimal empirical risk and the actual risk as defined in (3.2); moreover we would like to study the asymptotic behaviour of this deviation; this quantity is the *sample error* and will be considered in detail in later sections;

$$R_{\mathcal{S}} = R_{\mathcal{X}}[f_n^*] - R_{\mathcal{X}}[f^*] \tag{3.5}$$

There are two subtleties that must first be considered; to begin with it is clear that the effectiveness of ERM is highly dependent on its associated exploration algorithm which is primarily responsible for searching through the hypothesis space, i.e. iterating through each element of the space $\mathcal{H}$ so that computing the infimum in (3.4) is possible. Minimization of the empirical risk is only half of the ERM learning problem; it must be supplied with the arguments over which it can apply the minimization. It is possible for the learning algorithm, which is a combination of ERM and the exploration algorithm, to find a local minima not far from its starting position and get stuck; potential solutions to this problem will be discussed later.

Secondly, when no data is available the empirical risk (or training loss in this case) is zero and remains as such as long as the prediction function correctly classifies all elements in the training set. As more data becomes available it increases as the prediction function fails to correctly classify an increasing number of training set elements; so the empirical risk is a monotonically increasing function of n. Furthermore, it never surpasses the expected risk; in the limit the empirical risk plateaus but to be able to examine the convergence of the empirical risk in more detail, we introduce a probabilistic generalization bound.

Lemma 3.1.1 (Chernoff's Inequality) *For a fixed function $f \in \mathcal{H}$ and a bounded loss function $A \leq \ell(f) \leq B$, the probability of at least an absolute ϵ-difference between expected and empirical risks is bounded from above;*

$$P\left(R_{\mathcal{X}}[f] - \hat{R}_n[f] \geq \epsilon\right) \leq e^{-n\epsilon^2/(B-A)^2} \tag{3.6}$$

and varies only with ϵ and n as well as the loss function bounds A and B.

This is essentially a quantitative expression of the law of large numbers: as n increases the bound $2e^{-2n\epsilon^2}$ is reduced exponentially fast; this implies an

exponential convergence in probability so that the empirical risk is a probabilistically unbiased estimate of the expected risk;

$$\begin{aligned} &\lim_{n\to\infty} \quad \hat{R}_n[f] \xrightarrow{P} R_{\mathcal{X}}[f] \\ &\iff \quad \forall \epsilon > 0 \;\; \exists \delta = e^{-n\epsilon^2/(B-A)^2} \text{ s.t. } P\left(R_{\mathcal{X}}[f] - \hat{R}_n[f] \geq \epsilon\right) \leq \delta \\ &\iff \quad \forall \epsilon > 0 \;\; \exists \delta = e^{-n\epsilon^2/(B-A)^2} \text{ s.t. } P\left(R_{\mathcal{X}}[f] - \hat{R}_n[f] < \epsilon\right) > 1 - \delta \end{aligned} \tag{3.7}$$

The ϵ defines a one-sided confidence interval while the δ is its corresponding confidence level.

This closeness of the empirical risk to the expected risk defines the notion of *generalization*; it gives us an assurance that by minimizing the empirical risk (3.4), we are more likely to select a function that will have a small expected risk as well or in simpler terms; when test (expected) performance and training (empirical) performance are highly correlated which allows the learner to determine the parametrization of an accurate prediction function. Conditions for generalization and a diminishing sample error R_{S} are the focus of this chapter.

The *generalization error* is defined as the difference between the empirical and expected risks; (3.6) is an example of a *generalization bound* that attempts to link the performance of a prediction function on some training set to its potential performance on an unseen test set; since there exists the possibility that the distribution of the training set is highly unrepresentative of the actual distribution $P(\vec{x}, y)$, generalization bounds only hold with a certain probability. Furthermore, the generalization bound is void if the value of ϵ (in 3.6) exceeds the largest possible generalization error. Finally, the *generalization potential* of a learning method lies in its ability [Vap00] to regulate the rate of convergence defined by some generalization bound.

The convergence in (3.7), as well as others we will see in the sections that follow, define what is commonly referred to as a Probably Approximately Correct (PAC) Generalization; suppose we would like to specify with a certain confidence when generalization is likely to occur; then we can select a value for δ which as we see above is a function of both n and ϵ; PAC Generalization then occurs when with *probability* at least $1-\delta$, the empirical risk is ϵ-*approximately* equivalent to the expected risk or in simpler terms; the generalization error is *almost* surely *very* close to zero.

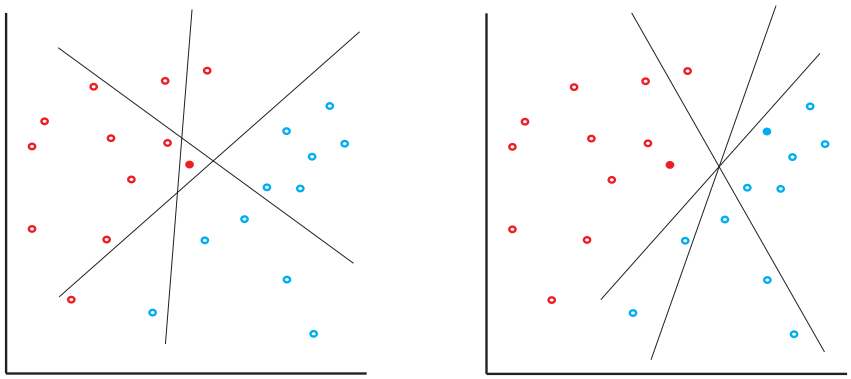

Figure 3–1: If we restrict the hypothesis space by considering only linear hypothesis; the number of admissible classification functions (which in this case implies perfect separation of the blue and red examples) decreases as the size of the training set (solid circles) increases; the generalization potential of functions in this reduced set simultaneously increases as they classify test examples (open circles) more accurately. However even amongst the set of admissible functions there is one unique function (which is possibly equivalent to the target function) that generalizes better than all others.

However, a learning algorithm that returns a prediction function with low empirical risk that is un-generalizable is of no use; conversely a prediction function that is able to generalize satisfying (3.7) but that has a large empirical risk is impossible to identify using the ERM approach.

Prediction functions chosen by ERM alone are often unable to generalize; this is because there can be infinitely many functions that have minimal risk, amongst which a single unique element maintains the highest generalization potential. As an example let us consider the hypothesis space that consists of all possible functions so that any training set can be fitted with an (unnecessarily complex) function whose empirical risk is zero but that has no generalization potential whatsoever; if we do not restrict the capacity of the hypothesis space then learning is simply not possible!

In section 2.4 we consider two regularization methods that exclude those sections of the hypothesis space that we know a priori will not contain the empirical target function; as more training data becomes available, we can make stronger assumptions on the distribution of the data and hence regulate the capacity of the hypothesis space further still.

3.2 Uniformly Convergent Generalization Bounds

It is obvious that for any fixed value of n, the function f^* defined in (3.1) that minimizes the expected risk is not necessarily equivalent to the function $\hat{f}^*$ defined in (3.4) that minimizes the empirical risk. This is due to a significant weakness of the convergence (3.7) in that it is a point-wise limit implying that the rate of convergence may differ amongst the various functions in the function space $\mathcal{H}$ so that even for very large n where we have 'convergence' for some subset of $\mathcal{H}$ there might exist functions that have not yet even begun to approach their limits; we must consider the worst-case in our analysis of the convergence of the empirical risk and hence need to extend Chernoff's Inequality to consider all functions collectively by bounding from above the supremum of the generalization error:

$$\sup_{f \in \mathcal{H}} \left(R_{\mathcal{X}}[f] - \hat{R}_n[f] \right) \leq \epsilon \tag{3.8}$$

This is a stronger generalization criteria than (3.7): intuitively, since we do not know in advance which function is optimal at future stages of the learning process, we must consider the worst case of every function and union these together to form a uniform (pessimistic) bound.

A generalization bound similar to Chernoff's inequality but for all functions in $\mathcal{H}$ may be derived by taking the union over $\mathcal{H}$ and then using the sub-additivity property of probability measures where the probability of the union is bounded from above by the sum of the individual probabilities in (3.6):

$$\begin{aligned} P\left(\exists f \in \mathcal{H} : \left(R_{\mathcal{X}}[f] - \hat{R}_n[f] \geq \epsilon\right)\right) &= P\left(\cup_{f \in \mathcal{H}} \left(R_{\mathcal{X}}[f] - \hat{R}_n[f] \geq \epsilon\right)\right) \\ &\leq \sum_{f \in \mathcal{H}} P\left(R_{\mathcal{X}}[f] - \hat{R}_n[f] \geq \epsilon\right) \\ &\leq \sum_{f \in \mathcal{H}} e^{-n\epsilon^2/(B-A)^2} \\ &= |\mathcal{H}| e^{-n\epsilon^2/(B-A)^2} \end{aligned} \tag{3.9}$$

We can rewrite (3.9) in a form similar to (3.8) where if the supremum of the generalization error is bounded from above by ϵ then all functions in $\mathcal{H}$ must also be bounded by ϵ:

$$
\begin{aligned}
P\left(\sup_{f\in\mathcal{H}}\left(R_{\mathcal{X}}[f]-\hat{R}_n[f]\right)\leq\epsilon\right) &= P\left(\forall f\in\mathcal{H}:\left(R_{\mathcal{X}}[f]-\hat{R}_n[f]\leq\epsilon\right)\right)\\
&= 1-P\left(\exists f\in\mathcal{H}:\left(R_{\mathcal{X}}[f]-\hat{R}_n[f]\geq\epsilon\right)\right)\\
&> 1-|\mathcal{H}|e^{-n\epsilon^2/(B-A)^2}
\end{aligned}
$$

Now let $\delta = |\mathcal{H}|e^{-n\epsilon^2/(B-A)^2}$; then solving for ϵ we have

$$\epsilon=\sqrt{\log\left(\frac{|\mathcal{H}|}{\delta}\right)\frac{(B-A)^2}{n}} \tag{3.10}$$

LEMMA 3.2.1 (HOEFFDING'S INEQUALITY) *A* distribution-free *bound that quantifies the deviation of the empirical mean* $\hat{R}_n[f]$ *from its true value* $R_{\mathcal{X}}[f]$ *over* $\mathcal{H}$

$$\sup_{f\in\mathcal{H}}\left(R_{\mathcal{X}}[f]-\hat{R}_n[f]\right)\leq\sqrt{\log\left(\frac{|\mathcal{H}|}{\delta}\right)\frac{(B-A)^2}{n}} \tag{3.11}$$

and which holds with probability at least $1-\delta$ *for a finite hypothesis space* $|\mathcal{H}|<\infty$.

The convergence is still exponentially fast (3.9) but the generalization bound now depends only on the choice of function class $\mathcal{H}$, the size of the training set and a parameter $\delta : 0\leq\delta\leq 1$; it is said to be distribution-free because it holds independently of $P(\vec{x},y)$, the distribution of data generation. Also the bound holds with probability at least $1-\delta$ for the ERM prediction function $\hat{f}^*$ defined in (3.4); moreover it holds (with the exact same probability $1-\delta$) for *all* other hypothesis in the function space $\mathcal{H}$ and hence is a *uniform convergence bound*. To see this let us first formally define the notion of one-sided uniform convergence:

$$\forall\epsilon>0\ \ \exists N\in\mathbb{N}\text{ such that }\forall n>N\text{ and }\forall f\in\mathcal{H}\quad\left(R_{\mathcal{X}}[f]-\hat{R}_n[f]\right)<\epsilon \tag{3.12}$$

Now for any choice of $\epsilon>0$, we can show that (3.11) satisfies (3.12) by taking a value of $N(\epsilon,\delta)\in\mathbb{N}$ large enough so that Hoeffding's bound is itself bounded by ϵ for all $n>N(\epsilon,\delta)$;

$$\sqrt{\log\left(\frac{|\mathcal{H}|}{\delta}\right)\frac{(B-A)^2}{n}}<\epsilon$$

The value of N (which depends on our choice of ϵ and δ) is called the *sample complexity* of the learning algorithm; more specifically it is a probabilistic estimate of the number of training examples that are necessary and sufficient for an algorithm to learn (generalize) some (unknown) target concept; instead of solving for ϵ in (3.10) we solve for n to get:

$$n \geq \left(\frac{B-A}{\epsilon}\right)^2 \log\left(\frac{|\mathcal{H}|}{\delta}\right) = N$$

so with probability (at least) $1-\delta$ and (at least) $n \geq N$ samples the generalization error is epsilon bounded for all functions.

Now does satisfying Chernoff's inequality imply uniform convergence of the empirical risk to the expected risk over the entire function space $\mathcal{H}$? Comparing the bounds in (3.9) and (3.6) we see that the former is simply a multiple (by the size of the hypothesis space) of the latter and hence both bounds are essentially equivalent. So if every function $f \in \mathcal{H}$ satisfies Chernoff's inequality individually then it must satisfy Hoeffding's inequality collectively and hence (3.12) is satisfied.

It is important to note that there are two ways in which the generalization bound (3.11) can be tightened; by either bounding the capacity of the hypothesis space (whose cardinality can then be roughly measured even if it is uncountably infinite) or by bounding the stability (definition 2.4.1) or sensitivity of the prediction function, output by some learning algorithm, to perturbations in the training set (definition 2.4.2).

The search for an optimal prediction function is conducted in the loss class defined over some hypothesis space and not in the hypothesis space itself, we have so far ignored this technicality; we can extend the notion of uniform convergence over a hypothesis space and characterize uniformly convergent loss classes as follows:

Definition 3.2.1 (Uniform Glivenko-Cantelli Class (uGC)) *is a class of functions* $\mathcal{L}(\mathcal{H}) = \{\ell(f) : f \in \mathcal{H}\}$ *for a fixed bounded loss function* $A < \ell < B$ *such that the functions* $f \in \mathcal{H}$ *are integrable with respect to the probability measure* $P(\vec{x}, y)$ *and the following one-sided uniform convergence is satisfied;*

$$\forall \epsilon > 0 \quad \lim_{n\to\infty} P\left(\sup_{\ell(f)\in\mathcal{L}(\mathcal{H})} \left(R_{\mathcal{X}}[f] - \hat{R}_n[f]\right) > \epsilon\right) = 0$$

In the following subsection we will prove that a necessary and sufficient condition for consistency of ERM is that the loss class $\mathcal{L}(\mathcal{H})$ is uGC.

3.3 Generalization and the Consistency of ERM

In this learning framework, the key quantity that is being estimated is the actual risk; so we say the learning method is *consistent* [Vap00] for function class $\mathcal{H}$ and the distribution $P(\vec{x}, y)$ if the empirical risk (for the prediction function f_n output by the learning algorithm) converges in probability to the actual risk:

$$\lim_{n\to\infty} \hat{R}_n[f_n] \xrightarrow{P} \inf_{f\in\mathcal{H}} R_{\mathcal{X}}[f] \iff$$
$$\forall \epsilon > 0 \;\; \exists \delta \text{ s.t. } P\left(\left|\hat{R}_n[f_n] - \inf_{f\in\mathcal{H}} R_{\mathcal{X}}[f]\right| \geq \epsilon\right) \leq \delta \tag{3.13}$$

There are two essential differences between consistency as defined above and generalization; firstly, consistency is defined by a convergence of the empirical risk of the prediction made by the learning algorithm (i.e. choice of the prediction function is dependent on n) whereas the weaker generalization (3.7) is a point-wise convergence over a fixed prediction (i.e. choice of the prediction function is independent of n) and the stronger generalization (3.11) is a uniform convergence over all predictions; so in this respect consistency is dependent on the learning algorithm (of which the exploration of the hypothesis space is an essential part) although generalization is not. In fact we will later show that uniform convergence (strong generalization) and consistency of the learning algorithm ERM are essentially equivalent.

Secondly, the limit of the consistency convergence is the minimized expected risk. Consistency is therefore stronger than the weaker generalization but weaker than the stronger generalization criteria and requires that the learning algorithm speculate on optimality of functions in the hypothesis space (which involves its exploration) before precisely estimating its expected risk.

The performance of ERM is optimal if the function f_n^* that minimizes the empirical risk is equivalent (in probability) to the function f^* that minimizes the expected risk;

$$\exists N \in \mathbb{N} \text{ such that } \forall n > N \;:\; f_n^* = \arg\inf_{f\in\mathcal{H}} \hat{R}_n[f_n] \overset{P}{=} \arg\inf_{f\in\mathcal{H}} R_{\mathcal{X}}[f] = f^*$$

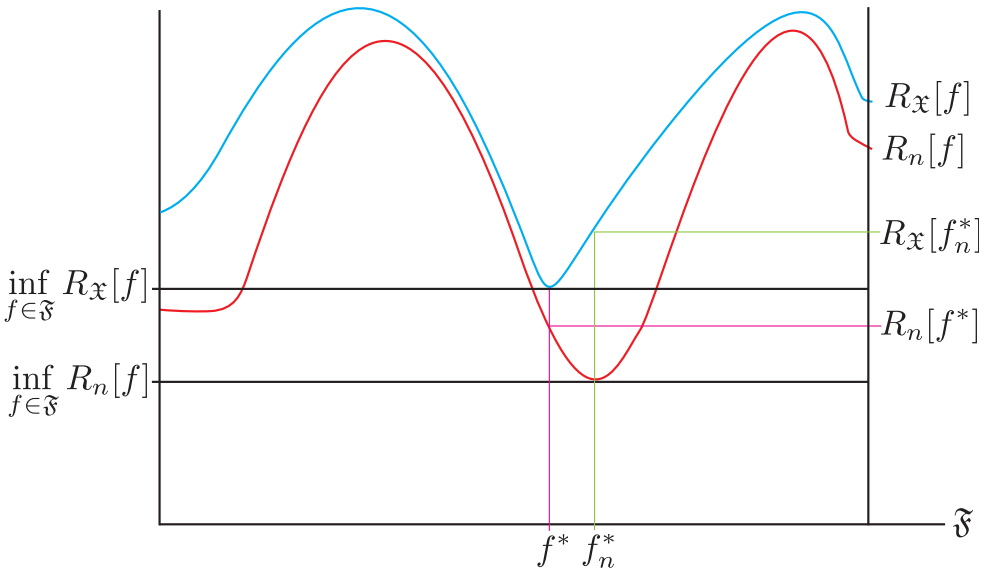

Figure 3–2: Uniform convergence of the empirical risk (red) to the expected risk (blue) implies a consistent learning method.

