Bayesian Learning in Reproducing Kernel Hilbert Spaces

Ralf Herbrich
Computer Science Department
Technical University of Berlin
10587 Berlin, Germany

Thore Graepel
Computer Science Department
Technical University of Berlin
10587 Berlin, Germany

Colin Campbell
Department of Engineering Mathematics
Bristol University
Bristol BS8 1TR, United Kingdom

6. July 1999

TR 99–11

Department of Computer Science
Technical University of Berlin, Franklinstr. 28/29, 10587 Berlin
(for more infomation look at http://stat.cs.tu-berlin.de/publications)

Abstract

Support Vector Machines find the hypothesis that corresponds to the centre of the largest hypersphere that can be placed inside *version space*, i.e. the space of all consistent hypotheses given a training set. The boundaries of version space touched by this hypersphere define the support vectors. An even more promising approach is to construct the hypothesis using the whole of version space. This is achieved by the *Bayes point*: the midpoint of the region of intersection of all hyperplanes bisecting version space into two volumes of equal magnitude. It is known that the centre of mass of version space approximates the Bayes point [30]. The centre of mass is estimated by averaging over the trajectory of a billiard in version space. We derive bounds on the generalisation error of Bayesian classifiers in terms of the volume ratio of version space and parameter space. This ratio serves as an *effective* VC dimension and greatly influences generalisation. We present experimental results indicating that *Bayes Point Machines* consistently outperform Support Vector Machines. Moreover, we show theoretically and experimentally how Bayes Point Machines can easily be extended to admit training errors.

1 Introduction

Recently, there has been considerable interest in the theory and application of Support Vector Machines (SVMs) [28]. Compared to neural networks they have a number of advantages. For example, the hypothesis modelling the data is explicitly represented in terms of the most informative patterns (the support vectors), the learning task amounts to optimisation of a Lagrangian which is provably convex, and they exhibit good generalisation, a property which is motivated by theoretical results from statistical learning theory [21, 28]. The SVM classifier corresponds to the centre of the largest inscribable hypersphere in *version space*, i.e. the space of all hypotheses consistent with the training data. Those boundaries of version space with which the hypersphere makes tangential contact correspond to the *support vectors*.

A potentially better approach is to use all of version space to define the hypothesis. As illustrated in Figure 4 this is a superior strategy if the version space is elongated and asymmetric. Here SVMs are condemned to fail. We will consider learning machines based on approximating the *Bayes point*, i.e. the midpoint of the region of intersection of all hyperplanes which divide version space into two halves of equal volume. This approach is purely Bayesian: if we consider a new test point $\mathbf{x}$, the set of Bayes–optimal

decision functions is given by those weight vectors $\mathbf{w}$ whose posterior on a binary decision at $\mathbf{x}$ is greater than 0.5. As in general the intersection of Bayes–optimal decision functions for all $\mathbf{x}$ is empty we could approximate it by the centroid $\mathbf{w}_{\text{Bayes}}$ having knowledge of $P_{\mathcal{C}}(\mathbf{x})$. This hypothesis is called the Bayes point. It was shown elsewhere [30, 15] that in high–dimensional spaces $\mathbf{w}_{\text{Bayes}}$ converges to the centre of mass of version space. An additional insight into the usefulness of the Bayes point comes from the statistical mechanics approach to neural computing where the generalisation error for Bayesian learning algorithms has been calculated for the case of randomly constructed and unbiased patterns $\mathbf{x}$ [15]. Thus if ζ is the number of training examples per weight and ζ is large, the generalisation error of the centre of mass scales as $0.44/\zeta$ whereas scaling with ζ is poorer for the solutions found by the linear Support Vector Machine (maximally stable perceptron) (scales as $0.50/\zeta$ [16]), Adaline (scales as $0.24/\sqrt{\zeta}$ [17]) and other approaches.

The paper is structured as follows: in Section2 we revisit methods of learning linear classifiers. In Section 3 we introduce the *Bayes Point Machine* (BPM) algorithm which approximates the centre of mass of the version space $\mathcal{V}(S)$ by a billiard. In Section 4 we investigate bounds on the generalisation error for Bayesian classifiers, SVMs, and BPMs. In Section 5 we demonstrate one method to incorporate training errors in BPMs. Then, in Section 6 we present experimental results which support the usefulness of our approach. The detailed derivations have been relegated to the appendix for better legibility of the main document.

We denote the logarithm to base 2 by $\log_2$, and the natural logarithm by $\ln$. If S is a set set, $|S|$ denotes its cardinality. Vectors are denotes by bold letters, e.g. $\mathbf{w}$. Vector components are denoted by subscripts, e.g. w_i. The norm of a vector $\mathbf{w}$ in the metric space $\mathcal{F}$ is denoted by $\|\mathbf{w}\|_{\mathcal{F}}$, whereas the inner product between two elements $\mathbf{a}, \mathbf{b} \in \mathcal{F}$ is given by $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{F}}$. The symbols $\mathbb{R}$ and $\mathbb{N}$ denote the set of real and natural numbers, respectively. We do not explicitly state the measurability conditions needed for our arguments to hold. We assume with no further discussion "permissibility" of the function classes involved. We denote real–valued functions by f whereas binary classifiers obtained by thresholding are denoted by $h = \text{sign}(f)$.