So for ERM in particular whose choice in prediction function satisfies (3.4), consistency is also implied [Gun98] by:

$$\lim_{n\to\infty} \arg\inf_{f\in\mathcal{H}} \hat{R}_n[f_n] \overset{P}{=} \arg\inf_{f\in\mathcal{H}} R_{\mathfrak{X}}[f] = f^* \tag{3.14}$$

So is the consistency of the ERM learning algorithm implied by the uniform convergence (3.12) in probability of the generalization error to zero and vice versa, i.e. is a uGC loss class sufficient for consistency of ERM? Yes it is, in fact the reasoning that led us to move from a point-wise to a uniform convergence at the beginning of Section 3.2 was precisely so that the consistency criteria (3.13) would be satisfied.

To see this more formally; let us assume that we have uniform convergence (strong generalization) so that the supremum of the generalization error is bounded by some $\epsilon > 0$; looking at Figure 3–2 we see that the empirical risk evaluated for any prediction function must then lie wholly within an ϵ-tube defined around the expected risk; in particular the empirical risk of the function f^* that minimizes the expected risk (and the function f_n^* that minimizes the empirical risk) must lie within this ϵ-tube which leads to the implications

(3.15) and (3.16).

$$\sup_{f\in\mathcal{H}}\Big(R_{\mathcal{X}}[f]-\hat{R}_n[f]\Big)\leq\epsilon\Longrightarrow\inf_{f\in\mathcal{H}}R_{\mathcal{X}}[f]-\hat{R}_n\left[\arg\inf_{f\in\mathcal{H}}R_{\mathcal{X}}[f]\right]=R_{\mathcal{X}}[f^*]-\hat{R}_n[f^*]<\epsilon \tag{3.15}$$

$$\sup_{f\in\mathcal{H}}\Big(R_{\mathcal{X}}[f]-\hat{R}_n[f]\Big)\leq\epsilon\Longrightarrow R_{\mathcal{X}}\left[\arg\inf_{f\in\mathcal{H}}\hat{R}_n[f]\right]-\inf_{f\in\mathcal{H}}\hat{R}_n[f]=R_{\mathcal{X}}[f_n^*]-\hat{R}_n[f_n^*]<\epsilon \tag{3.16}$$

Also, since f^* minimizes the expected risk (3.1) and f_n^* minimizes the empirical risk (3.4) the following are trivially satisfied:

$$\begin{aligned}\hat{R}_n[f_n^*] &\leq \hat{R}_n[f^*] \\ R_{\mathcal{X}}[f^*] &\leq R_{\mathcal{X}}[f_n^*]\end{aligned} \tag{3.17}$$

Combining inequalities (3.16) and (3.17) together we have the following:

$$R_{\mathcal{X}}[f^*]\leq R_{\mathcal{X}}[f_n^*]\leq\hat{R}_n[f_n^*]+\epsilon\leq\hat{R}_n[f^*]+\epsilon\leq R_{\mathcal{X}}[f^*]+2\epsilon \tag{3.18}$$

So we have shown that $\hat{R}_n[f_n^*]$ and $R_{\mathcal{X}}[f^*]$ are ϵ-equivalent in the limit which is in fact the definition of consistency for which uniform convergence is therefore a sufficient criteria;

$$\sup_{f\in\mathcal{H}}\Big(R_{\mathcal{X}}[f]-\hat{R}_n[f]\Big)<\epsilon\Longrightarrow\inf_{f\in\mathcal{H}}R_{\mathcal{X}}[f]-\hat{R}_n\left[\arg\inf_{f\in\mathcal{H}}\hat{R}_n[f]\right]=R_{\mathcal{X}}[f^*]-\hat{R}_n[f_n^*]<\epsilon \tag{3.19}$$

The *sample error* $R_{\mathcal{S}}$ was defined in (3.5); intuitively it gauges the true error of the optimal prediction made by the empirical process. Generalization dictates that $\hat{R}_n[f_n^*]$ tends to $R_{\mathcal{X}}[f_n^*]$ while a small sample error implies that $R_{\mathcal{X}}[f_n^*]$ tends to $R_{\mathcal{X}}[f^*]$; so consistency demands generalization of the empirical process *and* a sample error $R_{\mathcal{S}}$ that diminishes to zero. Combining the inequalities (3.15), (3.16) and (3.17) together we have the following:

$$R_{\mathcal{X}}[f_n^*]\leq\hat{R}_n[f_n^*]+\epsilon\leq\hat{R}_n[f^*]+\epsilon\leq R_{\mathcal{X}}[f^*]+2\epsilon \tag{3.20}$$

The first and last terms in the above sequence of inequalities include those in the definition of the sample error as well as 2ϵ which is arbitrarily small to begin with, so the exponentially fast rate of uniform convergence is approximately half the rate at which the sample error is guaranteed to diminish to zero.

So we have shown that the learning process depends on the distribution $P(\vec{x},y)$ but more significantly on the function space $\mathcal{H}$; this is because the

uniform convergence of the empirical risk to the expected risk and hence the consistency of the learning method is dependent on it. We would now like to study properties of loss classes (and their associated function spaces) that guarantee that it is uGC and hence that learning is in fact possible.

3.4 Vapnik-Chervonenkis Theory

One serious limitation of Hoeffding's bound (3.11) is that it was necessary to assume that the function space is finite $|\mathcal{H}| < \infty$ since we use the finite sub-additivity of probability measures to derive it in (3.9); it is possible to extend Hoeffding's bound for countably infinite hypothesis spaces [BBL03] however we would like to examine learning in an infinite *uncountable* function space for which the union bound does not hold.

Since we cannot use the cardinality of the hypothesis space in deriving a generalization bound since it is possibly infinite, we need to find a new measure that relates to the notion of generalization; specifically we need to know why, for a given learning task, functions from one infinite space are able to generalize whereas those from another infinite space are not. In previous sections we saw that for ERM, generalization and consistency were equivalent which in turn was necessitated and guaranteed by the uniform convergence of the empirical risk to the expected risk; this much has not changed.

The cardinality of a function space is a count of the number of functions in it and is essentially a measure of its complexity; since we are dealing with infinite hypothesis spaces we will now consider various other measures through which we can gauge the complexity of a hypothesis space and then relate it to the uniform convergence of the generalization error to zero in order to determine if learning is possible. We begin by defining a measure that is essentially an ϵ-count of the number of functions in a function space in terms of the supremum norm:

Definition 3.4.1 (Epsilon Net) *Given a function space $\mathcal{H}$ and some $\epsilon > 0$, we say that a subset $\mathfrak{U} \subset \mathcal{H}$ is an ϵ-net (or ϵ-cover) for $\mathcal{H}$ if*

$$\forall f \in \mathcal{H} \;\; \exists \breve{f} \in \mathfrak{U} \text{ such that } \|f, \breve{f}\|_\infty < \epsilon$$

Members of the set $\mathfrak{U}$ are referred to as prototype functions. If for all $\epsilon > 0$, $\mathcal{H}$ has a finite ϵ-net then it is totally bounded (or precompact) which along

with Cauchy completeness implies compactness. The converse also holds true so that a space which is compact must also be Cauchy complete and totally bounded (generalization of the Heine-Borel Theorem) and hence have a finite ϵ-net. Finally, a space that is bounded must also be totally bounded although the converse is not necessarily implied. We can also define the ϵ-net $\mathbb{C}_t$ of a single function $\breve{f}_t \in \mathcal{H}$ as the set of functions that are within its 'reach' as measured by the supremum norm:

$$\forall f \in \mathbb{C}_t \quad \|f - \breve{f}_t\|_\infty \leq \epsilon$$

So the basic intuition behind VC-Theory is that in any function (hypothesis) space, if two functions are ϵ-close then it is reasonable to assume that they will perform similarly on a fixed training set (or any fixed test set) and hence any generalization bound that holds for one function will naturally hold for the other.

Since the measure defined above groups functions together by ensuring that each one is entirely contained in the ϵ-tube of at least one of a fixed set of functions, we will also require a contrasting measure to assess the size of the gap between functions as measured by the infimum norm:

Definition 3.4.2 (Epsilon Separation) *Given a function space $\mathcal{H}$ and some $\epsilon > 0$, we say that a subset of l functions of $\mathcal{H}$ are ϵ-separated if*

$$\{f_i\}_{i=1}^{l} \subset \mathcal{H} \text{ satisfies } \|f_i, f_j\| > \epsilon \ \ \forall i \neq j$$

3.4.1 Compact Hypothesis Spaces $\mathcal{H}$

All hypothesis that produce the same classification (or the same ϵ-close regression) on a given training data set can be grouped together into an equivalence class since, from the perspective of ERM, they are alike in that they have the same empirical risk. The number of such equivalence classes is called the VC-Entropy of $\mathcal{H}$ when the outputs are binary $y \in \{+1, -1\}$ and analogously in the case of regression estimation is called the *covering number* of $\mathcal{H}$. We now define the latter as well as a related measure:

Definition 3.4.3 (Epsilon Covering Number) *Given an infinite cardinality function space $\mathcal{H}$, the covering (or entropy) number $\mathcal{N}(\mathcal{H}, \epsilon)$ is the minimal $c \in \mathbb{N}$ such that*

$$\exists \left\{ \breve{f}_i \right\}_{i=1}^{c} \text{ where } \forall f \in \mathcal{H} \ \exists t : 1 \leq t \leq c \text{ such that } \|f - \breve{f}_t\|_\infty \leq \epsilon$$

Essentially, it is smallest number of functions in $\mathcal{H}$ that can serve as an ϵ-net for $\mathcal{H}$. Geometrically, $\mathcal{N}(\mathcal{H}, \epsilon)$ is the minimal number of disks in $\mathcal{H}$ with radius ϵ needed to cover $\mathcal{H}$. The empirical *covering number is restricted to the training data set $\mathcal{S}_n = \{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n\}$; denoted by $\hat{\mathcal{N}}(\mathcal{H}, \epsilon, \mathcal{S}_n)$ it is then the minimal $c \in \mathbb{N}$ such that*

$$\exists \left\{ \breve{f}_i \right\}_{i=1}^{c} \text{ where } \forall f \in \mathcal{H} \ \exists t : 1 \leq t \leq c \text{ such that } \max_{j=1,\cdots,n} \left| f(\vec{x}_j) - \breve{f}_t(\vec{x}_j) \right| \leq \epsilon$$

Since the empirical covering number is dependent on the data we must work with its expected value, which is taken with respect to the input distribution and denoted as $\mathbb{E}_{\mathcal{S}_n} \hat{\mathcal{N}}(\mathcal{H}, \epsilon, \mathcal{S}_n)$.

Use of the expected empirical covering number in a generalization bound results in its dependence on the input distribution $P(\vec{x}, y)$; since in practice the true covering number for most compact real spaces of interest is not calculable, finding distribution independent bounds is generally quite difficult or not even possible.

Definition 3.4.4 (Epsilon Packing Number) *Given a function space $\mathcal{H}$, the packing number $\mathcal{D}(\mathcal{H}, \epsilon)$ is the maximal $l \in \mathbb{N}$ such that:*

$$\{f_i\}_{i=1}^{l} \subset \mathcal{H} \text{ satisfies } \|f_i, f_j\|_p > \epsilon \ \ \forall i \neq j$$

Essentially, it is the maximal number of functions in $\mathcal{H}$ that can be ϵ-separated.

The following inequalities upper and lower bound the covering number in terms of the packing number;

$$\mathcal{D}(\mathcal{H}, 2\epsilon) \leq \mathcal{N}(\mathcal{H}, \epsilon) \leq \mathcal{D}(\mathcal{H}, \epsilon) \tag{3.21}$$

therefore we can use the latter in computing an approximation to the former. In [Muk07] we see derivations for such approximations.

In section 2.4 the RKHS $\mathcal{H}_\mathcal{T}$ was bounded which implies it is totally bounded (precompact) and therefore must have a finite, minimal (not necessarily unique) ϵ-net $\mathfrak{U} = \{\breve{f}_1, \breve{f}_2, \cdots \breve{f}_c\} \subset \mathcal{H}_\mathcal{T}$ where c is the covering number $\mathcal{N}(\mathcal{H}_\mathcal{T}, r_\epsilon(\ell))$; the radius $r_\epsilon(\ell)$ of the covering is dependent on both the loss function ℓ and the value of ϵ we use to bound the supremum of the generalization error. Let us denote the ϵ-net of the prototype function $\breve{f}_t$ by $\mathbb{C}_t$ which satisfies the following:

$$\bigcup_{t=1}^{c} \mathbb{C}_t = \mathcal{H}_\mathcal{T} \tag{3.22}$$

Let us now consider two distinct functions; a prototype function $\breve{f}_t$ of the space $\mathcal{H}$ and any other function $f \in \mathbb{C}_t$. Our goal is to bound the difference between the generalization errors of $\breve{f}_t$ and f; this will lead us to a new generalization bound involving the covering number instead of the cardinality of the hypothesis space.

$$\begin{aligned} |R_\mathcal{X}(f) - \hat{R}_n(f) - R_\mathcal{X}(\breve{f}_t) + \hat{R}_n(\breve{f}_t)| &\leq |R_\mathcal{X}(f) - R_\mathcal{X}(\breve{f}_t)| + |\hat{R}_n(f) - \hat{R}_n(\breve{f}_t)| \\ &= \left| \int \ell(f, \{\vec{x}, y\}) - \ell(\breve{f}_t, \{\vec{x}, y\})\, dP(\vec{x}, y) \right| \\ &+ \left| \frac{1}{n} \sum \ell(f, \{\vec{x}, y\}) - \ell(\breve{f}_t, \{\vec{x}, y\}) \right| \end{aligned} \tag{3.23}$$

Definition 3.4.5 (Lipschitz Loss Functions) *are a class of functions that satisfy the following inequality*

$$\|\ell(f_1, \cdot) - \ell(f_2, \cdot)\|_\infty \leq L\|f_1 - f_2\|_\infty$$

for a given Lipschitz constant L. Examples of Lipschitz loss functions include the ϵ-insensitive function, the square loss function (only when the annotations can be bounded) and the hinge loss function.

So for any Lipschitz loss function, the integral in (3.23) can then be bounded:

$$\begin{aligned} \int \ell(f, \{\vec{x}, y\}) - \ell(\breve{f}_t, \{\vec{x}, y\})\, dP(\vec{x}, y) &\leq \int \left\| \ell(f, \{\vec{x}, y\}) - \ell(\breve{f}_t, \{\vec{x}, y\}) \right\|_\infty dP(\vec{x}, y) \\ &= \left\| \ell(f, \{\vec{x}, y\}) - \ell(\breve{f}_t, \{\vec{x}, y\}) \right\|_\infty \int dP(\vec{x}, y) \\ &= \left\| \ell(f, \{\vec{x}, y\}) - \ell(\breve{f}_t, \{\vec{x}, y\}) \right\|_\infty \\ &\leq L\|f - \breve{f}_t\|_\infty \end{aligned} \tag{3.24}$$

Similarly for the sum in (3.23)

$$\begin{aligned}\frac{1}{n}\sum \ell(f,\{\vec{x},y\}) - \ell(\breve{f}_t,\{\vec{x},y\}) &\leq \frac{1}{n}\sum \left\|\ell(f,\{\vec{x},y\}) - \ell(\breve{f}_t,\{\vec{x},y\})\right\|_\infty \\ &= \left\|\ell(f,\{\vec{x},y\}) - \ell(\breve{f}_t,\{\vec{x},y\})\right\|_\infty \\ &\leq L\|f-\breve{f}_t\|_\infty\end{aligned}$$

From which it then follows that:

$$\begin{aligned}|R_{\mathcal{X}}(f) - \hat{R}_n(f) - R_{\mathcal{X}}(\breve{f}_t) + \hat{R}_n(\breve{f}_t)| &\leq L\left|\|f-\breve{f}_t\|_\infty\right| + L\left|\|f-\breve{f}_t\|_\infty\right| \\ &\leq 2L\left|\|f-\breve{f}_t\|_\infty\right| \qquad (3.25)\end{aligned}$$

If we consider a square loss function $\ell(f,\{\vec{x},y\}) = (f(\vec{x})-y)^2$ that is obviously positive but also bounded from above $\ell(f,\cdot) \leq B$ then we can derive ([PMRR04],[Muk07]) a value for the Lipschitz constant; following from (3.23) we have:

$$\begin{aligned}&\left|\int (f(\vec{x})-y)^2 - (\breve{f}_t(\vec{x})-y)^2\, dP(\vec{x},y)\right| + \left|\frac{1}{n}\sum (f(\vec{x})-y)^2 - (\breve{f}_t(\vec{x})-y)^2\right| \\ =\ &\left|\int \left(f(\vec{x})-\breve{f}_t(\vec{x})\right)\left(f(\vec{x})+\breve{f}_t(\vec{x})-2y\right)\, dP(\vec{x},y)\right| \\ &+\left|\frac{1}{n}\sum \left(f(\vec{x})-\breve{f}_t(\vec{x})\right)\left(f(\vec{x})+\breve{f}_t(\vec{x})-2y\right)\right| \\ \leq\ &\|f-\breve{f}_t\|_\infty \int \left|\left(f(\vec{x})-y+\breve{f}_t(\vec{x})-y\right)\right|\, dP(\vec{x},y) \\ &+\|f-\breve{f}_t\|_\infty \frac{1}{n}\left|\sum \left(f(\vec{x})-y+\breve{f}_t(\vec{x})-y\right)\right| \\ \leq\ &\|f-\breve{f}_t\|_\infty \int \left|\ell(f,\{\vec{x},y\}) + \ell(\breve{f}_t,\{\vec{x},y\})\right|\, dP(\vec{x},y) \\ &+\|f-\breve{f}_t\|_\infty \frac{1}{n}\left|\sum \ell(f,\{\vec{x},y\}) + \ell(\breve{f}_t,\{\vec{x},y\})\right| \\ \leq\ &2B\|f-\breve{f}_t\|_\infty + 2B\|f-\breve{f}_t\|_\infty \\ \leq\ &4B\|f-\breve{f}_t\|_\infty\end{aligned}$$

Let us return to the general case of Lipschitz loss functions and arbitrarily set the radius of the covering to be a function of ϵ and the Lipschitz constant:

$$\forall f \in \mathbb{C}_t \ \ \|f-\breve{f}_t\|_\infty \leq r_\epsilon(\ell) = \epsilon/4L$$

from which it follows that the difference between the generalization errors of $\breve{f}_t$ and f is bounded by $\epsilon/2$:

$$\begin{aligned} \sup_{f\in\mathbb{C}_t} |R_{\mathcal{X}}(f) - \hat{R}_n(f) - R_{\mathcal{X}}(\breve{f}_t) + \hat{R}_n(\breve{f}_t)| &\leq 2L \, \|f - \breve{f}\|_\infty \\ &\leq 2L \, r_\epsilon(\ell) \\ &= \epsilon/2 \end{aligned} \tag{3.26}$$

So if the largest generalization error of functions in $\mathbb{C}_t$ is at least ϵ then the generalization error of the prototype function $\breve{f}_t$ must be at least $\epsilon/2$;

$$\begin{aligned} \sup_{f\in\mathbb{C}_t} |R_{\mathfrak{X}}(f) - \hat{R}_n(f)| \geq \epsilon \quad &\Longrightarrow \quad |(\geq \epsilon) - R_{\mathcal{X}}(\breve{f}_t) + \hat{R}_n(\breve{f}_t)| \leq \epsilon/2 \\ &\Longrightarrow \quad |R_{\mathfrak{X}}(\breve{f}_t) - \hat{R}_n(\breve{f})| \geq \epsilon/2 \end{aligned} \tag{3.27}$$

When one event implies another as above, then the formers probability of incidence is always less than or equal to the latter's:

$$P(\sup_{f\in\mathbb{C}_t} |R_{\mathfrak{X}}(f) - \hat{R}_n(f)| \geq \epsilon) \leq P\left(|R_{\mathfrak{X}}(\breve{f}) - \hat{R}_n(\breve{f})| \geq \epsilon/2\right) \tag{3.28}$$

We can apply Chernoff's Inequality (Definition 3.1.1) to the fixed prototype function $\breve{f}_t$:

$$P\left(|R_{\mathfrak{X}}(\breve{f}_t) - \hat{R}_n(\breve{f}_t)| \geq \epsilon/2\right) \leq 2\exp\left\{-n\frac{(\epsilon/2)^2}{(B-A)^2}\right\} \tag{3.29}$$

which holds for all prototype functions of which there are a finite number; hence we can apply the union bound since (3.22) holds and then use (3.28) and (3.29) to get the following PAC bound that converges exponentially fast:

$$\begin{aligned} P\left(\sup_{f\in\mathcal{H}_{\mathcal{T}}} |R_{\mathfrak{X}}(f) - \hat{R}_n(f)| \geq \epsilon\right) &\leq \sum_{t=1}^{|\mathfrak{U}|} P\left(\sup_{f\in\mathbb{C}_t} |R_{\mathfrak{X}}(f) - \hat{R}_n(f)| \geq \epsilon\right) \\ &\leq \sum_{t=1}^{|\mathfrak{U}|} P\left(|R_{\mathfrak{X}}(\breve{f}_t) - \hat{R}_n(\breve{f}_t)| \geq \epsilon/2\right) \\ &\leq 2\,\mathcal{N}(\mathcal{H}_{\mathcal{T}}, r_\epsilon(\ell)) \, \exp\left\{-n\frac{(\epsilon/2)^2}{(B-A)^2}\right\} \\ &\approx 2\,\mathbb{E}_{\mathcal{S}_n}\hat{\mathcal{N}}(\mathcal{H}, r_\epsilon(\ell), \mathcal{S}_n) \, \exp\left\{-n\frac{(\epsilon/2)^2}{(B-A)^2}\right\} \end{aligned} \tag{3.30}$$

Note that the supremum is now taken over the space $\mathcal{H}_{\mathcal{T}}$ instead of over the cover $\mathbb{C}_t$. Finally applying the logic of (3.9) we have our result:

$$P\left(\sup_{f\in\mathcal{H}_{\mathcal{T}}}\left|R_{\mathcal{X}}[f]-\hat{R}_n[f]\right|\leq\epsilon\right)\geq 1-2\,\mathbb{E}_{\mathcal{S}_n}\hat{\mathcal{N}}(\mathcal{H},r_\epsilon(\ell),\mathcal{S}_n)\;\exp\left\{-n\frac{(\epsilon/2)^2}{(B-A)^2}\right\} \tag{3.31}$$

Upon careful inspection we see that it is almost identical to Hoeffding's Bound (3.10) with the exception of the substitution of $|\mathcal{H}|$ for the expected empirical covering number $\mathbb{E}_{\mathcal{S}_n}\hat{\mathcal{N}}(\mathcal{H},r_\epsilon(\ell),\mathcal{S}_n)$. Let $\delta=2\mathbb{E}_{\mathcal{S}_n}\hat{\mathcal{N}}(\mathcal{H},r_\epsilon(\ell),\mathcal{S}_n)\;e^{-n(\epsilon/2)^2/(B-A)^2}$, then solving for ϵ we have:

$$\sup_{f\in\mathcal{H}_{\mathcal{T}}}\left|R_{\mathcal{X}}[f]-\hat{R}_n[f]\right|\;\leq\;2(B-A)\sqrt{\frac{\log 2\mathbb{E}_{\mathcal{S}_n}\hat{\mathcal{N}}(\mathcal{H},r_\epsilon(\ell),\mathcal{S}_n)+\log(1/\delta)}{n}} \tag{3.32}$$

Examining the above inequality we can rewrite the sufficiency condition for uniform convergence in terms of the covering number alone since all other terms diminish to zero;

$$\lim_{n\to\infty}\frac{\log\mathbb{E}\hat{\mathcal{N}}(\mathcal{H}_{\mathcal{T}},r_\epsilon(\ell))}{n}=0,\quad\forall\epsilon \tag{3.33}$$

in [Vap00] this is referred to as the 'second milestone' in learning theory because it is sufficient and necessary for consistency (as well exponentially fast uniform convergence [SS01]); note that (3.33) is satisfied as long as the capacity of the hypothesis space, as measured by the empirical covering number, increases at most polynomially in n; if it were to increase exponentially in n then the limit above does not converge to zero. So given a compact hypothesis space (which always has a finite cover) as well as a Lipschitz loss function, uniform convergence and therefore consistency are then implied; so compactness and Lipschitz loss are sufficient criteria for uGC classes.

Unfortunately, the notion of covering numbers and ϵ-nets does not translate well to binary classification; and so the generalization bound (3.32) above cannot be applied either. This is because binary thresholding is scale insensitive , i.e. the zero-one loss function does not satisfy the Lipschitz criteria; two classification functions that are only slightly different can have a difference in loss of one.

3.4.2 Indicator Function Hypothesis Spaces $\mathcal{B}$

In [SS01] we see distribution *dependent* generalization bounds that are derived in terms of another measure of the complexity of a function class known as the *VC-Entropy* (or the related measure VC-Annealed-Entropy); distribution *independent* bounds are also derived in terms of the *growth function*. We begin by defining these measures for binary classification:

Definition 3.4.6 (VC-Entropy) *is the finite number of permutations of annotations assigned, by all hypothesis in the potentially infinite (fixed) space $\mathcal{B}$, to an entire (fixed) observation vector set $\mathcal{S}_n$; it varies with the space $\mathcal{B}$ as well as the set $\mathcal{S}_n$ and so is denoted by $\mathcal{N}(\mathcal{B}, \mathcal{S}_n)$; since there are only two possible annotations, it can attain a maximum value of 2^n. Each equivalence class (whose members impose the same classification on the training set) will be denoted by $\mathbb{C}_t$ and satisfies (3.22) as before; furthermore we will select a single representative $\breve{f}_t$ from each class; this prototype function can be any member of $\mathbb{C}_t$; the set of prototype functions is denoted by $\mathfrak{U}$. Since the VC-Entropy depends on the training data we must integrate it with respect to the input distribution over all observation sets of size n so that it can be used in a generalization bound that is applicable to any given data set; hence we define the Annealed VC-Entropy which is simply the logarithm of the expected value of the VC-Entropy and is denoted by $\log \mathbb{E}_{\mathcal{S}_n} \mathcal{N}(\mathcal{B}, \mathcal{S}_n)$.*

Definition 3.4.7 (Symmetrization) *Given a second independent 'ghost' sample set $\breve{\mathcal{S}}$ also of size n, the generalization error can be bounded as follows:*

$$P\left(\sup_{f \in \mathcal{B}} R_{\mathcal{X}}(f) - \hat{R}_n(f, \mathcal{S}) \geq \epsilon\right) \leq 2P\left(\sup_{f \in \mathcal{B}} \hat{R}_n(f, \breve{\mathcal{S}}) - \hat{R}_n(f, \mathcal{S}) \geq \epsilon/2\right) \tag{3.34}$$

Intuitively, if the difference in empirical risk between two independent samples tends (uniformly) to zero then they should both tend (uniformly) to the expected risk as well. For a proof refer to [BBL03].

Now we can derive a generalization bound in terms of the Annealed VC-Entropy; let us denote the VC-Entropy of the set $\mathcal{S} \cup \breve{\mathcal{S}}$ by $k = \mathcal{N}(\mathcal{B}, \mathcal{S} \cup \breve{\mathcal{S}})$ then the supremum of the loss between the training $\mathcal{S}$ and ghost $\breve{\mathcal{S}}$ samples over the space $\mathcal{B}$ is equivalent to the supremum of the same loss over each representative $\breve{f}_t$ from each equivalence class $\mathbb{C}_t$ which collectively form the set $\mathfrak{U} = \{\breve{f}_1, \breve{f}_2, \cdots \breve{f}_k\}$ of size k; we can then apply the union bound since

there are a finite number of equivalence classes:

$$
\begin{aligned}
2P & \left(\sup_{f\in\mathcal{B}}\left(\hat{R}_n(f,\breve{\mathcal{S}})-\hat{R}_n(f,\mathcal{S})\right)\geq\epsilon/2\right) \\
= & \quad 2P\left(\sup_{\breve{f}_t\in\mathfrak{U}}\left(\hat{R}_n(f,\breve{\mathcal{S}})-\hat{R}_n(f,\mathcal{S})\right)\geq\epsilon/2\right) \\
\leq & \quad 2\sum_{i=1}^{k}P\left(\hat{R}_n(\breve{f}_i,\breve{\mathcal{S}})-\hat{R}_n(\breve{f}_i,\mathcal{S})\geq\epsilon/2\right) \qquad (3.35) \\
= & \quad 2\mathcal{N}(\mathcal{B},\mathcal{S}\cup\breve{\mathcal{S}})\ P\left(\hat{R}_n(\breve{f}_i,\breve{\mathcal{S}})-\hat{R}_n(\breve{f}_i,\mathcal{S})\geq\epsilon/2\right) \\
\leq & \quad 2\mathcal{N}(\mathcal{B},\mathcal{S}\cup\breve{\mathcal{S}})\ \exp\{-(\epsilon/2)^2n\} \qquad (3.36) \\
\leq & \quad 2\mathbb{E}_{\mathcal{S}_n}\mathcal{N}(\mathcal{B},\mathcal{S}_n)\ \exp\{-(\epsilon/2)^2n\} \qquad (3.37)
\end{aligned}
$$

where (3.36) follows from an application of Chernoff's Inequality while (3.37) results from taking the expected value over the training set; if (3.36) holds for all possible training sets then it must naturally hold for the expected value as well. Note that although we have derived the above generalization bound using a ghost sample set, in practice this set need not be generated and is used only when theoretically applying the symmetrization principle.

Combining (3.34) and (3.35) we have the following distribution dependent, exponentially fast PAC generalization bound:

$$
P\left(\sup_{f\in\mathcal{B}}R_{\mathcal{X}}(f)-\hat{R}_n(f)\geq\epsilon\right)\leq 2\exp\{\log\mathbb{E}_{\mathcal{S}_n}\mathcal{N}(\mathcal{B},\mathcal{S}_n)-\epsilon^2n\} \qquad (3.38)
$$

A condition similar to (3.33) can be procured from the above generalization bound; it too serves as a criteria for testing if learning is in fact possible when the zero-one loss function is employed:

$$
\lim_{n\to\infty}\frac{\log\mathbb{E}_{\mathcal{S}_n}\mathcal{N}(\mathcal{B},\mathcal{S}_n)}{n}=0,\quad\forall\epsilon \qquad (3.39)
$$

Definition 3.4.8 (Growth Function) *or shattering coefficient is defined as the maximal (worst-case) VC-Entropy over all observation vector sets of size n:*

$$
\Pi_{\mathcal{B}}(n)=\sup\{\mathcal{N}(\mathcal{B},\mathcal{S}_n)\mid\forall\mathcal{S}_n\in\mathcal{X}\}
$$

Note that $\Pi_{\mathcal{B}}(n)$ depends only on the class of functions $\mathcal{B}$ under consideration as well as the size of the training data set n; therefore only one set of patterns in $\mathcal{X}$ might attain the maximal value $\Pi_{\mathcal{B}}(n)$.

The growth function serves as an upper bound for both the VC-Entropy and the Annealed VC-Entropy:

$$\mathcal{N}(\mathcal{B}, \mathcal{S}_n) \leq \log \mathbb{E}_{\mathcal{S}_n} \mathcal{N}(\mathcal{B}, \mathcal{S}_n) \leq \Pi_{\mathcal{B}}(n) \leq \Pi_{\mathcal{B}}(n) \left(1 + \log \frac{n}{\Pi_{\mathcal{B}}(n)}\right)$$

To derive a generalization bound in terms of the growth function, we can make use of the above inequalities and replace the Annealed VC-Entropy in (3.38) with the growth function which gives us the following distribution independent, exponentially fast PAC bound:

$$P\left(\sup_{f \in \mathcal{B}} R_{\mathcal{X}}(f) - \hat{R}_n(f) \geq \epsilon\right) \leq \exp\{\log \Pi_{\mathcal{B}}(n) - \epsilon^2 n\} \tag{3.40}$$

Let $\delta = \exp\{\log \Pi_{\mathcal{B}}(n) - \epsilon^2 n\}$ then after solving for ϵ we have the most significant PAC generalization bound:

THEOREM 3.4.1 (VAPNIK AND CHERVONENKIS) *For all hypothesis $f \in \mathcal{B}$ and some $\delta : 0 \leq \delta \leq 1$ the following generalization bound, given in terms of the growth function of $\mathcal{B}$, holds with probability $1 - \delta$ independent of the input distribution;*

$$\sup_{f \in \mathcal{B}} \left(R_{\mathcal{X}}(f) - \hat{R}_n(f)\right) \leq \sqrt{\frac{\log \Pi_{\mathcal{B}}(n) + \log(1/\delta)}{n}} \tag{3.41}$$

A limit can be procured from the above generalization bound which serves as a criteria for testing if learning is possible when the zero-one loss function is employed;

$$\lim_{n \to \infty} \frac{\log \Pi_{\mathcal{B}}(n)}{n} = 0, \quad \forall \epsilon \tag{3.42}$$

in [Vap00] this is referred to as the 'third milestone' in learning theory because it is sufficient and necessary for consistency and exponentially fast uniform convergence for *all underlying input distributions*; it is therefore more general than either (3.33) or (3.39).