2 Approaches to Learning Linear Classifiers

Let us consider the set of *kernel classifiers* [28, 31]

$$h(\mathbf{x}) = \text{sign}(f(\mathbf{x})) = \text{sign}\left(\sum_{i=1}^{\ell} \alpha_i k(\mathbf{x}_i, \mathbf{x})\right) \qquad \boldsymbol{\alpha} \in \mathbb{R}^\ell. \tag{1}$$

Here, k is referred to as a kernel and is assumed to be symmetric and positive definite. Then it is known from the theory of reproducing kernel Hilbert spaces (RKHS) [31] that there exists a *feature space* $\mathcal{F}$ and a mapping $\phi : \mathcal{X} \mapsto \mathcal{F}$ — not necessarily unique — such that f can be expressed as an inner product between the mapped point $\mathbf{x}$ and a vector $\mathbf{w} \in \mathcal{F}$, i.e.

$$f(\mathbf{x}) \quad = \quad \langle \mathbf{w}, \phi(\mathbf{x}) \rangle_{\mathcal{F}} = \sum_{i=1}^{\ell} \alpha_i \phi(\mathbf{x}_i) \qquad \mathbf{w} \in \mathcal{F}, \boldsymbol{\alpha} \in \mathbb{R}^\ell. \tag{2}$$

Without loss of generality we assume in the following that $\mathcal{F}$ is the surface of a hypersphere $\|\phi(\mathbf{x})\|_{\mathcal{F}} = 1$. Suppose we are given a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell} \subset (\mathcal{X} \times \{-1, +1\})^\ell$. In a similar fashion to PAC analysis [25] we assume that there exists a function f^* such that $y_i = \text{sign}(f^*(\mathbf{x}_i))$. Then the space of consistent hypotheses — in the following referred to as the *version space* — is defined by

$$\mathcal{V}(S) = \{f \in \mathcal{H} \subseteq \mathcal{F} : y_i f(\mathbf{x}_i) > 0; \quad i = 1, \ldots, \ell\}. \tag{3}$$

In order to enforce a unique parameterisation of f in $\mathbf{w}$ (see Equation (2)) we restrict ourself to a compact bounded set $\mathcal{H} \in \mathcal{F}$ of linear classifiers in the feature space $\mathcal{F}$. In the subsequent sections we will use the set $\mathcal{H}$ given by

$$\mathcal{H} = \left\{ f(\cdot) = \langle \mathbf{w}, \phi(\cdot) \rangle_{\mathcal{F}} : \|\mathbf{w}\|_{\mathcal{F}}^2 = 1 \right\}, \tag{4}$$

which is known to have finite volume iff k fulfils the Mercer conditions [12] and $\mathcal{X}$ is restricted to a compact set (see [31]). For any probability measure $P_{\mathcal{H}}$ over $\mathcal{H}$ we define the volume $\text{vol}(A; P)$ of $A \subseteq \mathcal{H}$ to be

$$\text{vol}(A; P) = \int_A P(f)\, df < \infty, \tag{5}$$

where $\text{vol}(A)$ is understood as the volume under the uniform distribution.

Given a version space $\mathcal{V}(S)$ the main question is: which linear classifier in $\mathcal{V}(S)$ is optimal and consequently should be returned by a learning algorithm? From an empirical risk minimisation point of view every linear classifier in $\mathcal{V}(S)$ is optimal. Thus, besides the possibility of returning a randomly selected classifier out of $\mathcal{V}(S)$ — as done by the classical perceptron algorithm [18], or the Gibbs learning rule (e.g. [9]) — different approaches have been devised.

2.1 PAC Style Analysis

Bounding the complexity of a subset of classifiers from above, the VC/PAC theory of learning recommends returning the classifier $h_{\mathrm{PAC}} = \mathrm{sign}(f_{\mathrm{PAC}})$ originating from a subset of small complexity. Hence, the term complexity refers to the VC dimension, fat shattering dimension, or the margin attained on the training set (for a detailed discussion and definition of these concepts see [21, 28] or Section 4). The following theorem to be found in [21] can serve as a basis for the well known class of large margin algorithms.

Theorem 1. *Suppose inputs are drawn independently according to a distribution whose support is contained in the ball of radius R. If we succeed in correctly classifying ℓ such inputs by a hyperplane $f \in \mathcal{V}(S)$ (see Equation (3) and (4)) achieving a margin of $\gamma = \min_S (y_i f(\mathbf{x}_i))$, then with confidence $1-\epsilon$ the generalisation error will be bounded from above by*

$$\frac{2}{\ell}\left(\kappa \log_2\left(\frac{8e\ell}{\kappa}\right)\log_2(32\ell) + \log_2\left(\frac{8\ell}{\epsilon}\right)\right),$$

where $\kappa = \lfloor 577 R^2/\gamma^2 \rfloor$.

Maximising the margin γ minimises κ and thus allows algorithms to control their generalisation. The corresponding learning problem is therefore given by

$$\begin{array}{lll} \text{maximize}_{\mathbf{w}} & \min_S (y_i \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle_{\mathcal{F}}) \equiv \Omega & \\ & y_i \langle w, \phi(\mathbf{x}_i)\rangle_{\mathcal{F}} \geq \Omega > 0 & i = 1, \ldots, \ell \\ \text{s.t.} & \|\mathbf{w}\|^2_{\mathcal{F}} = 1\,. & \end{array}$$

Let us relax the unit norm constraint on $\mathbf{w}$ but instead fix

$$\min_S (y_i \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle_{\mathcal{F}}) = 1 = \frac{\Omega}{\|\mathbf{w}\|_{\mathcal{F}}}\,.$$

Then, the solution $\mathbf{w}_{\mathrm{PAC}}$ to the above problem is equivalent – up to a scaling – to the solution $\mathbf{w}_{\mathrm{SVM}}$ of the following problem

$$\begin{aligned} &\text{minimize}_{\mathbf{w}} && \|\mathbf{w}\|_{\mathcal{F}}^2 \\ &\text{s.t.} && y_i\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle_{\mathcal{F}} \geq 1 \qquad i = 1, \ldots, \ell\,. \end{aligned}$$

This optimisation problem is a QP problem and its solution $\mathbf{w}_{\mathrm{SVM}}$ corresponds to the solution found by the Support Vector Machine (SVM). Note, that $y_i\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle_{\mathcal{F}}$ can also be read as the distance of $\mathbf{w}$ from the hyperplane with normal $y_i\phi(\mathbf{x}_i)$ if $\|\phi(\mathbf{x}_i)\|_{\mathcal{F}} = 1$. Therefore SVMs can be viewed as finding the centre of the largest hypersphere inscribable in version space (see Figure 4).