We can now try to illustrate why restricting the capacity of the hypothesis space (as was the case with Ivanov and Tikhinov Regularization, Section 2.4) is absolutely necessary for learning to occur; if for some training set of size n, the functions in $\mathcal{B}$ can shatter it so that $\Pi_{\mathcal{B}}(n) = 2^n$ then (3.42) does not converge to zero which implies there exists input distribution(s) for which the generalization error does not converge uniformly to zero. So we see that it is important that choice of the hypothesis space must be made with reference to

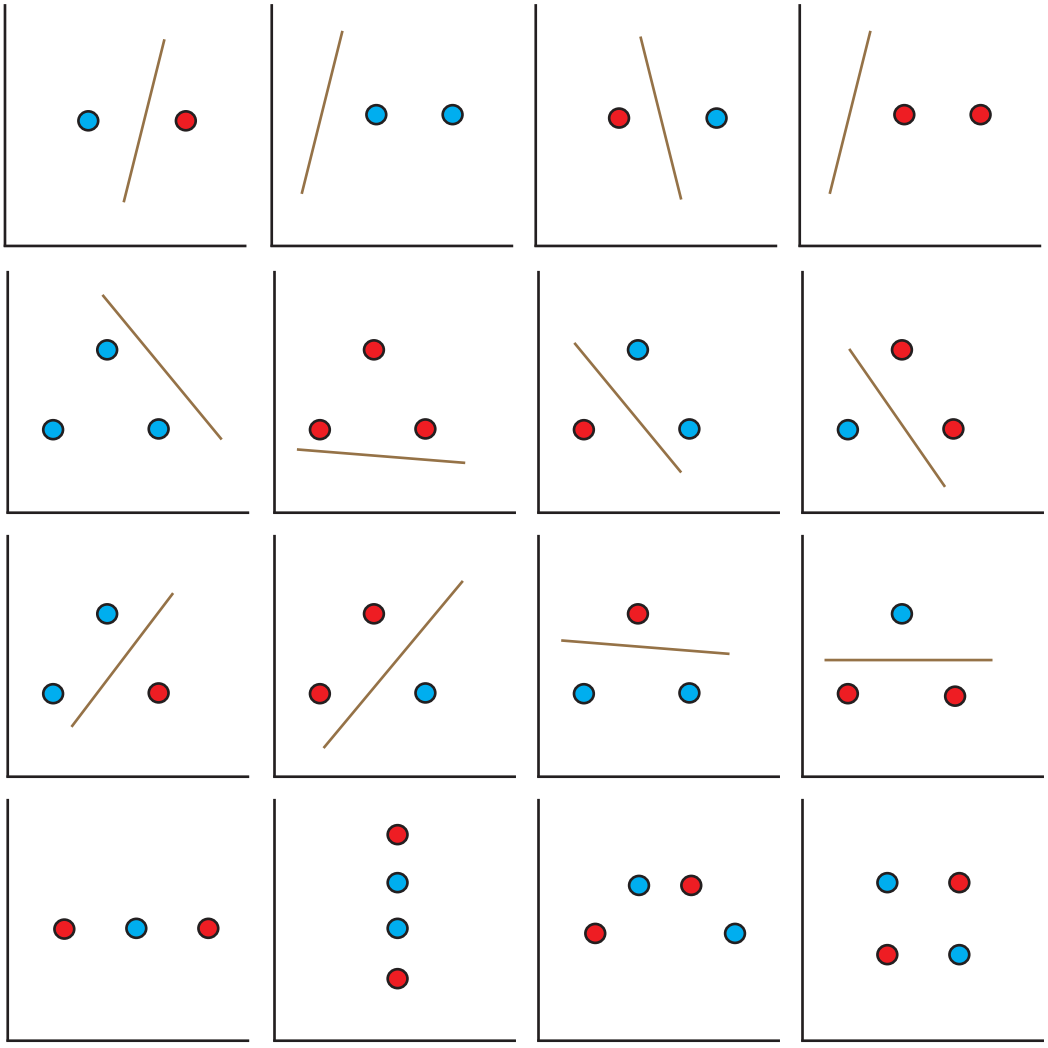

Figure 3–3: Consider the hypothesis space comprising of all discriminant hyperplanes in the feature space $\mathbb{R}^2$; [row 1] any configuration and labeling of 2 points can be separated by a hyperplane and hence $\mathcal{V} \geq \log 2$. [rows 2 & 3] there exists a non-collinear configuration of 3 points that can be shattered and hence $\mathcal{V} \geq \log 3$. The VC-Entropy of a collinear configuration [row 4, left] of 3 points is less than that of the previous configuration; the former cannot be shattered by a hyperplane. Finally, no configuration of 4 points can be shattered by a hyperplane and hence $\mathcal{V} = \log 3$. More generally, the VC-Dimension of half-spaces in $\mathbb{R}^d$ is $d + 1$.

the current size of the training set; in particular the hypothesis space is too rich if it can shatter the training set.

DEFINITION 3.4.9 (VC-DIMENSION) *Intuitively, it is the maximum number of observation vectors for which the hypothesis space $\mathcal{B}$ is unbiased; a rough measure of the capacity of $\mathcal{B}$. Technically, it is the logarithm of the maximum number of observation vectors that can be shattered or separated into two classes in all possible ways by functions in a particular hypothesis space $\mathcal{B}$:*

$$\mathcal{V}(\mathcal{B}) = \log \sup\{n : \Pi_{\mathcal{B}}(n) = 2^n\}$$

The VC-Dimension has value $\log n$ *if there exists even a single (maximal) set of n patterns in $\mathcal{X}$ that can be shattered. The VC-Dimension is infinite if for any n it is possible to shatter n observation vectors with functions taken from $\mathcal{B}$.*

It is also possible to define VC-Dimensions for hypothesis spaces of real-valued functions, see [EP99] for details. Let us assume that the VC-Dimension for a particular class $\mathcal{B}$ is finite; if the VC-Dimension is greater than the size of the training set then it can obviously be shattered by functions in the hypothesis space so that the growth function has value $\log 2^n$. Sauer's Lemma provides a bound for the growth function when n exceeds the VC-Dimension:

$$\Pi_{\mathcal{B}}(n) = \begin{cases} = \log 2^n & \text{when } n \leq \mathcal{V} \\ \leq \sum_{i=0}^{\mathcal{V}} \binom{n}{i} \leq \left(\frac{en}{\mathcal{V}}\right)^{\mathcal{V}} & \text{when } n > \mathcal{V} \end{cases}$$

We have already seen that when the growth function attains its maximum value ($\log 2^n$) then learning is not always possible; it is now interesting to note that this is always the case when the VC-Dimension is greater or equal to the number of training examples available; intuitively we must have enough training examples to represent all sections of the space shattered by a hypothesis. Hence the algorithm is unable to learn properly until it has more than $\mathcal{V}$ training examples for which reason we ignore the first case of the above bound.

Using the above bound for the case when $n > \mathcal{V}$ along with (3.41) we can now bound the generalization error in terms of the VC-Dimension; following from (3.42) we have a PAC (VC-Confidence Interval) bound that holds with

probability (VC-Confidence Level) $1-\delta$:

$$\sup_{f\in\mathcal{B}}\left(R_{\mathcal{X}}(f)-\hat{R}_n(f)\right)\leq\sqrt{\frac{\mathcal{V}\log\frac{en}{\mathcal{V}}+\log\frac{1}{\delta}}{n}} \tag{3.43}$$

In contrast to (3.33), (3.39) and (3.42), the following constructive (can actually be computed) criteria for learning can be derived from the above generalization bound:

$$\lim_{n\to\infty}\frac{\mathcal{V}\log\frac{en}{\mathcal{V}}}{n}=\lim_{n\to\infty}\frac{\mathcal{V}\left(1+\log\frac{n}{\mathcal{V}}\right)}{n}=0,\quad\forall\epsilon \tag{3.44}$$

Necessary and sufficient conditions [DGZ91] for the consistency of the ERM method and the fast (uniform) convergence of the generalization error to zero (the loss class is uGC) over all underlying input distributions can now be succinctly given as a single criteria; the finiteness of the VC-Dimension. Moreover, the number of training examples required (sample complexity), to approximate (learn) the target concept well, must exceed the VC-Dimension $\mathcal{V}$ since this forces the term $n^{-1}\log\frac{n}{\mathcal{V}}$ (and hence the entire limit (3.44) as long as $\mathcal{V}$ is finite) to tend to zero.

The next section explores how we can choose an appropriate learning space (model selection), for a particular data set, using the concept of VC-Dimension.

3.5 Structural Risk Minimization (SRM)

We must redesign the machine for each different size of training data, and we must have some clever way of picking the right complexity a priori to avoid the above trade off.

So far we have considered PAC bounds for single fixed hypothesis classes; we can apply these PAC bounds individually to a whole collection of hypothesis classes and in this way select a space (or model) that best suits the current training data set.

The first step in SRM is defining a nested sequence of spaces $S_1\subset S_2\cdots\subset S_k$ such that they have increasing capacity, as measured by the VC-dimension,: $\mathcal{V}(S_1)\leq\mathcal{V}(S_2)\leq\cdots\leq\mathcal{V}(S_k)$. For instance in a classification task, we could

take the following sequence of linear functions:

$$\begin{aligned} S_1 &= \{f : f(\vec{x}) = \text{sgn}[b + w_1 x_1]\} \\ S_2 &= \{f : f(\vec{x}) = \text{sgn}[b + w_1 x_1 + w_2 x_2]\} \\ &\vdots \\ S_{\mathfrak{t}} &= \{f : f(\vec{x}) = \text{sgn}[b + (\vec{w} \cdot \vec{x})]\} \end{aligned}$$

where $\mathfrak{t}$ is the size of an observation vector and the VC-Dimension increases linearly and is equal to the number of free parameters; $\mathcal{V}(S_1) = 2, \mathcal{V}(S_2) = 3, \cdots, \mathcal{V}(S_{\mathfrak{t}}) = \mathfrak{t} + 1$. Alternatively, we could define the following sequence of families of non-linear classification functions:

$$\begin{aligned} S_1 &= \{f : f(\vec{x}) = \text{sgn}[b + (\vec{w} \cdot \vec{x})]\} \\ S_2 &= \{f : f(\vec{x}) = \text{sgn}[b + (\vec{w} \cdot \vec{x}) + (\vec{w} \cdot \vec{x})^2]\} \\ &\vdots \end{aligned}$$

We could also consider a sequence of linear classification functions with bounded weight vectors:

$$\begin{aligned} S_1 &= \{f : f(\vec{x}) = \text{sgn}[b + (\vec{w} \cdot \vec{x})] \text{ such that } 2/\|\vec{w}\| \leq \mathcal{R}_1\} \\ S_2 &= \{f : f(\vec{x}) = \text{sgn}[b + (\vec{w} \cdot \vec{x})] \text{ such that } \mathcal{R}_1 < 2/\|\vec{w}\| \leq \mathcal{R}_2\} \\ &\vdots \end{aligned} \tag{3.45}$$

or we can reformulate it in terms of the geometric margin:

$$\begin{aligned} S_1 &= \{f : f(\vec{x}) = \text{sgn}[b + (\vec{w} \cdot \vec{x})] \text{ such that } \gamma^* \geq g_1\} \\ S_2 &= \{f : f(\vec{x}) = \text{sgn}[b + (\vec{w} \cdot \vec{x})] \text{ such that } \gamma^* \geq g_2 \geq g_1\} \\ &\vdots \end{aligned} \tag{3.46}$$

The choice of nested models to use can be made by considering a priori information about the classification/regression task, for instance if the data is assumed to be non-linearly distributed then we can consider polynomial classification/regression functions of increasing degree; however this decision *must* be made before the training set is generated so as to satisfy the VC condition of distribution-independence. However, choice of the geometric margin

depends on the training set; so technically, SRM cannot be applied to SV classification where maximizing the geometric margin is essential. See [STB98] for alternatives.

If a particular family is too simple (where the VC-Dimension is low) then the empirical risk will likely be high since it becomes difficult to correctly classify the entire training set; on the other hand if the family is too complex then the VC-Confidence Interval will be large. So the next step in the SRM procedure is to find an optimal parametrization for each space using the empirical risk minimization methodology and then finally to add this minimized empirical risk to the value of the PAC bound (3.43) on the generalization error for the space in question:

$$R_{srm}(S_i) = \min_{f \in S_i} \left(\hat{R}_n(f) \right) + \sqrt{\frac{\mathcal{V}(S_i) \log \frac{en}{\mathcal{V}(S_i)} + \log \frac{1}{\delta}}{n}} \tag{3.47}$$

The family that minimizes the above expression has optimal generalization potential since we have an optimal balance between the capacity of the family in question (measured before generation of the training set) and the empirical risk (within each S_i and hence is dependent on the training set); it is optimal because moving to the next space in the sequence S_{i+1} does not reduce the empirical risk sufficiently to accommodate the increase in capacity ($\mathcal{V}(S_{i+1}) - \mathcal{V}(S_i)$) and moving to the previous space in the sequence S_{i-1} increases the empirical risk beyond the decrease in the capacity of the hypothesis space. As we have seen in section 2.4 this is essentially a regularization method.

4

Support Vector Machines for Binary Classification

Provided with n input vectors in the Hilbert space $\mathcal{H} = \mathbb{R}^d$ and their corresponding binary annotations:

$$\mathcal{S} = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \cdots, (\vec{x}_n, y_n)\} \subseteq \mathbb{R}^d \times \{+1, -1\} = \mathcal{H} \times \mathcal{Y}$$

all of which are identically and independently distributed (iid) according to some probability distribution $P(\vec{x}, y) = P(\vec{x}) \cdot P(y|\vec{x})$, we seek a *prediction function* $f(\vec{x})$ that will predict the correct annotation in the presence of noise:

$$y_t = \max_{y \in \{+1,-1\}} P(y|\vec{x}_t)$$

for a test example $\vec{x}_t$. The search for an *optimal* prediction function $f(\vec{x})$ is usually performed in a restricted functional space using the principle of empirical risk minimization (ERM) as outlined in the previous chapter.

For binary classification the zero-one loss function

$$\ell[f, \{\vec{x}, y\}] = |f(\vec{x}) - y| \tag{4.1}$$

may be used in which case the expected risk is then just the *probability of misclassification*; to see this note that the loss function $|f(\vec{x}) - y|$ can be

written as $(1 - \mathbb{I}_{f(x),y})$ and then the expected risk

$$\begin{aligned} R_{\mathcal{X}}[f] &= \int_{\mathcal{X}\times\mathcal{Y}} 1 - \mathbb{I}_{f(x),y}\, dP(\vec{x}, y) \\ &= \int_{\mathcal{X}\times\mathcal{Y}} 1\, dP(\vec{x}, y) - \int_{\mathcal{X}\times\mathcal{Y}} \mathbb{I}_{f(x),y}\, dP(\vec{x}, y) \\ &= 1 - \int_{\mathcal{X}\times\mathcal{Y}} \mathbb{I}_{f(x),y}\, dP(\vec{x}, y) \end{aligned} \tag{4.2}$$

is simply 1 minus the total probability of generating training examples which have been correctly classified: $\mathbb{I}_{f(x),y} = 1$.

The zero-one loss function is not a Lipschitz function (definition 3.4.5); it is discontinuous and scale insensitive; it is also impossible to provide a confidence in the classifiers predictions. The hinge loss is υ-insensitive to scale and is given by:

$$\ell_\upsilon[f, \{\vec{x}, y\}] = \max[0, \upsilon - yf(\vec{x})] \tag{4.3}$$

so that only those points whose classification we have a high confidence in (are at least υ away from the decision boundary) do not contribute to the loss, even if they are correctly classified. In the rest of this chapter we consider optimality conditions for (and justify our choice of) prediction functions of the form $f(\vec{x}|\vec{w}, b) = \texttt{sgn}\,[(\vec{w} \cdot \vec{x}) + b]$ so that the empirical risk is given by:

$$\hat{R}_n[f] = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\vec{x}|\vec{w}, b), \{\vec{x}, y\}) \tag{4.4}$$

4.1 Geometry of the Dot Product

We begin by defining a linear function in a real-valued, pre-Hilbert (inner product) space $\mathcal{H} = \mathbb{R}^d$, parameterized in terms of a weight vector $\vec{w} \in \mathbb{R}^d$ and a threshold or bias $b \in \mathbb{R}$ (a total of $d + 1$ free parameters):

$$\mathfrak{h}(\vec{x}) = (\vec{w} \cdot \vec{x}) + b$$

The *scalar resolute* is the length of the perpendicular projection of $\vec{x}$ onto $\vec{w}$ and is given by the dot product

$$(\hat{w} \cdot \vec{x}) = \|\vec{x}\| \cos\theta \tag{4.5}$$

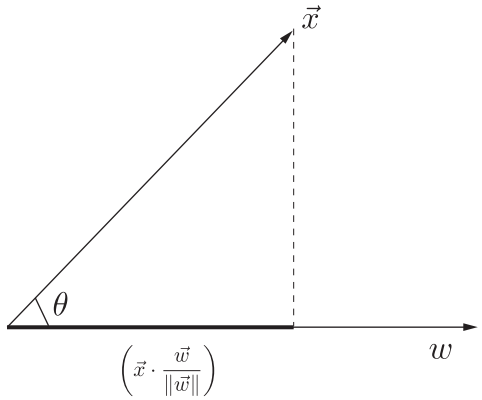

Figure 4–1: The inner product as a perpendicular projection

where $\hat{w} = \vec{w}/\|\vec{w}\|$ is a unit (normalized) vector and θ is the angle between $\vec{w}$ and $\vec{x}$. We can now rewrite the dot product in the definition of $\mathfrak{h}(\vec{x})$ as

$$(\vec{w} \cdot \vec{x}) = \|\vec{w}\| \, \|\vec{x}\| \, cos(\theta)$$

We can use the dot product as a similarity measure between two input vectors ($\vec{x}_1$ and $\vec{x}_2$) by comparing their corresponding dot products with some fixed weight vector $\vec{w}$; it is important to note that the dot product can distinguish between two vectors that lie in the same direction but have differing magnitudes:

$$\|\vec{x}_1\| = \|\vec{x}_2\| \text{ and } \theta_1 = \theta_2 \Longrightarrow (\vec{w} \cdot \vec{x}_1) = (\vec{w} \cdot \vec{x}_2) \Longrightarrow \vec{x}_1 = \vec{x}_2$$

$$\|\vec{x}_1\| \neq \|\vec{x}_2\| \text{ and } \theta_1 = \theta_2 \Longrightarrow (\vec{w} \cdot \vec{x}_1) \neq (\vec{w} \cdot \vec{x}_2) \Longrightarrow \vec{x}_1 \neq \vec{x}_2$$

where θ_1 (and θ_2) is the angle between $\vec{x}_1$ (and $\vec{x}_2$) and $\vec{w}$. Similarly the dot product can also distinguish between two vectors that have the same magnitude but lie in different directions:

$$\theta_1 = \theta_2 \text{ and } \|\vec{x}_1\| = \|\vec{x}_2\| \Longrightarrow (\vec{w} \cdot \vec{x}_1) = (\vec{w} \cdot \vec{x}_2) \Longrightarrow \vec{x}_1 = \vec{x}_2$$

$$\theta_1 \neq \theta_2 \text{ and } \|\vec{x}_1\| = \|\vec{x}_2\| \Longrightarrow (\vec{w} \cdot \vec{x}_1) \neq (\vec{w} \cdot \vec{x}_2) \Longrightarrow \vec{x}_1 \neq \vec{x}_2$$

But if neither the magnitude or direction of two input vectors is equal then it is impossible to make any general inferences about the equality of the vectors since we can decrease $\cos(\theta_1)$ (increase θ_1) and then increase the magnitude of a vector $\vec{x}_1$ by an equal amount to produce a new vector $\vec{x}_2$ that has the same

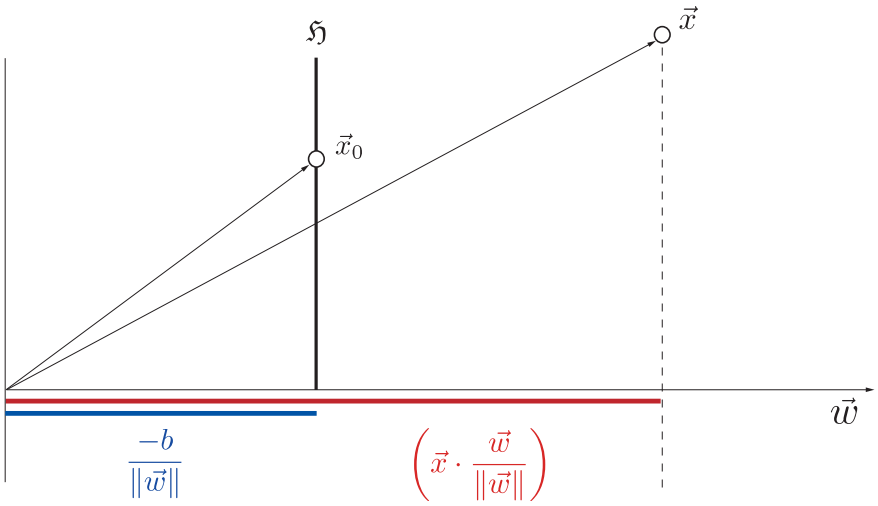

Figure 4–2: The distance of a point $\vec{x}$ from the hyperplane $\mathfrak{H}$ is the difference between the length of the perpendicular projection of $\vec{x}$ on $\vec{w}$ and the distance of the hyperplane from the origin: $\left(\vec{x} \cdot \frac{\vec{w}}{\|\vec{w}\|}\right) - \frac{-b}{\|\vec{w}\|}$.

dot product;

$$(\vec{w} \cdot \vec{x}_1) = (\vec{w} \cdot \vec{x}_2) \nRightarrow \vec{x}_1 = \vec{x}_2 \tag{4.6}$$

This is an inherent weakness of the dot product.