2.2 Bayesian Analysis

Assuming an *a–priori* distribution $P_{\mathcal{H}}$ over the space $\mathcal{H}$ of classifiers *and* a distribution $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ over the objects and conditional classes, return that function h_{MAP} having the maximal posterior probability (MAP) or — using the posterior — the *average* decision h_{Bayes} of the linear classifiers under the posterior distribution. Note, that the latter is not necessarily contained in the original set of functions. In a similar fashion to PAC analysis let us make a Bayesian model and consider h_{MAP} as well as the average classifier h_{Bayes}. Hence, we make the following assumptions:

$$P_{\mathcal{Y}}(y|\mathbf{x}, f) \;=\; \delta(y - \mathrm{sign}(f(\mathbf{x}))\,.$$

where δ refers to the delta function and f is defined by Equation(1) and (4). This distributional assumption can be viewed as a noise free learning scenario. Then given a training set S, we have the following estimate for the posterior distribution

$$\begin{aligned} P_{\mathcal{V}(S)}(f|S) \;&\stackrel{\text{iid}}{=}\; \frac{\prod_{i=1}^{\ell} P_{\mathcal{Y}}(y_i|\mathbf{x}_i, f) P_{\mathcal{H}}(f)}{P(S)} = \begin{cases} \frac{P_{\mathcal{H}}(f)}{Z} & \text{if } f \in \mathcal{V}(S) \\ 0 & \text{otherwise} \end{cases}, \quad (6) \\ Z \;&=\; P(S) = \int_{\mathcal{H}} P_{\mathcal{Y}}(S|f') P_{\mathcal{H}}(f')\, df' = \mathrm{vol}(\mathcal{V}(S); P_{\mathcal{H}})\,. \end{aligned}$$

If we assume a uniform (flat) prior $P_{\mathcal{H}}$ it becomes apparent that the MAP estimate h_{MAP} is not unique and thus classical perceptron learning is well

justified. Let us derive the Bayes decision h_{Bayes} at point $\mathbf{x}$ using the derived posterior $P_{\mathcal{V}(S)}$:

$$\begin{aligned} f_{\text{Bayes}}\left(\mathbf{x}; P_{\mathcal{V}(S)}\right) &= \int_{\mathcal{H}} \text{sign}(f(\mathbf{x})) P_{\mathcal{V}(S)}(f|S)\, df \qquad (7) \\ &= \text{vol}\left(\mathcal{W}_{+1}(S,\mathbf{x}); P_{\mathcal{V}(S)}\right) - \text{vol}\left(\mathcal{W}_{-1}(S,\mathbf{x}); P_{\mathcal{V}(S)}\right) , \end{aligned}$$

where

$$\mathcal{W}_y(S,\mathbf{x}) = \{f \in \mathcal{V}(S) : \text{sign}(f(\mathbf{x})) = y\} .$$

We see that h_{Bayes} decides at a point $\mathbf{x}$ for the class y whose consistent functions $\mathcal{W}_y(S,\mathbf{x})$ occupy the larger volume under the posterior. Unfortunately, there is in general no unique function $f \in \mathcal{H}$ which implements f_{Bayes}. Given complete knowledge of $P_{\mathcal{X}}$ we can single out the classifier $f_{\text{bp}}$with the smallest distance to f_{Bayes} in the L_2–metric, i.e.

$$f_{\text{bp}} = \text{argmin}_{f\in\mathcal{H}} \int \left(f_{\text{Bayes}}\left(\mathbf{x}; P_{\mathcal{V}(S)}\right) - f_{\text{bp}}(\mathbf{x})\right)^2 P_{\mathcal{X}}(\mathbf{x})\, d\mathbf{x} .$$

This classifier is called the *Bayes point*. It was shown elsewhere [30, 15] that under very mild conditions regarding $P_{\mathcal{X}}$ in high dimensional spaces the following approximation f_{cm} converges at a fast rate to f_{bp}:

$$\begin{aligned} f_{\text{cm}}(\mathbf{x}) &= \int_{\mathcal{H}} f(\mathbf{x})\, P_{\mathcal{V}(S)}(f|S)\, df = \langle \mathbf{w}_{\text{cm}}, \phi(\mathbf{x})\rangle_{\mathcal{F}} , \qquad (8) \\ \mathbf{w}_{\text{cm}} &= \int_{\mathcal{H}} \mathbf{v}\, P_{\mathcal{V}(S)}(\mathbf{v}|S)\, d\mathbf{v} . \qquad (9) \end{aligned}$$

Note that $f_{\text{cm}} \in \mathcal{H}$. The hyperplane $\mathbf{w}_{\text{cm}}$ — which is merely the centre of mass of $\mathcal{V}(S)$ — is also called the *optimal perceptron* [30]. In the following section we will present the BPM algorithm which is expected to return the centre of mass of $\mathcal{V}(S)$ under the assumption of a uniform distribution $P_{\mathcal{H}}$. Note, that in this case $P_{\mathcal{V}(S)}$ is also a uniform distribution over the version space. Finally note that there is no distribution independent rule to find the Bayes point f_{bp} solely on the basis of the training set S.

3 Estimating the Bayes Point in Kernel Space

We now outline an algorithm for approximating the Bayes point by the centre of mass assuming a uniform prior $P_{\mathcal{H}}$ (the whole pseudo code is given

on page 39). The approach develops a method presented by Pal Ruján [19]: in order to obtain the centre of mass of $\mathcal{V}(S)$ we randomly and uniformly generate points (hyperplanes in input space) and average over them. Since it is difficult to generate hyperplanes consistent with S we average over the trajectory of a ball which is placed inside $\mathcal{V}(S)$ and bounced like a billiard ball. The boundaries constraining the billiard are given by the hyperplanes with normal vectors $y_i\phi(\mathbf{x}_i)$. This process converges to the centre of mass under the assumption of *ergodicity* with respect to the uniform distribution in $\mathcal{V}(S)$ [4].

Based on the fact that we play billiards in $\mathcal{V}(S)$ each position $\mathbf{b}$ of the ball, direction vector $\mathbf{v}$, and estimate $\mathbf{w}_n$ of the centre of mass of $\mathcal{V}(S)$ can be expressed as a linear combination of the mapped input points, i.e.

$$\mathbf{w}_n = \sum_{i=1}^{\ell} \alpha_i \phi(\mathbf{x}_i)\,, \quad \mathbf{b} = \sum_{i=1}^{\ell} \gamma_i \phi(\mathbf{x}_i)\,, \quad \mathbf{v} = \sum_{i=1}^{\ell} \beta_i \phi(\mathbf{x}_i)\,, \quad \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^{\ell}\,.$$

Using this notation inner products and norms in $\mathcal{F}$ become, e.g.

$$\langle \mathbf{b}, \mathbf{v} \rangle_{\mathcal{F}} = \sum_{i,j=1}^{\ell} \beta_i \gamma_j k(\mathbf{x}_i, \mathbf{x}_j)\,, \qquad \|\mathbf{b}\|_{\mathcal{F}}^2 = \sum_{i,j=1}^{\ell} \gamma_i \gamma_j k(\mathbf{x}_i, \mathbf{x}_j)\,. \tag{10}$$

At the beginning we assume that $\mathbf{w}_0 = \mathbf{0} \Leftrightarrow \boldsymbol{\alpha} = \mathbf{0}$.

Before generating a billiard trajectory in version space we first run any kernel perceptron learning algorithm to find an initial starting point $\mathbf{b}_0$ inside the version space (e.g. SVM [27]). Then the algorithm consists of three steps:

1. Determine the closest boundary starting from the current position $\mathbf{b}$ into direction $\mathbf{v}$.

   Since it is computationally very demanding to calculate the flight time of the ball *on* geodesics of the hypersphere $\mathcal{H}$ (see also [14]) we make use of the fact that the shortest distance in Euclidean space (if it exists) is also the shortest distance on the hypersphere $\mathcal{H}$. Thus, we have for the flight time τ_j of the ball at position $\mathbf{b}$ in direction $\mathbf{v}$ to the hyperplane with normal vector $y_j\phi(\mathbf{x}_j)$ (for further details see Appendix A)

$$\tau_j = \frac{d_j}{\nu_j} \stackrel{\text{def}}{=} \frac{\langle \mathbf{b}, \phi(\mathbf{x}_j) \rangle_{\mathcal{F}}}{\langle \mathbf{v}, \phi(\mathbf{x}_j) \rangle_{\mathcal{F}}}\,. \tag{11}$$

After computing all ℓ flight times, we look for the smallest positive, i.e.

$$m = \operatorname{argmin}_{j:\tau_j>0} \tau_j \, .$$

Computing the closest bounding hyperplane in Euclidean space rather than on geodesics causes problems if the curvature of the hypersphere $\mathcal{H}$ is almost orthogonal to the direction vector $\mathbf{v}$, in which case $\tau_m \to \infty$. If this happens we randomly generate a direction vector $\mathbf{v}$ pointing *towards* the version space. Assuming that the last bounce took place at the hyperplane having normal $y_m \phi(\mathbf{x}_m)$ this condition can easily be checked by

$$y_m \langle \mathbf{v}, \phi(\mathbf{x}_m) \rangle_{\mathcal{F}} > 0 \, . \tag{12}$$

2. Update the ball's position to $\mathbf{b}'$ and the new direction vector to $\mathbf{v}'$.

   The new point $\mathbf{b}'$ and the new direction $\mathbf{v}'$ are calculated from (see Appendix A)

$$\mathbf{b}' = \mathbf{b} + \tau_m \mathbf{v} = \sum_{i=1}^{\ell} (\gamma_i + \tau_m \beta_i) \phi(\mathbf{x}_i) \, , \tag{13}$$

$$\mathbf{v}' = \mathbf{v} - 2\nu_m \phi(\mathbf{x}_m) = \sum_{i=1}^{\ell} (\beta_i - 2\delta_{im}\nu_i) \, \phi(\mathbf{x}_i) \, . \tag{14}$$

   Afterwards the position $\mathbf{b}'$ and the direction vector $\mathbf{v}'$ need to be normalised. This can easily be achieved using Equation (10).

3. Update the centre of mass $\mathbf{w}_n$ of the whole trajectory by the new line segment from $\mathbf{b}$ to $\mathbf{b}'$ calculated on the hypersphere $\mathcal{H}$.