Since (4.5) is a scalar it does not have direction; the *vector resolute* combines the scalar value $(\hat{w} \cdot \vec{x})$ with the direction of $\vec{w}$ and is given by multiplying the scalar resolute by $\hat{w}$:

$$\frac{\vec{w}}{\|\vec{w}\|}\left(\frac{\vec{w}}{\|\vec{w}\|} \cdot \vec{x}\right)^{\top} \tag{4.7}$$

4.2 Regulating the Hypothesis Space

The primary concern in binary classification is dividing the input space into two half-spaces, one each corresponding to the positive and negative classes; a hyperplane in an affine subspace of dimension $d-1$ achieves this:

$$\mathfrak{H} = \{\vec{x} \in \mathbb{R}^d : \mathfrak{h}(\vec{x}) = 0\} \tag{4.8}$$

so that the positive and negative classes are defined as the disjoint subspaces $\{\vec{x} \mid \mathfrak{h}(\vec{x}) > 0\}$ and $\{\vec{x} \mid \mathfrak{h}(\vec{x}) < 0\}$ respectively. From this definition of the hyperplane we see that the weight vector $\vec{w}$ is perpendicular to $\mathfrak{H}$ since for any two points $\vec{x}_1$ and $\vec{x}_2$ satisfying 4.8 we have that $(\vec{x}_1 - \vec{x}_2) \cdot \vec{w} = 0 \Rightarrow (\vec{x}_1 - \vec{x}_2) \perp \vec{w}$, while the scalar bias b translates $\mathfrak{H}$

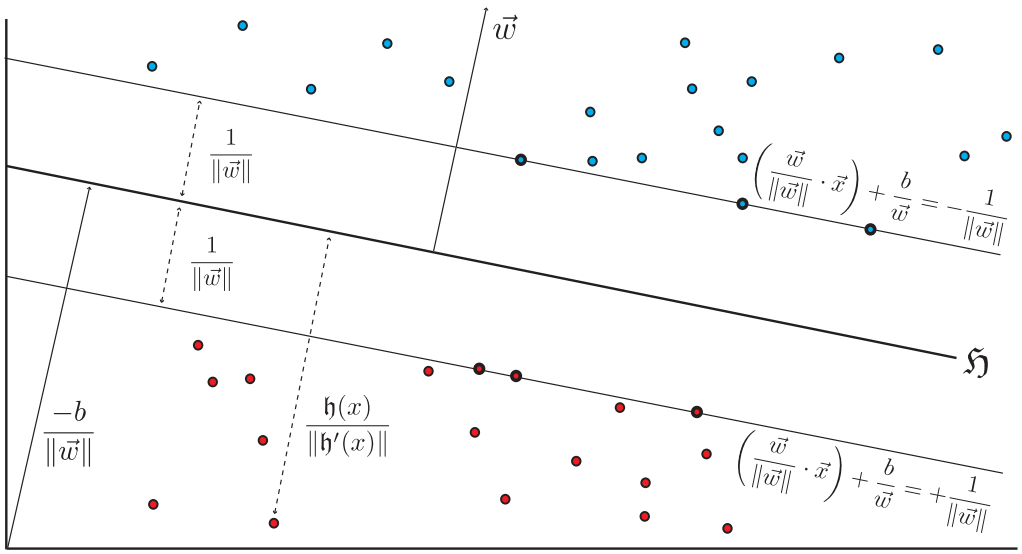

Figure 4–3: The margin boundaries $\mathfrak{H}_+$ and $\mathfrak{H}_-$ lie on either side of the classification boundary $\mathfrak{H}$ and are defined by the support vectors. The geometrical margin for the canonical hyperplane $\mathfrak{H}$ is $1/\|\vec{w}\|$; the distance of a point $\vec{x}$ from $\mathfrak{H}$ is $\mathfrak{h}(\vec{x})/\|\mathfrak{h}'(\vec{x})\|$; changes to the bias term b cause the hyperplane $\mathfrak{H}$ to shift in a perpendicular direction.

in a parallel direction so that the perpendicular distance of $\mathfrak{H}$ from the origin is $-b/\|\vec{w}\|$. The distance of a point $\vec{x}$ from the hyperplane $\mathfrak{H}$ can then be calculated as follows; take any point $\vec{x}_0$ (see Figure 4–2) on the hyperplane $\mathfrak{H}$, then calculate the euclidean distance between the perpendicular projections of $\vec{x}$ and $\vec{x}_0$ on $\vec{w}$:

$$\begin{aligned}\left(\frac{\vec{w}}{\|\vec{w}\|}\cdot\vec{x}\right)-\left(\frac{\vec{w}}{\|\vec{w}\|}\cdot\vec{x}_0\right) &= \frac{1}{\|\vec{w}\|}[(\vec{w}\cdot\vec{x})-(\vec{w}\cdot\vec{x}_0)]\\ &= \frac{1}{\|\vec{w}\|}[(\vec{w}\cdot\vec{x})+b]\\ &= \frac{1}{\|\mathfrak{h}'(\vec{x})\|}\mathfrak{h}(\vec{x})\end{aligned}$$

where the second equality follows from (4.8).

4.2.1 Discriminant Hyperplanes

In the previous chapter we saw that regularization methods (section 2.4) bound the capacity of hypothesis spaces to ensure well-posedness (uniqueness, existence and stability) of the ERM solution, which as a result is able to generalize well to unannotated test examples since it does not over-fit the training data. In the following sections the search for a suitable prediction function is restricted to the hypothesis space of discriminant hyperplanes:

$$\mathcal{J} = \left\{ \mathfrak{H} : \ \forall \vec{w}, b \ \in \ \mathbb{R}^{n+1} \right\} \tag{4.9}$$

This is the first of three restrictions that are placed on the hypothesis space; of these three restrictions only the first and third will affect the capacity (VC-Dimension) of the hypothesis space. An extension to non-linear discriminant surfaces, through kernelizing the algorithm, is detailed in section 2.5; in this case the restriction 4.9, as well as restrictions 4.10 and 4.14, are removed from the input space and applied instead to an expanded space or feature space $\mathcal{F}$.

The training data is said to be *linearly separable* when there exists some hyperplane that can divide the input space $\mathcal{X}$ such that each half-space contains examples with identical annotations; in this case the empirical risk is zero since no training example is miss-classified.

4.2.2 Canonical Hyperplanes

Multiplying both $\vec{w}$ and b by the same scalar constant doesn't change the orientation or position of a hyperplane although its parametric representation does change since the function $\mathfrak{h}(\vec{x})$ changes; the VC-Dimension of each of these parameterizations are the same since they define the same hyperplane. We arbitrarily select a unique representation from amongst this infinite class of parameterizations by isolating the so called *canonical hyperplane* that is parametrized such that the points closest to it are a distance of 1 away:

$$\min_{\vec{x}_i \in \mathcal{S}} |(\vec{w} \cdot \vec{x}_i) + b| = 1 \tag{4.10}$$

This is the second restriction on the hypothesis space. It is important to note that any separating hyperplane can be transformed into a canonical hyperplane by multiplying the parameters $\vec{w}$ and b by the inverse of the perpendicular distance from the hyperplane to the nearest training example.

Training data points that satisfy $|(\vec{w}, \vec{x}_i)+b| = 1$ are called *support vectors*; these vectors shoulder the hyperplanes $\mathfrak{H}_+ = \{\vec{x} : (\vec{w} \cdot \vec{x}_i) + b = +1\}$ and $\mathfrak{H}_- = \{\vec{x} : (\vec{w} \cdot \vec{x}_i) + b = -1\}$ on either side of $\mathfrak{H}$ and in doing so define the *margin* or space between $\mathfrak{H}_+$ and $\mathfrak{H}_-$.

Although we have already fixed the distance between two support vectors $\vec{x}_+^{sv}$ and $\vec{x}_-^{sv}$, one each on the hyperplanes $\mathfrak{H}_+$ and $\mathfrak{H}_-$, by removing the scaling freedom of the parameters; it is useful to view this distance in geometric terms

by taking the difference between their normalized perpendicular projections onto $\vec{w}$

$$\begin{aligned}\left(\frac{\vec{w}}{\|\vec{w}\|}\cdot\vec{x}_{+}^{sv}\right)-\left(\frac{\vec{w}}{\|\vec{w}\|}\cdot\vec{x}_{-}^{sv}\right)=&\frac{1-b}{\|\vec{w}\|}-\frac{-1-b}{\|\vec{w}\|}\\=&\frac{2}{\|\vec{w}\|}\end{aligned}\tag{4.11}$$

Since two points $(\frac{\vec{w}}{\|\vec{w}\|}\cdot\vec{x}_{+}^{sv})$ and $(\frac{\vec{w}}{\|\vec{w}\|}\cdot\vec{x}_{-}^{sv})$ on opposite sides of $\mathfrak{H}$ (both of which lie on the vector $\vec{w}$) are $\frac{2}{\|\vec{w}\|}$ apart, they must each be $\frac{1}{\|\vec{w}\|}$ away from $\mathfrak{H}$; the size of the margin is now an expression of n of the $(n+1)$ parameters that define the classification boundary $\mathfrak{H}$ (excluding the bias b) - this is convenient since it is now possible to define an optimization in terms of $\vec{w}$ to regulate both the orientation of $\mathfrak{H}$ as well as the size of the margin.

Definition 4.2.1 *The functional (perpendicular, signed) distance between a training example $(\vec{x}_i, y_i)$ and a hyperplane $\mathfrak{H}$ is:*

$$\gamma_i \equiv y_i\ \mathfrak{h}(\vec{x}_i)$$

The functional margin γ between a set of training examples $\mathcal{S}$ and a hyperplane $\mathfrak{H}$ is then simply the minimum over all functional distances between $\mathfrak{H}$ and each example in $\mathcal{S}$:

$$\gamma \equiv \min_{(x_i,y_i)\in\mathcal{S}} \gamma_i = \min_{(x_i,y_i)\in\mathcal{S}} y_i\ \mathfrak{h}(\vec{x}_i)$$

Definition 4.2.2 *The geometric (normalized, euclidean) distance between the hyperplane $\mathfrak{H}$ and $\vec{x}_i$ is:*

$$\gamma_i^* \equiv y_i\ \left[\left(\frac{\vec{w}}{\|\vec{w}\|}\cdot\vec{x}_i\right)+\frac{b}{\|\vec{w}\|}\right]$$

The geometric margin γ^* is the minimum over all geometric distances γ_i^* between $\mathfrak{H}$ and each example in $\mathcal{S}$.

The classification of a test example $(\vec{x}_t, y_t)$ can be verified through the condition $\gamma_t > 0$ since $\mathfrak{h}(\vec{x}_t) > 0$ is the subspace associated with $y_t > 0$ and $\mathfrak{h}(\vec{x}_t) < 0$ is the subspace associated with $y_t < 0$; the resulting classification rule or decision/prediction function is defined as:

$$y_t = f(\vec{x}_t) = \mathtt{sgn}\left[\mathfrak{h}(\vec{x}_t)\right] = \mathtt{sgn}\left[(\vec{w}\cdot\vec{x}_t)+b\right]\tag{4.12}$$

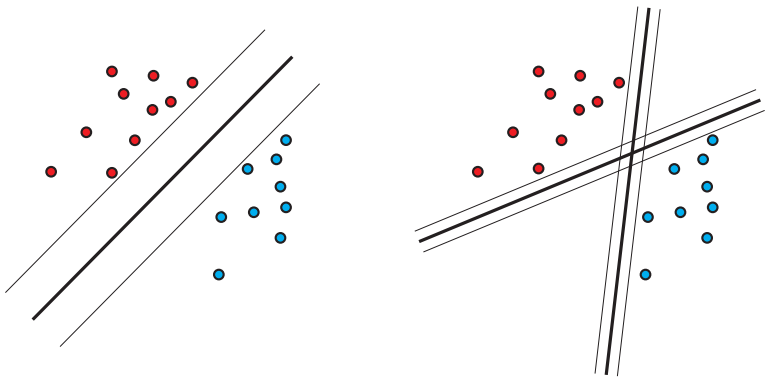

Figure 4–4: As the size of the margin (as indicated by the margin boundaries $\mathfrak{H}_+$ and $\mathfrak{H}_-$) decreases, the number of possible separating hyperplanes increases implying an increase in the VC-Dimension.

If the positive and negative training examples can be separated using a hyperplane then it must also be the case that the geometric margin is positive; the converse also holds so that if the geometric margin is negative then the training data has not been *linearly separated* by the current hyperplane. In the remainder of this section as well as the next, it is assumed that the training examples are linearly separable.

4.2.3 Maximal Margin Hyperplanes

The final restriction is the toughest to deal with and will eventually require us to solve a quadratic optimization whose unique solution is the separating hyperplane that has the highest generalization potential. Now assume the training set is sparse and real-valued; it is then possible to apply an infinitesimally small transformation (rotation or translation) to any canonical separating hyperplane to generate a new canonical separating hyperplane whose geometric margin is different. So the existence of a single separating hyperplane implies the existence of an infinite class of distinct canonical separating hyperplanes all with varying geometric margins. From amongst this infinite set we must select a single generalizable hyperplane using insight provided by the training data. The following theorem links the generalization potential of a hyperplane with its margin;

Theorem 4.2.1 (Vapnik, 1995) *Let the hypersphere enclosing the entire training data set* $\mathcal{S}$ *have radius* r *so that* $\|\vec{x}\| \leq r$*. Then the VC-Dimension of the space* $\mathcal{J}_\rho$ *of canonical hyperplanes with bounded weight vectors* $\|\vec{w}\| \leq \rho$ *is*

given by:

$$\mathcal{V}(\mathcal{J}_\rho) \leq \min\left\{4r^2\rho^2, n\right\} + 1$$

Since the VC-Dimension is finite, consistency of the classifier is guaranteed.

Maximizing the margin (4.11) reduces the value of the norm of the weight vector $\|\vec{w}\|$; hence we can lower the value of the upper bound ρ on the weight vector. From the above theorem we see that the VC-Dimension is reduced once we enlarge the margin so that $\mathcal{V}(\mathcal{J}) \geq \mathcal{V}(\mathcal{J}_\rho)$ is satisfied, where $\mathcal{J}$ is the space of canonical hyperplanes with *unbounded* weight vectors. Now if we consider the PAC bound (3.43) we see that for a training set of size n there is a particular hypothesis space (with a particular VC-Dimension) that minimizes the generalization bound; since we can control the VC-Dimension (by adjusting the margin) we can perform structural risk minimization (SRM) for the sequence of spaces given in (3.45) to find the optimal capacity $\mathcal{V}(\mathcal{J}_\rho)$ for a given training data set. Intuitively, the further away from the margin boundaries (beyond which the classification changes) a test example is, the more confident we are in its predicted classification and so we would like all training examples to be as far away from the separating hyperplane which basically amounts to maximizing the margin.

Finally, it is important to note that the maximum margin hyperplane is constructed on the basis of the positions of the support vectors alone in whose predicted classification we are not entirely confident since they are closest to the decision boundary; whilst making predictions the rest of the training examples may be ignored and this leads to significant generalization.

4.3 Hard Margin Classifiers

Based on our choice of parameters for the canonical hyperplane, for which $\gamma = 1$, we have already shown that $\gamma^* = \frac{1}{\|\vec{w}\|}$. To summarize, the following inequalities defined in terms of the functional

$$\gamma_i = y_i \left[(\vec{x}_i \cdot \vec{w}) + b\right] \geq 1$$

and geometric margins:

$$\gamma_i^* = y_i \left[\left(\vec{x}_i \cdot \frac{\vec{w}}{\|\vec{w}\|}\right) + \frac{b}{\|\vec{w}\|}\right] \geq \frac{1}{\|\vec{w}\|} \tag{4.13}$$

are satisfied for all training data examples. The maximal margin hyperplane $\{\vec{x} : \mathfrak{h}(\vec{x}) = (\vec{w}^* \cdot \vec{x}) + b^* = 0\}$ is then given [SS01] by the following optimization for the parameters $\vec{w}^*$ and b^*:

$$\begin{aligned}
\vec{w}^*, b^* &= \underset{\vec{w},b}{\operatorname{argmax}} \{\gamma^*\} \\
&= \underset{\vec{w},b}{\operatorname{argmax}} \left\{ \min_{i=1}^{n} \gamma_i^* \right\} \\
&= \underset{\vec{w},b}{\operatorname{argmax}} \left\{ \min_{i=1}^{n} y_i \left[\left(\frac{\vec{w}}{\|\vec{w}\|} \cdot \vec{x}_i \right) + \frac{b}{\|\vec{w}\|} \right] \right\} \\
&= \underset{\vec{w},b}{\operatorname{argmax}} \left\{ \min_{i=1}^{n} y_i \left[\mathtt{sgn}\left(\left(\frac{\vec{w}}{\|\vec{w}\|} \cdot \vec{x}_i \right) + \frac{b}{\|\vec{w}\|} \right) \left\| \left(\frac{\vec{w}}{\|\vec{w}\|} \cdot \vec{x}_i \right) + \frac{b}{\|\vec{w}\|} \right\| \right] \right\} \\
&= \underset{\vec{w},b}{\operatorname{argmax}} \left\{ \min_{i=1}^{n} y_i \; f(\vec{x}) \left\| \left(\frac{\vec{w}}{\|\vec{w}\|} \cdot \vec{x}_i \right) + \frac{b}{\|\vec{w}\|} \right\| \right\} \\
&= \underset{\vec{w},b}{\operatorname{argmax}} \left\{ \min_{i=1}^{n} y_i \; f(\vec{x}) \left\| \left(\frac{\vec{w} \cdot \vec{x}_i}{\|\vec{w}\|^2} \vec{w} \right) + \frac{b}{\|\vec{w}\|^2} \vec{w} \right\| \right\}
\end{aligned} \tag{4.14}$$

where the fourth equality follows from splitting a vector into its sign and size components. The last equality includes a norm taken over the sum of two vectors; the first is the vector resolute defined in (4.7) and the second $b\vec{w}/\|\vec{w}\|^2$ has the same direction as $\vec{w}$ and ends right on the boundary of the hyperplane $\mathfrak{H}$ since the perpendicular projection of this vector onto $\vec{w}$ is also a distance of $-b/\|\vec{w}\|$ from the origin;

$$\left(\frac{b}{\|\vec{w}\|^2} \vec{w} \right) \cdot \frac{\vec{w}}{\|\vec{w}\|} = \frac{-b}{\|\vec{w}\|} \tag{4.15}$$

Geometrically, the optimization attempts to maximise the difference in lengths of (4.7) and $b\vec{w}/\|\vec{w}\|^2$ and thereby maximizes the margin. Finally the constraints defined in (4.13) are included as part of the optimization;

$$\vec{w}^*, b^* = \underset{\vec{w},b}{\operatorname{argmax}} \{\gamma^*\} \;\; \mathtt{subject\ to} \;\; y_i \left[(\vec{x}_i \cdot \vec{w}) + b \right] \geq 1 \;\; \forall i \tag{4.16}$$

Using (4.13) we can rewrite this as a minimization in terms of the weight vector;

$$\vec{w}^*, b^* = \underset{\vec{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\vec{w}\| \right\} \;\; \mathtt{subject\ to} \;\; y_i \left[(\vec{x}_i \cdot \vec{w}) + b \right] \geq 1 \;\; \forall i \tag{4.17}$$

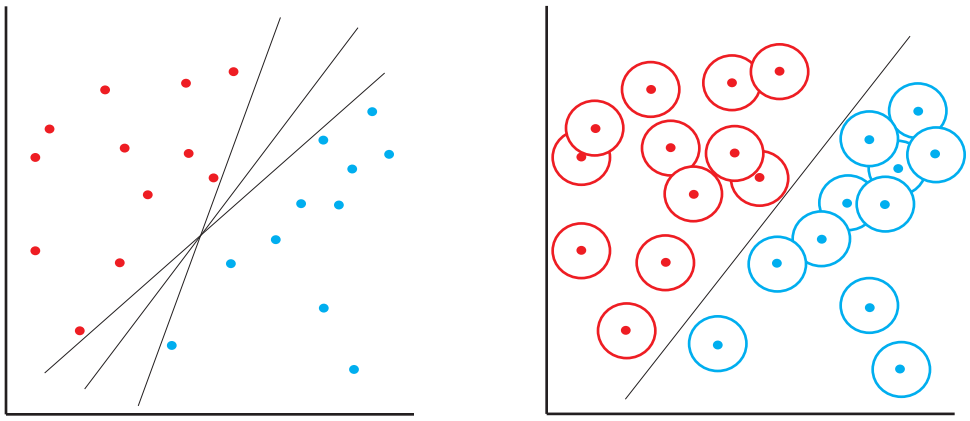

Figure 4–5: The minimum distance between a canonical separating hyperplane and a data point is $r = 1/\|\vec{w}\|$ so we can enclose each training point in a hyper-sphere of radius r, so that the hyperplane does not bisect any hyper-sphere. [left] many hyperplane classifiers are admissible [right] as the radius of the hyper-sphere increases, the number of admissible hyperplanes decreases. So maximizing the margin leads to a restricted hypothesis space with lower VC-Dimension. Intuitively we expect that when a training example $\vec{x}$ is far away from the margin boundary, then small perturbations in the training space $\vec{x}_t = (\vec{x} + \epsilon)$ should leave the classification unchanged: $f(\vec{x}) = f(\vec{x}_t)$. Additionally, small perturbations to the parameters $(\vec{w} + \epsilon)$ are also more likely to leave the classification unchanged.

4.4 Soft Margin Classifiers

So far we have assumed that the geometric margin is positive; in such cases the training data is said to be *linearly separable*; when this is not the case we make use of *margin slack variables* ξ_i (that allow the training data to cross either the *margin boundaries* $\mathfrak{H}_+$ and $\mathfrak{H}_-$ or the *classification boundary* $\mathfrak{H}$) which are then used to define relaxed inequality constraints;

$$y_i \left[(\vec{x}_i \cdot \vec{w}) + b\right] \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{4.18}$$

There are three possible values for ξ_i:

i $\xi_i = 0$: x_i is correctly classified; it lies on or beyond the margin boundary for its class

ii $0 < \xi_i \leq 1$: x_i is correctly classified; it lies between the margin boundary and the classification boundary for its class

iii $\xi_i > 1$: x_i has been misclassified: it lies on the wrong side of the classification boundary

So we see that the classification of a training example (using 4.12) is correct only when its geometric margin is positive in which case its associated slack variable is less than or equal to 1. An upper bound on the *training*

classification error (or empirical risk under the zero-one loss function) is given by the norm of the margin slack vector $\|\vec{\xi}\|$ since satisfying condition (ii) does not constitute a miss-classification.

Even in cases where the data is linearly separable it might not be optimal to restrict the search to only those hyperplanes that satisfy (4.13); for example training data may include a single noisy outlier which should be ignored (in the sense that we modify its functional margin so that it becomes a support vector and affects choice of the hyperplane as such), which is essentially what (4.18) achieves with non-zero margin slack variables. However by making all ξ_i large enough it is possible to satisfy all the constraints defined in (4.18) for any choice of hyperplane and so it is therefore crucial to restrict the size of the margin slack variables by constraining $\|\vec{\xi}\|$.

We consider optimizing two quantities; maximizing the size of the margin while minimizing the size of the margin slack variables subject to the constraints defined in (4.18). We can define an optimization based on these criteria by modifying (4.17) so that we have the following which is said to be in its primal form:

$$\begin{aligned} \vec{w}^*, b^*, \xi^* &= \underset{\vec{w},b,\vec{\xi}}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\vec{w}\| + C\|\vec{\xi}\| \right\} \\ &\quad \texttt{subject to } y_i \left[(\vec{x}_i \cdot \vec{w}) + b \right] \geq 1 - \xi_i, \quad \xi_i \geq 0 \ \ \forall i \end{aligned} \tag{4.19}$$

We must now reexamine how the maximum margin is constructed; lets assume we have the hyperplane from the optimization above parametrized in terms of $\vec{w}^*, b^*, \xi^*$ - it is clear that only a fraction of those examples satisfying $\xi_i^* = 0$ serve as support vectors, specifically those which lie on the margin boundary. This is in contrast to those points satisfying $\xi_i^* \neq 0$ which are all support vectors since they are forced onto the margin boundary of their class. So the choice of parameters in (4.19) are affected by all vectors satisfying (ii) and (iii) and a subset of those satisfying (i).

We must also scrutinize the affects of the parameter C on the results of the primal optimization; as its value decreases it gradually switches from constraining the training classification error to showing a preference for maximal margin hyperplanes instead; so as C decreases the size of hypothesis space diminishes which in turn reduces the computational complexity and run-time of the optimization. When the value of C is high enough so that non-zero margin

slack variables are highly penalized, the resulting hyperplane is equivalent to the hard margin hyperplane.

So we have shown that the optimal hyperplane in a binary classification task has maximal geometric margin and can be found by optimizing the primal form given in (4.19), in the following section we will see that finding such a hyperplane is in its dual form, a quadratic programming problem.

4.5 Quadratic Programming

Using the method of Lagrange multipliers for nonlinear constrained optimizations, we define the Lagrangian Λ as the objective function plus a linear combination of the constraints:

$$\Lambda(\vec{w}, b, \vec{\xi} \mid \vec{\alpha}) = \frac{1}{2}\|w\| + C\|\vec{\xi}\| - \sum_{i=1}^{n} \alpha_i \left(y_i \langle \vec{x}_i \cdot \vec{w} \rangle + y_i b - 1 + \xi_i\right) \tag{4.20}$$

where $\alpha_i \geq 0$, $\beta_i \geq 0$ are dual variables or Lagrange multipliers which must be non-negative since this is implied by the non-negativity of their corresponding constraints:

$$y_i(\vec{x}_i \cdot \vec{w}) + y_i b - 1 + \xi_i \geq 0 \implies \alpha_i \geq 0$$

Now we can rewrite (4.19) in its dual form as an unconstrained maximization over the dual (Lagrange) variables:

$$\vec{\alpha}^* = \underset{\vec{\alpha}}{\operatorname{argmax}} \left\{ \underset{\vec{w},b,\vec{\xi}}{\operatorname{argmin}} \Lambda(\vec{w}, b, \vec{\xi} \mid \vec{\alpha}) \right\} \tag{4.21}$$

To find the minimum we differentiate the Lagrangian with respect to the parameters $\vec{w}, b$ and $\vec{\xi}$ and set it equal to zero:

$$\begin{aligned} \frac{\partial \Lambda}{\partial \vec{w}} = 0 \quad &\implies \quad \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i = \vec{w} \\ \frac{\partial \Lambda}{\partial b} = 0 \quad &\implies \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \frac{\partial \Lambda}{\partial \vec{\xi}} = 0 \quad &\implies \quad \vec{\alpha} = 2C\vec{\xi} \end{aligned} \tag{4.22}$$

Incorporating the first and third of the above into the dual form gives:

$$\begin{aligned}
\Lambda(\vec{w}, b, \vec{\xi} \mid \vec{\alpha}) &= \frac{1}{2}\left\|\sum_{i=1}^{n}\alpha_i y_i \vec{x}_i\right\| + C\left\|\frac{\vec{\alpha}}{2C}\right\| \\
&- \sum_{i=1}^{n}\left(\alpha_i y_i \langle \vec{x}_i \cdot \vec{w}\rangle + \alpha_i y_i b - \alpha_i + \alpha_i \xi_i\right) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle y_i \cdot y_j\rangle\langle \vec{x}_i \cdot \vec{x}_j\rangle + \frac{\langle \vec{\alpha}\cdot\vec{\alpha}\rangle}{4C} \\
&\quad -\sum_{i=1}^{n}\alpha_i y_i\left\langle \vec{x}_i \cdot \sum_{i=1}^{n}\alpha_i y_i \vec{x}_i\right\rangle + \sum_{i=1}^{n}\alpha_i - \langle \vec{\alpha}\cdot\vec{\xi}\rangle \\
&= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle y_i \cdot y_j\rangle\langle \vec{x}_i \cdot \vec{x}_j\rangle + \frac{\langle \vec{\alpha}\cdot\vec{\alpha}\rangle}{4C} + \sum_{i=1}^{n}\alpha_i - \left\langle \vec{\alpha}\cdot\frac{\vec{\alpha}}{2C}\right\rangle \\
&= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle y_i \cdot y_j\rangle\langle \vec{x}_i \cdot \vec{x}_j\rangle - \frac{\langle \vec{\alpha}\cdot\vec{\alpha}\rangle}{4C} + \sum_{i=1}^{n}\alpha_i
\end{aligned}$$

So our final dual quadratic optimization is given by:

$$\vec{\alpha}^* = \underset{\vec{\alpha}}{\operatorname{argmax}}\left\{-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle y_i \cdot y_j\rangle\langle \vec{x}_i \cdot \vec{x}_j\rangle - \frac{\langle \vec{\alpha}\cdot\vec{\alpha}\rangle}{4C} + \sum_{i=1}^{n}\alpha_i\right\} \quad (4.23)$$

subject to the constraints; $\alpha_i \geq 0, \forall i$ and $\sum_{i=1}^{n}\alpha_i y_i = 0$. Typical quadratic optimizers solve the following minimization:

$$\vec{\alpha}^* = \underset{\vec{\alpha}}{\operatorname{argmin}}\left(\vec{s}^{\top}\vec{\alpha} + \frac{1}{2}\vec{\alpha}^{\top}H\vec{\alpha}\right) \quad (4.24)$$
$$\texttt{subject to:} \qquad A\vec{\alpha} = \vec{b} \ \texttt{and} \ \vec{l} \leq \vec{\alpha} \leq \vec{u}$$

We can rewrite (4.23) as a quadratic minimization in the matrix form given above as:

$$\vec{\alpha}^* = \underset{\vec{\alpha}}{\operatorname{argmin}}\left\{\frac{1}{2}\vec{\alpha}^{\top}\left((\vec{y}\vec{y}^{\top})\odot(XX^{\top}) + \frac{1}{2C}\right)\vec{\alpha} - \mathbb{I}^{\top}\vec{\alpha}\right\}$$
$$\texttt{subject to:} \quad \vec{y}^{\top}\vec{\alpha} = 0 \ \texttt{and} \ \alpha_i \geq 0 \forall i \quad (4.25)$$

There are several methods for solving this optimization, some more efficient than others; refer to [Pla98], [CBM02], [MM01] and [Joa99].

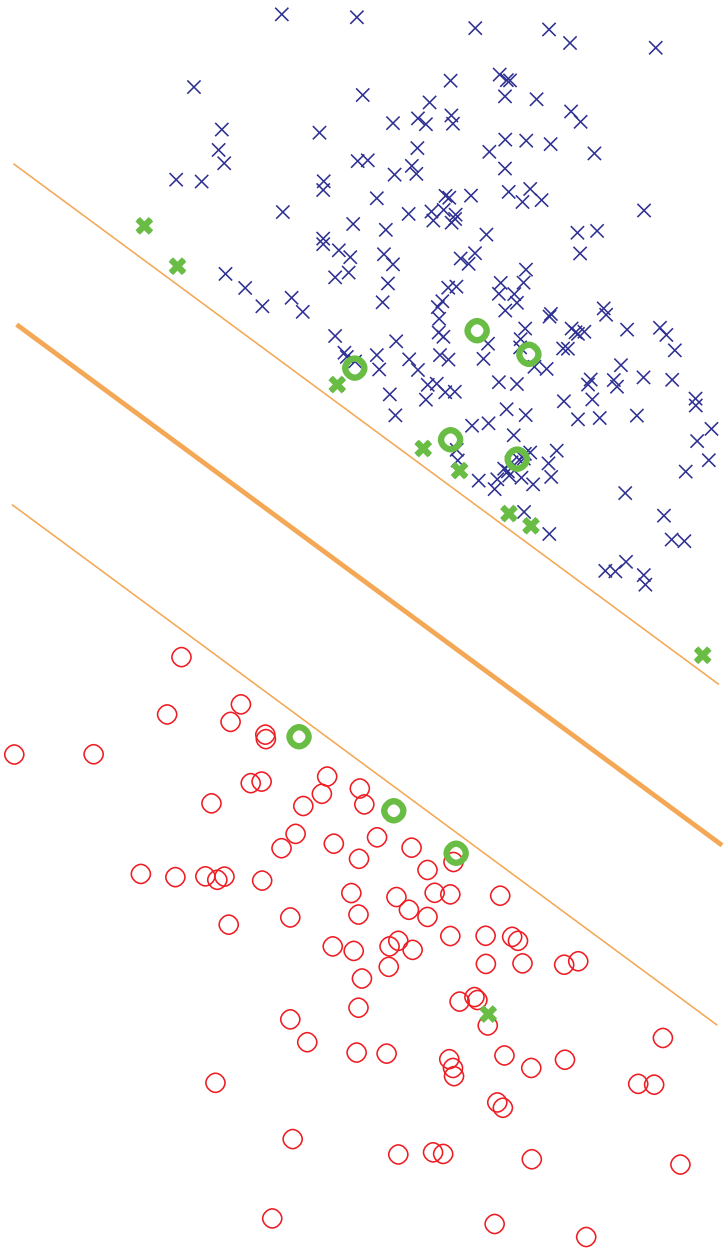

Figure 4–6: Results of binary classification task; 17 support vectors (green) define the decision boundary which separates the positive (blue) from the negative (red) examples. Notice that all the training examples that are misclassified by the learnt decision boundary serve as support vectors.