   Since the solution $\mathbf{w}_\infty$ lies on the hypersphere $\mathcal{H}$ we cannot simply update the centre of mass using a weighted vector addition. Let us introduce the operation $\oplus_\mu$ acting on vectors of unit length. This function has to have the following properties

$$\begin{aligned} \|\mathbf{s} \oplus_\mu \mathbf{t}\|_{\mathcal{F}}^2 &= 1 \, , \\ \|\mathbf{t} - \mathbf{s} \oplus_\mu \mathbf{t}\|_{\mathcal{F}} &= \mu \, \|\mathbf{t} - \mathbf{s}\|_{\mathcal{F}} \, , \\ \mathbf{s} \oplus_\mu \mathbf{t} &= \rho_1(\mathbf{s}, \mathbf{t}, \mu)\mathbf{s} + \rho_2(\mathbf{s}, \mathbf{t}, \mu)\mathbf{t} \, , \\ \rho_1(\mathbf{s}, \mathbf{t}, \mu) \geq 0 \quad &, \quad \rho_2(\mathbf{s}, \mathbf{t}, \mu) \geq 0 \, . \end{aligned}$$

This rather arcane definition implements a weighted addition of $\mathbf{s}$ and $\mathbf{t}$ such that μ is the fraction between the resulting chord length $\|\mathbf{t} - \mathbf{s} \oplus_\mu \mathbf{t}\|_{\mathcal{F}}$ and the total chord length $\|\mathbf{t} - \mathbf{s}\|_{\mathcal{F}}$. A few lines of algebra (see Appendix A) then give the following formulae for $\rho_1(\mathbf{s}, \mathbf{t}, \mu)$ and $\rho_2(\mathbf{s}, \mathbf{t}, \mu)$

$$\begin{aligned} \rho_1(\mathbf{s}, \mathbf{t}, \mu) &= \mu \sqrt{-\frac{\mu^2 - \mu^2 \langle \mathbf{s}, \mathbf{t} \rangle_{\mathcal{F}} - 2}{\langle \mathbf{s}, \mathbf{t} \rangle_{\mathcal{F}} + 1}}\,, \\ \rho_2(\mathbf{s}, \mathbf{t}, \mu) &= -\rho_1(\mathbf{s}, \mathbf{t}, \mu) \langle \mathbf{s}, \mathbf{t} \rangle_{\mathcal{F}} \pm [\mu^2 (1 - \langle \mathbf{s}, \mathbf{t} \rangle_{\mathcal{F}}) - 1]\,. \end{aligned}$$

By assuming a constant line density on the manifold $\mathcal{V}(S)$ the whole line between $\mathbf{b}$ and $\mathbf{b}'$ can be represented by the midpoint $\mathbf{m}$ on the manifold $\mathcal{V}(S)$ given by

$$\mathbf{m} = \frac{\mathbf{b} + \mathbf{b}'}{\|\mathbf{b} + \mathbf{b}'\|_{\mathcal{F}}}\,.$$

Thus, one updates the centre of mass of the trajectory by

$$\mathbf{w}_{n+1} = \rho_1 \left(\mathbf{w}_n, \mathbf{m}, \frac{\Lambda_n}{\Lambda_n + \lambda_n} \right) \mathbf{w}_n + \rho_2 \left(\mathbf{w}_n, \mathbf{m}, \frac{\Lambda_n}{\Lambda_n + \lambda_n} \right) \mathbf{m}\,,$$

where $\lambda_n = \|\mathbf{b}_n - \mathbf{b}'_n\|_{\mathcal{F}}$ is the length of the trajectory in the n–th step and $\Lambda_n = \sum_{i=1}^{n} \lambda_i$ for the accumulated length up to the n–th step.

As a stopping criterion we suggest computing an upper bound on ρ_2, the weighting factor of the new part of the trajectory. If this value falls below a pre-specified threshold (TOL) we stop the algorithm. Note that the increase in Λ_n will always lead to termination.

4 Bounds on the Generalisation Error

In this section we start by deriving bounds on the generalisation error of the Bayesian classifier h_{Bayes} given by Equation (7), i.e. the classifier which decides at each point for the class whose functions occupy the larger volume under the posterior distribution $P_{\mathcal{V}(S)}$. A similar study was done in [6, 23, 9]. While the result presented in [6] is similar to ours in considering the ratio of the volume of parameter space to version space, we improve their bound exponentially due to a fundamentally different reasoning. The luckiness based results of [23] are limited to a set $\mathcal{H}$ of classifiers with low dimensionality. Furthermore, only the spherical approximation is taken care of. Their luckiness approach, however, is similar in spirit to our considerations. Lastly,

the analysis of [9] aims at deriving upper bounds on the expected loss over the prior $P_{\mathcal{H}}$ *and* $P_{\mathcal{D}}$. While such a result is of use if the prior is known to be correct, their worst case bounds are essentially as bad as classical VC bounds. Let us start with some definitions and background results.

Definition 1. Let $f \in \mathcal{H}$ be a real valued function. Let S be a set of ℓ points $(\mathbf{x}_i, y_i)$ drawn randomly according to $P_{\mathcal{D}} = P_{\mathcal{Y}} P_{\mathcal{X}}$. Then the quantity

$$R_{\text{emp}}(f;S) = \frac{1}{\ell} |\{(\mathbf{x}_i, y_i) \in S : \text{sign}(f(\mathbf{x}_i)) \neq y_i\}|$$

is defined as the *training error* of f on S. Furthermore, we define

$$R(f;P_{\mathcal{D}}) = P_{\mathcal{D}}\{(\mathbf{x},y) : \text{sign}(f(\mathbf{x}_i)) \neq y_i\}$$

as the *generalisation error* of f w.r.t. the distribution $P_{\mathcal{D}}$.

The ultimate goal of a learning algorithm is the minimisation of the generalisation error based on the iid sample S, i.e. based on the training error $R_{\text{emp}}(f;S)$ accessible during learning.Since the learning task is usually viewed as selecting a function f_{emp} from a given set of functions $\mathcal{H}$, distribution independent bounds are inherently connected with results about the uniform convergence of means to expectation values[1] (see [29, 26, 28, 1]). Hence, all these bounds involve a complexity measure of the set of functions $\mathcal{H}$ known as the VC dimension. It was shown elsewhere [**?**] that the minimal real valued output used in a thresholded classification provides a scale–sensitive VC dimension of $\mathcal{H}$ (from which the classification function was chosen). Moreover, this real–valued complexity measure allows algorithms to minimise an upper bound on the generalisation error (e.g. SVMs [21], Adaboost [20], Weight decay [2]).