5

Support Vector Machines for Regression

Support Vector Machine Regression (SVMR) is similar to SVM Classification (SVMC) in that the regression function that it learns is *linear* in some higher dimensional feature space and *non-linear* in the input space. The learnt function deviates the least from the training data amongst all such linear surfaces in the expanded space, according to some loss function. As an example consider the ϵ-tube loss function:

$$\ell_\epsilon(f, \{x_i, y_i\}) = \begin{cases} 0 & \text{when } |y_i - f(\vec{x}_i)| \leq \epsilon \\ |y_i - f(\vec{x}_i)| - \epsilon & \text{otherwise} \end{cases} \tag{5.1}$$

We have already seen how to build optimal, linear decision boundaries in the feature space in the previous chapter on SVMC for a binary classification task. Now given a training set where the annotation space is real-valued $\mathcal{Y} = \mathbb{R}$, we will still consider linear surfaces of the form:

$$f(\vec{x}) = \langle \vec{w} \cdot \vec{x} \rangle + b$$

where $\vec{w} : \mathcal{X} \to \mathcal{Y}$ is a linear operator and $b \in \mathcal{Y}$ is a bias vector [SSTPH05]. However instead of attempting to separate and then maximise the region between the two classes, we will require that the input vectors are positioned within an ϵ-tube around any hyperplane under consideration; the inputs failing to satisfy this will contribute positively to the loss. Ideally as the ϵ-tube is reduced in size, we would like to find the linear regression surface that has minimal loss. Following from the optimization defined in (4.17) we define the following *quadratic* optimization:

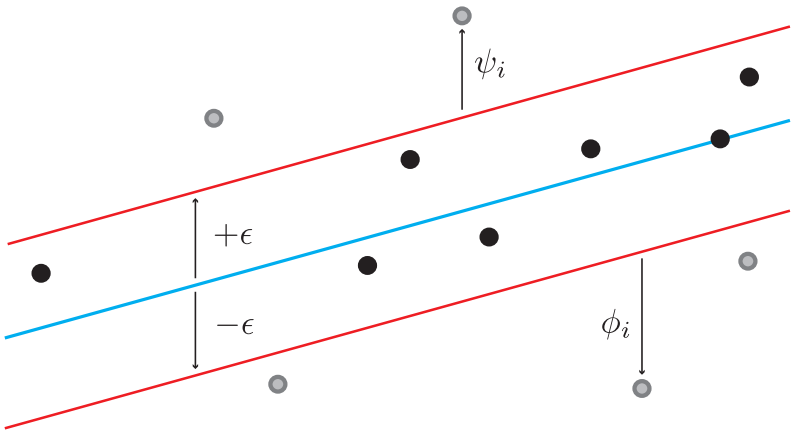

Figure 5–1: Training examples within the ϵ-tube (in black) do not incur a loss although those examples outside it (in gray) do with the loss increasing linearly as a function of the distance from the ϵ-tube.

$$\begin{aligned} \min_{\vec{w}} \ \Gamma(\vec{w}) &= \tfrac{1}{2}\vec{w}\vec{w}^T \\ \text{subject to:} \quad & y_i - \langle \vec{w} \cdot \vec{x}_i \rangle - b \leq \epsilon \\ & \langle \vec{w} \cdot \vec{x}_i \rangle + b - y_i \leq \epsilon \end{aligned} \tag{5.2}$$

It is possible that for a given value of ϵ no function satisfying the constraint $|f(\vec{x}_i) - y_i| \leq \epsilon$ exists. So we define slack variables $\psi_i > 0$ and $\phi_i > 0$ and re-write the (primal) optimization as:

$$\begin{aligned} \min_{\vec{w},\vec{\psi},\vec{\phi}} \ \Gamma(\vec{w},\vec{\psi},\vec{\phi}) &= \frac{1}{2}\vec{w}\vec{w}^T + \zeta \sum_i^n (\psi_i + \phi_i) \\ \text{subject to:} \quad & y_i - \langle \vec{w} \cdot \vec{x}_i \rangle - b \leq \epsilon + \psi_i \\ & \langle \vec{w} \cdot \vec{x}_i \rangle + b - y_i \leq \epsilon + \phi_i \\ & \psi_i \geq 0, \phi_i \geq 0, \forall n \end{aligned} \tag{5.3}$$

This is a soft version of the previous (5.2) optimization similar to (4.19); the constant $\zeta \in \mathbb{R}$ maintains the trade-off between how much deviation outside the ϵ-tube is permitted versus the generalization or in this case the flatness of the regression function.

5.1 Langrangian Dual Formulation for Regression

Instead of solving the primal optimization, we will work with its dual form which often has a structure thats easier to work with and in many instances also has a more intuitive interpretation. We begin by defining the Lagrangian Λ as a linear combination of the objective function and the various equality/inequality constraints of the optimization (5.3):

$$\begin{aligned}\Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi}|\vec{\alpha}, \vec{\beta}) &= \frac{1}{2}\|w\|^2 + \zeta \sum_{i=1}^{n}(\psi_i + \phi_i) \\ &- \sum_{i=1}^{n} \alpha_i(\epsilon + \psi_i - y_i + \langle \vec{w}, \vec{x}_i \rangle + b) \\ &- \sum_{i=1}^{n} \beta_i(\epsilon + \phi_i + y_i - \langle \vec{w}, \vec{x}_i \rangle - b) \end{aligned} \tag{5.4}$$

where α_i and β_i are non-negative dual variables or Lagrange multipliers. The dual objective function Ω of the original optimization is defined as:

$$\Omega(\vec{\alpha}, \vec{\beta}) = \min_{w,b,\phi,\psi} \Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi}|\vec{\alpha}, \vec{\beta}) \tag{5.5}$$

and has a value of $-\infty$ when the Lagrangian is unbounded from below. The Lagrangian Dual, which in this particular case is still a quadratic optimization, is then given by:

$$\begin{aligned} &\max_{\vec{\alpha},\vec{\beta}} \ \Omega(\vec{\alpha}, \vec{\beta}) \\ &\text{subject to: } \vec{\alpha} \geq 0 \text{ and } \vec{\beta} \geq 0 \end{aligned} \tag{5.6}$$

More generally, it is easy to see that the dual of all linear or quadratic programs remain as such.

Weak duality is said [Wel07] to hold when any feasible dual solution lower bounds any feasible primal solution; in the case that they are equal it implies the optimality of both feasible solutions as we will see in the following theorem. Under certain conditions on the dual optimization, this lower bound is in fact *always* optimal and hence equal to the optimal primal solution; in such instances, *Strong Duality* is said [Boy07] to hold.

Theorem 5.1.1 (Weak Duality Theorem) *Let $(\vec{w}_f, b_f, \vec{\psi}_f, \vec{\phi}_f)$ be any feasible point for the primal and $(\vec{\alpha}_f, \vec{\beta}_f, \vec{\eta}_f, \vec{\eta^*}_f)$ any feasible point for the dual; it*

follows that the primal objective function Γ *and its dual* Ω *satisfy the following inequality:*

$$\Omega(\vec{\alpha}_f, \vec{\beta}_f) \leq \Gamma(\vec{w}_f, b_f, \vec{\psi}_f, \vec{\phi}_f) \tag{5.7}$$

Proof All feasible solutions of the dual must satisfy (5.5) and hence are minima of the Lagrangian function Λ:

$$\begin{aligned} \Omega(\vec{\alpha}_f, \vec{\beta}_f) &= \min_{w,b,\phi,\psi} \Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi} | \vec{\alpha}_f, \vec{\beta}_f) \\ &\leq \Lambda(\vec{w}_f, b_f, \vec{\phi}_f, \vec{\psi}_f | \vec{\alpha}_f, \vec{\beta}_f) \\ &\leq \Gamma(\vec{w}_f, b_f, \vec{\phi}_f, \vec{\psi}_f) \end{aligned}$$

where the last inequality follows from the definition of the Lagrangian (5.4), the positivity of the Lagrange multipliers and the constraints that define the primal optimization. □

Hence it follows that if the primal has a feasible solution then the dual objective function is bounded from above; alternatively if the dual is feasible then the primal is bounded from below. Furthermore, if the dual is unbounded from above ($\Omega = \infty$) then the primal is infeasible and if the primal is unbounded from below ($\Gamma = -\infty$) then the dual is infeasible.

The *duality gap* is the difference between the values of the primal Γ and dual Ω objective functions evaluated at some feasible primal and dual points respectively. The *optimal duality gap* is given by the difference between the optimal solutions of the primal and dual problems which still clearly satisfy (5.7):

$$\max_{\vec{\alpha},\vec{\beta}} \Omega(\vec{\alpha}, \vec{\beta}) \leq \min_{\vec{w},b,\vec{\psi},\vec{\phi}} \Gamma(\vec{w}, b, \vec{\psi}, \vec{\phi}) \tag{5.8}$$

Note that when the primal is a maximization and the dual is a minimization then the weak duality theorem gives us the opposite result, specifically that the primal objective function is bounded from above by the dual objective function. Finally, if the duality gap is zero for some feasible primal and dual points:

$$\Omega(\vec{\alpha}_f, \vec{\beta}_f) = \Gamma(\vec{w}_f, b_f, \vec{\psi}_f, \vec{\phi}_f) \tag{5.9}$$

it follows from the Weak Duality Theorem that $(\vec{w}_f, b_f, \vec{\psi}_f, \vec{\phi}_f)$ is an optimal primal solution while $(\vec{\alpha}_f, \vec{\beta}_f)$ is an optimal dual solution.[1] To see this note that if (5.9) holds then the dual objective function has attained its maximum (optimal) value (since it is bounded from above by the primal objective function) while the primal objective function has attained its minimum (optimal) value (since it is bounded from below by the dual objective function).

Definition 5.1.1 (Strong Duality) *When the existence of an optimal solution to the primal implies the existence of an optimal solution to the dual and vice versa, the optimal duality gap must be zero. In other words, the existence of an optimal primal solution* $(\vec{w}_o, \vec{\psi}_o, \vec{\phi}_o, \zeta_o)$ *implies the existence of Lagrange multipliers* $(\vec{\alpha}_o, \vec{\beta}_o)$ *satisfying*

$$\Omega(\vec{\alpha}_o, \vec{\beta}_o) = \Gamma(\vec{w}_o, b_o, \vec{\psi}_o, \vec{\phi}_o)$$

Definition 5.1.2 (Convex Optimization) *A convex optimization has a convex objective function, convex inequality constraints and linear equality constraints. Every strictly convex optimization has a unique solution.*

The objective function of the dual Ω is a *concave* (downward) function of the dual variables even when the primal objective function Γ is not convex (*concave upward*). This is because [Hin06] the dual is a point-wise minimum of a set of affine functions. Furthermore, when the primal problem is convex, then strong duality holds. Hence, in the case of quadratic programs which are always convex, the duality gap is always zero.

5.2 Complementary Slackness

Let $(\vec{\alpha}_o, \vec{\beta}_o)$ and $(\vec{w}_o, b_o, \vec{\psi}_o, \vec{\phi}_o)$ be optimal solutions of the dual and primal respectively. Then Strong Duality implies that $\Omega(\vec{\alpha}_o, \vec{\beta}_o) = \Gamma(\vec{w}_o, b_o, \vec{\psi}_o, \vec{\phi}_o)$.

[1] It is important to note that the converse is not necessarily implied: primal and dual objective functions evaluated at optimal primal and dual solutions need not be equal but must satisfy (5.8).

From (5.4) and (5.7) we then derive the KKT conditions:

$$\begin{aligned} \alpha_i(\epsilon + \psi_i - y_i + \langle \vec{w}, \vec{x}_i \rangle + b) &= 0 \\ \beta_i(\epsilon + \phi_i + y_i - \langle \vec{w}, \vec{x}_i \rangle - b) &= 0 \\ i &= 1, \cdots, n \end{aligned} \tag{5.10}$$

Proof

$$\begin{aligned} \Gamma(\vec{w}_o, b_o, \vec{\psi}_o, \vec{\phi}_o) &= \Omega(\vec{\alpha}_o, \vec{\beta}_o) \\ &= \min_{w,b,\phi,\psi} \Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi} | \vec{\alpha}_o, \vec{\beta}_o) \\ &\leq \Lambda(\vec{w}_o, b_o, \vec{\phi}_o, \vec{\psi}_o | \vec{\alpha}_o, \vec{\beta}_o) \\ &= \Gamma(\vec{w}_o, b_o, \vec{\phi}_o, \vec{\psi}_o) - \sum_{i=1}^{n} \alpha_i(\epsilon + \psi_i - y_i + \langle \vec{w}, \vec{x}_i \rangle + b) \\ &\quad - \sum_{i=1}^{n} \beta_i(\epsilon + \phi_i + y_i - \langle \vec{w}, \vec{x}_i \rangle - b) \\ &\leq \Gamma(\vec{w}_o, b_o, \vec{\phi}_o, \vec{\psi}_o) \end{aligned}$$

where the last inequality follows from the positivity of both the Lagrange multipliers and the constraints so that the following is implied:

$$\Gamma(\vec{w}_o, b_o, \vec{\phi}_o, \vec{\psi}_o) = \Lambda(\vec{w}_o, b, \vec{\phi}_o, \vec{\psi}_o | \vec{\alpha}_o, \vec{\beta}_o)$$

A constraint is said to be active or tight if for an optimal primal solution $(w_o, b_o, \phi_o, \psi_o)$ its corresponding Lagrange multiplier is strictly positive which implies that the constraint evaluated at the optimal solution is zero:

$$\begin{aligned} \epsilon + \phi_i - y_i + f(\vec{x}_i) = 0 \text{ implies } \alpha_i > 0 \\ \epsilon + \psi_i + y_i - f(\vec{x}_i) = 0 \text{ implies } \beta_i > 0 \end{aligned} \tag{5.11}$$

Constraints are otherwise said to be inactive:

$$\begin{aligned} \epsilon + \phi_i - y_i + f(\vec{x}_i) > 0 \text{ implies } \alpha_i = 0 \\ \epsilon + \psi_i + y_i - f(\vec{x}_i) > 0 \text{ implies } \beta_i = 0 \end{aligned} \tag{5.12}$$

The $\vec{x}_i$ with non-zero α_i or β_i are called *support vectors*; if we were to train the SVM on only these $\vec{x}_i$, ignoring all the examples for which $\alpha_i = 0$ and $\beta_i = 0$, we would still induce the same regression surface.

THEOREM 5.2.1 (LAGRANGIAN SADDLEPOINT EQUIVALENCE THEOREM) *If the conditions for strong duality are satisfied (i.e. the optimal duality gap is zero and hence the complementary slackness conditions are satisfied) then the optimal primal and dual solutions must be saddle-points of the Lagrangian Λ; modifying the optimal primal solution will not decrease the Lagrangian and similarly modifying the optimal value of the Lagrange multipliers will not increase the Lagrangian. The converse also holds so that if the Lagrangian has a saddle-point then there is no optimal duality gap (Strong Duality) which in turn implies that the complementary slackness conditions are satisfied.*

Proof By definition the dual optimization is given by:

$$(\alpha_o, \beta_o) = \max_{\vec{\alpha}\geq 0, \vec{\beta}\geq 0} \min_{w,b,\phi,\psi} \Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi}|\vec{\alpha}, \vec{\beta}) \tag{5.13}$$

It is easy to see that maximizing the Lagrangian over the dual variables, which can be set to zero in the case that either $(\epsilon + \psi_i - y_i + \langle \vec{w}, \vec{x}_i \rangle + b) > 0$ or $(\epsilon + \phi_i + y_i - \langle \vec{w}, \vec{x}_i \rangle - b) > 0$, for any feasible primal solution $(\vec{w}_f, b_f, \vec{\phi}_f, \vec{\psi}_f)$ is equal to the primal objective function evaluated at the same feasible primal solution:

$$\max_{\vec{\alpha}\geq 0, \vec{\beta}\geq 0} \Lambda(\vec{w}_f, b_f, \vec{\phi}_f, \vec{\psi}_f|\vec{\alpha}, \vec{\beta}) = \Gamma(\vec{w}_f, b_f, \vec{\phi}_f, \vec{\psi}_f)$$

As a result the primal optimization can be rewritten in terms of the Lagrangian as follows:

$$(\vec{w}_o, b_o, \vec{\phi}_o, \vec{\psi}_o) = \min_{w,b,\phi,\psi} \max_{\vec{\alpha}\geq 0, \vec{\beta}\geq 0} \Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi}|\vec{\alpha}, \vec{\beta}) \tag{5.14}$$

Since there is no optimal duality gap when strong duality holds we therefore have:

$$\min_{w,b,\phi,\psi} \max_{\vec{\alpha}\geq 0, \vec{\beta}\geq 0} \Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi}|\vec{\alpha}, \vec{\beta}) = \max_{\vec{\alpha}\geq 0, \vec{\beta}\geq 0} \min_{w,b,\phi,\psi} \Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi}|\vec{\alpha}, \vec{\beta})$$

So we can change the order of minimization and maximization and still arrive at the same optimal solution which must therefore be a saddle-point. □

We can identify the saddle-points of the Lagrangian by differentiating it with respect to the primal variables and setting the result equal to zero:

$$\begin{aligned}
\frac{\partial \Lambda}{\partial b} = 0 \quad &\Longrightarrow \quad \sum_{i=1}^{n}(\beta_i - \alpha_i) = 0 \\
\frac{\partial \Lambda}{\partial \vec{w}} = 0 \quad &\Longrightarrow \quad \vec{w} - \sum_{i=1}^{n}(\alpha_i - \beta_i)\vec{x}_i = 0 \\
\frac{\partial \Lambda}{\partial \phi_i} = 0 \quad &\Longrightarrow \quad \zeta - \beta_i = 0 \\
\frac{\partial \Lambda}{\partial \psi_i} = 0 \quad &\Longrightarrow \quad \zeta - \alpha_i = 0
\end{aligned} \tag{5.15}$$