In order to derive upper bounds on the generalisation error of a Bayesian classifier we note that the final prediction is solely based on the posterior distribution $P_{\mathcal{V}(S)}$ (given a set of functions $\mathcal{H}$) and the Bayesian decision rule. Hence, the task of learning[2] cannot be viewed as selecting a certain member f_{emp} from $\mathcal{H}$ but from the space $\mathcal{H}^{\mathcal{X}}$of posteriors over $\mathcal{H}$. Since the

[1]Given an unknown probability $P_{\mathcal{D}}$ the (infinite) set of expectations is given by $\{R(f;P_{\mathcal{D}}) : f \in \mathcal{H}\}$.

[2]Since *without* any test point $\mathbf{x}$, the Bayesian inference method does not return any classifier it is questionable if one can speak about *learning*. One potential idea to use the posterior is to calculate the posterior probability of certain labelings of a test set. This inference approach is also known as *transduction*.

latter is usually too complex, classical PAC style analysis fails to explain the excellent generalisation behaviour of Bayesian classifiers.

To set the stage for our result let us introduce the average generalisation error of classifiers.

Definition 2. Let $\mathcal{H}$ be a measurable space of real valued functions. Let $P_{\mathcal{Q}}$ denote a probability measure on $\mathcal{H}$. Let S be a set of ℓ points $(\mathbf{x}_i, y_i)$ drawn randomly according to $P_{\mathcal{D}}$. Then the quantity

$$R_{\text{emp}}(P_{\mathcal{Q}}; S) = \int_{\mathcal{H}} R_{\text{emp}}(f; S) P_{\mathcal{Q}}(f)\, df$$

is defined as the $P_{\mathcal{Q}}$*–average training error* on S. Furthermore, we define

$$R(P_{\mathcal{Q}}; P_{\mathcal{D}}) = \int_{\mathcal{H}} R(f; P_{\mathcal{D}}) P_{\mathcal{Q}}(f)\, df$$

as the $P_{\mathcal{Q}}$*–average generalisation error* of f.

The following theorem due to McAllester[11] serves as the basis for our analysis.

Theorem 2. *For any probability measure $P_{\mathcal{H}}$ over the space $\mathcal{H}$ of classifiers, for any probability measure $P_{\mathcal{D}}$ over the input space, with probability at least $1-\epsilon$ over the selection of the sample S of size ℓ we have the following for all measurable subsets $\mathcal{Q} \subseteq \mathcal{V}(S)$*

$$R(P_{\mathcal{Q}}; P_{\mathcal{D}}) \leq \frac{\ln \frac{1}{\text{vol}(\mathcal{Q}; P_{\mathcal{H}})} + \ln \frac{1}{\epsilon} + 2 \ln \ell + 1}{\ell}, \tag{15}$$

$$P_{\mathcal{Q}}(f) = \begin{cases} \frac{P_{\mathcal{H}}(f)}{\text{vol}(\mathcal{Q}; P_{\mathcal{H}})} & f \in \mathcal{Q} \\ 0 & \text{else} \end{cases}. \tag{16}$$

We are prepared to give our main result.

Theorem 3. *For any probability measure $P_{\mathcal{H}}$ over the space $\mathcal{H}$ of classifiers, for any probability measure $P_{\mathcal{D}}$ over the input space with probability at least $1-\epsilon$ over the selection of the sample S of size ℓ, for any measurable subset $\mathcal{Q} \subseteq \mathcal{V}(S)$ the generalisation error of the Bayesian classifier $f_{\text{Bayes}}(\cdot; P_{\mathcal{Q}})$ whose posterior is given by Equation (16) is bounded from above by*

$$R\left(f_{\text{Bayes}}(\cdot; P_{\mathcal{Q}}); P_{\mathcal{D}}\right) \leq \frac{2}{\ell}\left(\ln \frac{1}{\text{vol}\left(\mathcal{Q}; P_{\mathcal{H}}\right)} + \ln \frac{1}{\epsilon} + 2 \ln \ell + 1\right).$$

Proof. In order to prove the theorem we show that for any probability $P_{\mathcal{D}}$ and $P_{\mathcal{Q}}$, $R\left(f_{\text{Bayes}}\left(\cdot;P_{\mathcal{Q}}\right);P_{\mathcal{D}}\right) \leq 2R(P_{\mathcal{Q}};P_{\mathcal{D}})$. The result follows directly from Theorem 2.

Using Fubinis theorem (see e.g. [7]) let us rewrite $R(P_{\mathcal{Q}};P_{\mathcal{D}})$ by

$$\begin{aligned} R(P_{\mathcal{Q}};P_{\mathcal{D}}) &= \int_{\mathcal{H}} \left(\int_{\mathcal{D}} L(\text{sign}(f(\mathbf{x})),y) P_{\mathcal{D}}(\mathbf{x},y)\, d\mathbf{x}\, dy \right) P_{\mathcal{Q}}(f)\, df \\ &= \int_{\mathcal{D}} \left(\int_{\mathcal{H}} L(\text{sign}(f(\mathbf{x})),y) P_{\mathcal{Q}}(f)\, df \right) P_{\mathcal{D}}(\mathbf{x},y)\, d\mathbf{x}\, dy\,, \end{aligned}$$

where the function L in the inner integral captures the 0–1 loss of f at point $(\mathbf{x},y)$ and is defined by

$$L(\hat{y},y) = \begin{cases} 1 & \hat{y} \neq y \\ 0 & \hat{y} = y \end{cases} .$$

At each $(\mathbf{x},y)$ the set $\mathcal{H}$ boils down to two disjunctive sets $\mathcal{H}_0(\mathbf{x},y) = \{f : L(\text{sign}(f(\mathbf{x})),y) = 0\}$ and $\mathcal{H}_1(\mathbf{x},y) = \{f : L(\text{sign}(f(\mathbf{x})),y) = 1\}$. Hence, the $P_{\mathcal{Q}}$–average generalisation error is given by

$$R(P_{\mathcal{Q}};P_{\mathcal{D}}) = \int_{\mathcal{D}} \text{vol}(\mathcal{H}_1(\mathbf{x},y);P_{\mathcal{Q}}) P_{\mathcal{D}}(\mathbf{x},y)\, d\mathbf{x}\, dy\,.$$

Similarly, dividing the set of points into $\mathcal{D}_0 = \{(\mathbf{x},y) : L(\text{sign}(f_{\text{Bayes}}(\mathbf{x};\mathcal{P}_{\mathcal{Q}})),y) = 0\}$ and $\mathcal{D}_1 = \{(\mathbf{x},y) : L(\text{sign}(f_{\text{Bayes}}(\mathbf{x};\mathcal{P}_{\mathcal{Q}})),y) = 1\}$ the generalisation error of $f_{\text{Bayes}}\left(;\mathcal{P}_{\mathcal{Q}}\right)$ can be written as

$$R\left(f_{\text{Bayes}}\left(\cdot;\mathcal{P}_{\mathcal{Q}}\right);P_{\mathcal{D}}\right) = \int_{\mathcal{D}_1} P_{\mathcal{D}}(\mathbf{x},y)\, d\mathbf{x}\, dy\,.$$

If $(\mathbf{x},y) \in \mathcal{D}_1$, it follows that $\forall f \in \mathcal{H}_1(\mathbf{x},y) : f_{\text{Bayes}}(\mathbf{x};\mathcal{P}_{\mathcal{Q}}) = f(\mathbf{x})$. By definition, at any point $\mathbf{x}$ the Bayesian classifier $f_{\text{Bayes}}(\cdot;\mathcal{P}_{\mathcal{Q}})$ gives the same output as the functions occupying the larger volume (under the posterior $P_{\mathcal{Q}}$). Therefore, $\text{vol}(\mathcal{H}_1(\mathbf{x},y);P_{\mathcal{Q}}) \geq \frac{1}{2}$, or equivalently

$$2\text{vol}(\mathcal{H}_1(\mathbf{x},y);P_{\mathcal{Q}}) \geq 1 = L(\text{sign}(f_{\text{Bayes}}(\mathbf{x};\mathcal{P}_{\mathcal{Q}})),y)$$

This proves $R(f_{\text{Bayes}}(\cdot;\mathcal{P}_{\mathcal{Q}});P_{\mathcal{D}}) \leq 2R(P_{\mathcal{Q}};P_{\mathcal{D}})$. □

In contrast to the known results from PAC/VC theory (where the r.h.s. can be evaluated before learning) the above bound is to be evaluated after training. The result is essentially data–dependent and in the spirit of the luckiness

framework: It measures how well aligned the true (unknown) distribution $P_{\mathcal{D}}$ is with the assumed prior on dependencies $P_{\mathcal{C}}$. Note that for a fixed loss L each input distribution $P_{\mathcal{D}}$ singles out an optimal decision function f. The main difference to the luckiness results given in [21] is how classifiers are characterised, i.e. in the above result the effective decrease of complexity of the Bayesian classifier results from the fact that it summarises the classifications of all classifiers $f \in \mathcal{V}(S)$. In contrast, the luckiness framework treats each classifier separately and — as a consequence — has to consider worst–case scenarios for single classifiers. In the following we will study the value of the bound for the classical Bayesian decision. Furthermore, we will present an application of that bound to linear SVMs which leads to an exponential improvement over previous bounds. At the end of this section we give a bound for the classifier estimated by the billiard algorithm.

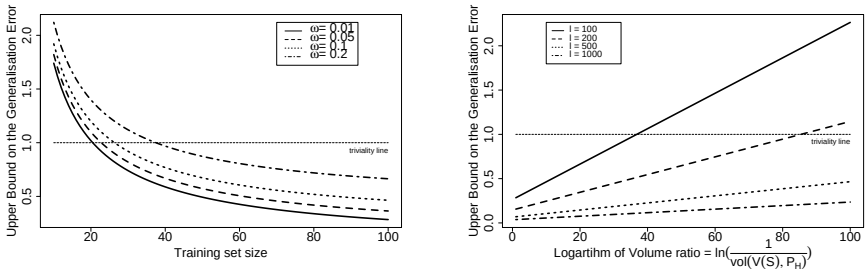
4.1 Applications to Classical Bayesian Classifiers

According to Equation (6) we see that for a PAC likelihood the posterior $P_{\mathcal{V}(S)}$ fulfils the assumptions of Theorem 3 for *any* sample S from any distribution $P_{\mathcal{D}}$. Noticing that $\mathrm{vol}(\mathcal{V}(S); P_{\mathcal{H}})$ is always less than or equal to one we see that the bound is minimized by choosing $\mathcal{Q} = \mathcal{V}(S)$. This gives the following corollary.