To remove the dependence on the primal variables we substitute (5.15) into the Lagrangian (5.4):

$$\begin{aligned}
\Lambda(\vec{w}, b, \vec{\phi}, \vec{\psi} | \vec{\alpha}, \vec{\beta}) \;=\; & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \beta_i)(\alpha_j - \beta_j)\langle \vec{x}_i \cdot \vec{x}_j \rangle \\
+ \; & \sum_{i=1}^{n}\psi_i(\zeta - \alpha_i) + \sum_{i=1}^{n}\phi_i(\zeta - \alpha_i) \\
- \; & \epsilon\sum_{i=1}^{n}(\alpha_i + \beta_i) \\
+ \; & \sum_{i=1}^{n} y_i(\alpha_i - \beta_i) + \sum_{i=1}^{n}(\beta_i - \alpha_i)b \\
+ \; & \sum_{i=1}^{n}(\beta_i - \alpha_i)\left\langle \sum_{j=1}^{n}(\alpha_j - \beta_j)\vec{x}_j \cdot \vec{x}_i \right\rangle
\end{aligned}$$

So the dual optimization can be given entirely in terms of the dual variables as:

$$\begin{aligned}
\max_{\vec{\alpha},\vec{\beta}} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \beta_i)(\alpha_j - \beta_j)\langle \vec{x}_i \cdot \vec{x}_j \rangle \\
& -\epsilon\sum_{i=1}^{n}(\alpha_i + \beta_i) + \sum_{i=1}^{n} y_i(\alpha_i - \beta_i) \\
\text{subject to:} \quad & \sum_{i=1}^{n}(\alpha_i - \beta_i) = 0 \\
& \alpha_i, \beta_i \in [0, \zeta]
\end{aligned} \tag{5.16}$$

We can rewrite (5.16) as a quadratic minimization in matrix form (4.24):

$$(\vec{\alpha}^*, \vec{\beta}^*) = \underset{\vec{\alpha},\vec{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \begin{bmatrix} \vec{\alpha} \\ \vec{\beta} \end{bmatrix}^\top \begin{bmatrix} (XX^\top) & -(XX^\top) \\ -(XX^\top) & (XX^\top) \end{bmatrix} \begin{bmatrix} \vec{\alpha} \\ \vec{\beta} \end{bmatrix} + \begin{bmatrix} \epsilon\mathbb{I} - \vec{y} \\ \epsilon\mathbb{I} + \vec{y} \end{bmatrix}^\top \begin{bmatrix} \vec{\alpha} \\ \vec{\beta} \end{bmatrix} \right\}$$
$$\texttt{subject to:} \quad \begin{bmatrix} \mathbb{I} \\ -\mathbb{I} \end{bmatrix}^\top \begin{bmatrix} \vec{\alpha} \\ \vec{\beta} \end{bmatrix} = 0 \ \texttt{and} \ \alpha_i, \beta_i \in [0, \zeta] \tag{5.17}$$

The primal solution can be given in terms of the dual solution (5.15) when strong duality holds which is convenient since the dual optimization is typically easier to solve that the primal.

5.3 Sparse Support Vector Expansion

The regression surface can also given entirely in terms of the dual variables as:

$$f(\vec{x}) = \langle \vec{w} \cdot \vec{x} \rangle + b = \sum_{i=1}^{n} (\alpha_i - \beta_i) \langle \vec{x}_i \cdot \vec{x} \rangle + b$$

Let $\Psi \subseteq \{1, 2, \cdots, n\}$ such that $\forall i \in \Psi$ we have *both* $\alpha_i > 0$ and $\beta_i > 0$. Then we can rewrite our regression function using a sparse expansion as:

$$f(\vec{x}) = \sum_{i \in \Psi} (\alpha_i - \beta_i) \langle \vec{x}_i \cdot \vec{x} \rangle + b \tag{5.18}$$

Prediction functions that are defined using a sparse expansion are able to generalize far better since they consider only the most 'important' training points or support vectors; in the case of binary classification the support vectors were those points that lie along the margin boundaries and are therefore closest to the separating hyperplane. For regression, the support vectors are those points that lie on or beyond the boundary of the epsilon tube and are hence furthest away from the regression surface.

5.4 Non-Linear SVM Regression

The machine we have described so far is linear but the data itself might be distributed non-linearly. As previously described in section (2.5), we first implicitly apply a mapping function ϕ to our input data, essentially projecting

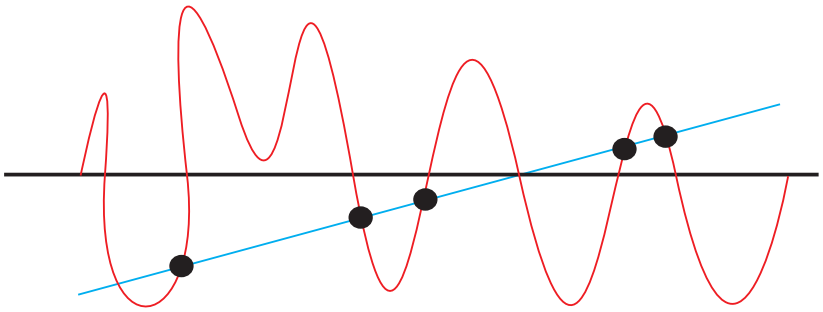

Figure 5–2: Over-fitting the training data; both functions pass through all five training points however the linear hypothesis is more likely to accurately predict the annotation of a test example.

it into a higher dimensional feature space $\mathcal{H}$ and then apply our linear machinery to find a linear regression in this new feature space. The corresponding regression surface in the input space will be non-linear. Explicitly, we make use of kernel functions that replace all dot products between feature vectors and in this way perform all computation in the input space while learning a linear regression surface in a higher dimensional feature space.

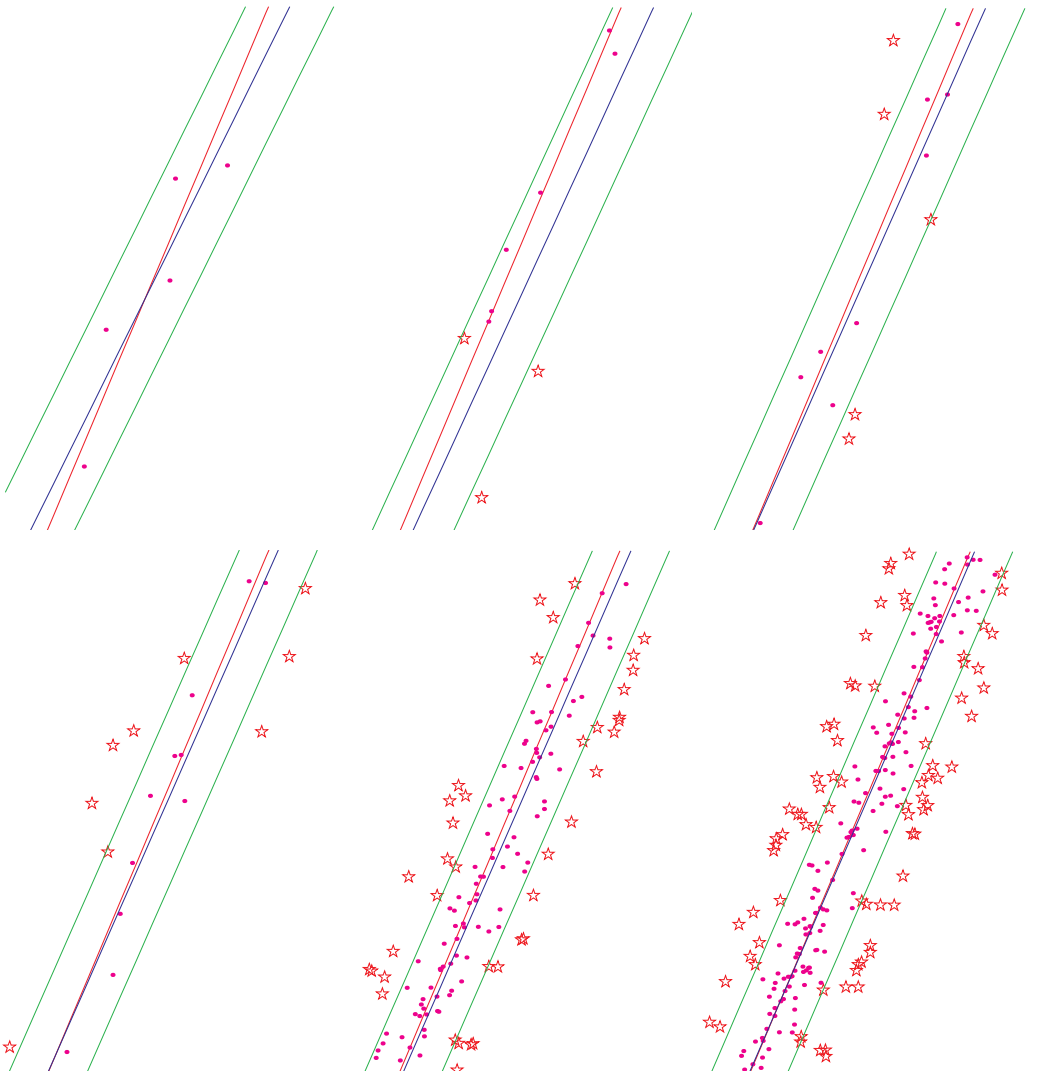

Figure 5–3: Training samples were randomly (normally) generated in the region around the target function (red line). The learnt regression function (blue line) approximates the target function better as the number of training samples increases, i.e. its slope and bias approach that of the target function. The support vectors (red stars) lie outside the ϵ-tube (green lines) while the other data points (red points) lie within it.

6

Conclusion

A linear methodology for performing classification and regression has been described in detail starting with a discussion on kernel methods which when used in conjunction with SVMs are able to extend it making non-linear classification and regression possible. The kernel trick is also described, which replaces inner-products in the feature space with a kernel evaluation in the input space so that the SVM operates in a reproducing kernel Hilbert space. Subsequent discussions have focused on statistical learning theory which describe the circumstances under which learning is possible and on the actual mechanics of Support Vector classification and regression.

The theoretical basis of Support Vector Machines has been researched intensively in the last few years. Advances include the use of new task specific kernel functions, quicker evaluation of the decision/prediction function and calibrating the SVM solution as a posterior probability. Advances in optimization theory have led to faster training methods such as the Sequential Minimal Optimization decomposition method [Pla98]. Many new applications of Support Vector Machines have also emerged including detecting remote protein homologies, forecasting weather, speaker verification, face detection and chaotic time series prediction, in particular estimating the price of derivative securities.

References

[Ama95] S. Amari. Learning and statistical inference. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 522–526. MIT Press, Cambridge, Massachusetts, 1995.

[BBL03] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 169–207. Springer, 2003.

[Boy07] Stephen Boyd. Ee364a: Convex optimization 1, 2007. Notes, Stanford University, `www.stanford.edu/class/ee364`.

[BTA04] Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

[Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. In *Knowledge Discovery and Data Mining*, volume 2, pages 121–167. 1998.

[CBM02] R. Collobert, S. Bengio, and J. Mariethoz. Torch: a modular machine learning software library, 2002.

[CGGR05] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox. Perception Systemes et Information, INSA de Rouen, Rouen, France, 2005.

[Che97] V. Cherkassky. The nature of statistical learning theory. *IEEE Transactions on Neural Networks*, 8(6):1564–1564, November 1997.

[CMR02] Stephane Canu, Xavier Mary, and Alain Rakotomamonjy. Functional learning through kernel, October 25 2002.

[CS02] Cucker and Smale. On the mathematical foundations of learning. *BAMS: Bulletin of the American Mathematical Society*, 39, 2002.

[CST00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods).* Cambridge University Press, March 2000.

[CST04] N. Cristianini and J. Shawe-Taylor. *Kernel Methods for Pattern Analysis*. Cambridge University Press, June 2004.

[CSTS98] N. Cristianini, J. Shawe-Taylor, and P. Sykacek. Bayesian classifiers are large margin hyperplanes in a Hilbert space. In *Proc. 15th International Conf. on Machine Learning*, pages 109–117. Morgan Kaufmann, San Francisco, CA, 1998.

[CT01] Lijuan Cao and Francis Eng Hock Tay. Financial forecasting using support vector machines. *Neural Computing and Applications*, 10(2):184–192, 2001. `www.springerlink.com/index/10.1007/s005210170010`.

[DGZ91] Dudley, Gin, and Zinn. Uniform and universal glivenko-cantelli classes. *Journal of Theoretical Probability*, 4(3):485–510, 1991.

[DM05] Tevian Dray and Corinne A. Manogue. The geometry of the dot and cross product. 2005.

[EP99] Evgeniou and Pontil. On the v_γ dimension for regression in reproducing kernel hilbert spaces. In *ALT: International Workshop on Algorithmic Learning Theory*, 1999.

[EPP00] Evgeniou, Pontil, and Poggio. Statistical learning theory: A primer. *IJCV: International Journal of Computer Vision*, 38, 2000.

[Fel68] William Feller. *An introduction to probability theory and its applications. - Vol. 1.* Wiley, 1968. Feller.

[Fuk72] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1972.

[Gir97] Federico Girosi. An equivalence between sparse approximation and support vector machines. Technical Report AIM-1606, Massachusetts Institute of Technology, 1997.

[Gun98] Steve Gunn. Support vector machines for classification and regression. Technical report, University of Southampton, April 07 1998.

[Hin06] Haitham Hindi. A tutorial on convex optimization ii: duality and interior point methods. *American Control Conference*, pages 11 –, 2006.

[HN01] John Hunter and Bruno Nachtergaele. *Applied Analysis.* World Scientific Publishing Company, 2001. `www.math.ucdavis.edu/~hunter/book/`.

[Hoc73] Harry Hochstadt. *Integral Equations.* John Wiley and Sons, New York, NY, USA, 1973.

[HTH01] Trevor Hastie, Robert Tibshirani, and J. H.Friedman. *The Elements of Statistical Learning.* Springer, 2001.

[Ihl03] Alexander Ihler. Kernel density estimation toolbox for matlab, 2003. `ssg.mit.edu/~ihler/code/kde.shtml`.

[Joa99] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, 1999.

[LV07] Marcel Luthi and Thomas Vetter. Machine learning (cs331) class notes, 2007. Notes, Universitat Basel, `informatik.unibas.ch/lehre/ss07/cs331/resources/`.

[MM01] O. L. Mangasarian and David R. Musicant. Lagrangian support vector machines. *Journal of Machine Learning Research*, 1:161–177, 2001.

[Muk07] Sayan Mukherjee. Statistical learning: Algorithms and theory, 2007. Notes, Duke University, `www.stat.duke.edu/~sayan/statlearn.pdf`.

[MZ00] C. Molina and J. Zerubia. Regularisation by convolution in probability density estimation is equivalent to jittering. In *Proc. IEEE International Workshop on Neural Networks for Signal Processing (NNSP)*, Sydney, Australie, December 2000.

[Nea96] Radford M. Neal. *Bayesian Learning for Neural Networks*. Number 118 in Lecture Notes in Statistics. Springer, New York, USA, 1996.

[Pla98] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines, 1998.

[PMRR04] Tomaso Poggio, Sayan Mukherjee, Ryan Rifkin, and Alexander Rakhlin. Statistical learning theory and applications, 2004. Notes, MIT, `www.mit.edu/~9.520/spring04/`.

[Qui01] Gene Quinn. Reisz representation theorem, 2001. Notes, University of Rhode Island, `www.math.uri.edu/~quinn/web/mth629_Reisz.pdf`.

[Rak06] Alexander Rakhlin. *Applications of Empirical Processes in Learning Theory: Algorithmic Stability and Generalization Bounds*. PhD thesis, MIT, 2006.

[Rud91] Walter Rudin. *Functional Analysis*. McGraw-Hill, New York, NY, USA, 1991.

[SBSC99] Alex J. Smola, Peter Bartlett, Bernhard Schölkopf, and C.Schuurmans. *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 1999.

[Seb77] G. A. F. Seber. *Linear Regression Analysis*. John Wiley, New York, 1977.

[SHS01] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 2001.

[SS01] Bernhard Scholköpf and Alex J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[SS04] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[SS05] Bernhard Schölkopf and Alex Smola. *Support Vector Machines and Kernel Algorithms*. John Wiley & Sons, 2005.

[SSM98] Alex J. Smola, Bernhard Schölkopf, and Klaus-Robert Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649, 1998.

[SSTPH05] Sandor Szedmak, John Shawe-Taylor, and Emilio Parado-Hernandez. Learning via linear operators: Maximum margin regression. *PASCAL*, October 04 2005. `eprints.pascal-network.org/archive/00001765/`.

[STB98] J. Shawe-Taylor and P. L. Bartlett. Structural risk minimization over data-dependent hierarchies. *IEEE Trans. on Information Theory*, 44(5):1926–1940, 1998.

[Vap96] Vapnik. Statistical theory of generalization (abstract only). In *ICML: Machine Learning: Proceedings of the Seventh International Conference, 1990*, 1996.

[Vap99] V. N. Vapnik. An overview of statistical learning theory. *IEEE-NN*, 10(5):988, September 1999.

[Vap00] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 2nd edition, 2000.

[Wai04] Martin Wainwright. Stat 260: Nonlinear and convex optimization, 2004. Notes, University of Berkeley, `www.eecs.berkeley.edu/~wainwrig/ee227a/`.

[Wan05] Lipo Wang, editor. *Support Vector Machines: Theory and Applications*, volume 177 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, 2005.

[Wel07] Max Welling. Essentials of convex optimization, 2007. Notes, University of California - Irvine, `www.ics.uci.edu/~welling/classnotes`.

[WGS+99] J. Weston, A. Gammerman, M. O. Stitson, V.Vapnik, V. Vovk, and C. Watkins. Support vector density estimation, 1999. `www.cs.rhbnc.ac.uk/~jasonw/density.ps`.

[Zho02] Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18, 2002.

[Zho03] Ding-Xuan Zhou. Capacity of reproducing kernel spaces in learning theory. *IEEE Transactions on Information Theory*, 49(7):1743–1752, 2003.