Corollary 1. *For any prior $P_{\mathcal{H}}$ over the space $\mathcal{H}$ of classifiers, for any probability measure $P_{\mathcal{D}}$ over the input space with probability at least $1 - \epsilon$ over the selection of the sample S of size ℓ the generalisation error of the (classical) Bayesian classifier $f_{\mathrm{Bayes}}\left(\cdot; \mathcal{P}_{\mathcal{V}(S)}\right)$ is bounded from above by*

$$\frac{2}{\ell}\left(\ln\left(\frac{1}{\mathrm{vol}(\mathcal{V}(S); P_{\mathcal{H}})}\right) + \ln\frac{1}{\epsilon} + 2\ln\ell + 1\right). \tag{17}$$

This result is very powerful because it relates the prior assumption $P_{\mathcal{H}}$ on the functions to the unknown probability distribution $P_{\mathcal{D}}$ underlying the data. In fact, if $\mathrm{vol}(\mathcal{V}(S); P_{\mathcal{H}}) \leq 2^{-\ell}$ the bound is trivial, i.e. greater than one. This does not imply that $f_{\mathrm{Bayes}}\left(\cdot; \mathcal{P}_{\mathcal{V}(S)}\right)$ has a high generalisation error, but only that we are unable to give any guarantee on the generalisation error. If the volume of version space under the posterior is significantly larger than $2^{-\ell}$ the theorem gives tight bounds. Assuming that $\mathrm{vol}(\mathcal{V}(S); P_{\mathcal{H}}) > \omega^{-1}2^{-\ell}$ we plotted the value of the bound for varying values of ω versus increasing training set size in Figure 1 (a). Interestingly, even for small training sets (of size less than 100) we get nontrivial guarantees on the generalisation error of



(a) (b)

Figure 1: **(a)** Upper bound on the generalisation error of the Bayesian classifier $f_{\text{Bayes}}\left(\cdot; P_{\mathcal{V}(S)}\right)$ (see Equation (17)) using a posterior $\text{vol}(\mathcal{V}(S); P_{\mathcal{H}}) > \omega^{-1} 2^{-\ell}$ versus training set size ℓ for $\epsilon = 0.05$. The dotted line shows the "triviality line", i.e. all values above that line are trivial bounds. **(b)** Upper bound versus the *effective* VC dimension $\ln\left(\frac{1}{\text{vol}(\mathcal{V}(S); P_{\mathcal{H}})}\right)$. Clearly, the bound scales linearly in this quantity.

$f_{\text{Bayes}}(\cdot; P_{\mathcal{V}(S)})$. Note, that $\text{vol}(\mathcal{V}(S); P_{\mathcal{H}}) = 100 \cdot 2^{-\ell}$ ($\omega = 0.01$) for $\ell = 100$ is a posterior probability of $7.8 \cdot 10^{-29}$. Even for such a small probability our result gives a guarantee for less than 70% generalisation error.

In Figure 1 (b) we plotted the upper bound on the generalisation error versus $\ln\left(\frac{1}{\text{vol}(\mathcal{V}(S); P_{\mathcal{H}})}\right)$ for varying training set sizes. Apart from the fact that this measure can be viewed as an *effective* VC–dimension we see that for $\ell = 1000$ the value of the bound is consistently less than 0.5. This can be made use of for the purpose of model selection. Nonetheless, we would like to remark that it is difficult in general to estimate the volume of version space accurately. Curiously, it seems possible to improve the bound by adjusting the prior probability. Note that it is assumed to fix $P_{\mathcal{H}}$ *before* the training data arrives. Hence, if we have knowledge which functions suit the problem at hand well (expressed by the unknown $P_{\mathcal{D}}$) we are able to bias our confidence. If this expectation fails, i.e. if we have chosen an incorrect prior $P_{\mathcal{H}}$ then the bound will report. This is clearly an advantage over classical guarantees on Bayesian classifiers.

4.2 Applications to Support Vector Machines

In order to make use of the result for SVMs we denote the largest inscribable ball in version space by $\mathcal{B}(S) \subseteq \mathcal{V}(S)$. If we now define a posterior

distribution P_{SVM} according to

$$P_{\mathrm{SVM}}(f) \quad = \quad \begin{cases} \frac{P_{\mathcal{H}}(f)}{\mathrm{vol}(\mathcal{B}(S);P_{\mathcal{H}})} & f \in \mathcal{B}(S) \\ 0 & \text{otherwise} \end{cases} ,$$

we see that P_{SVM} fulfils the assumptions of Theorem 3 for all training sets S and all distributions $P_{\mathcal{D}}$. Furthermore, $f_{\mathrm{Bayes}}\left(\cdot; P_{\mathcal{B}(S)}\right)$, the Bayesian classifier over P_{SVM} (see Equation (7)) coincides with $\mathbf{w}_{\mathrm{SVM}}$ due to the fact that $\mathcal{B}(S)$ ball being a ball is pointsymmetric w.r.t. its centre.

A linear SVM bound Let us assume that our data space $\mathcal{X}$ is $\mathbb{R}^n$ endowed with the kernel $k(\mathbf{x},\mathbf{x}') = \sum_{i=1}^n x_i x_i' = \langle \mathbf{x},\mathbf{x}' \rangle_{\mathbb{R}^n}$. Furthermore, we assume that for all $\mathbf{x}$ the kernel satisfies $k(\mathbf{x},\mathbf{x}) = 1$, i.e the data lives on the unit hypersphere in $\mathbb{R}^n$. Then, given the margin γ we know that for a uniform prior $P_{\mathcal{H}}$ (see Appendix B.1)

$$\ln\left(\frac{\mathrm{vol}(\mathcal{H})}{\mathrm{vol}(\mathcal{B}(S))}\right) \leq n \ln\left(\frac{4}{\gamma^2}\right) .$$

This gives the following bound on the generalisation error for linear SVM classifiers.

Corollary 2. *Suppose inputs are drawn independently according to a distribution whose support is contained on the sphere of radius one in $\mathbb{R}^n$. If we succeed in correctly classifying ℓ such inputs by a hyperplane $f \in \mathcal{V}(S)$ (see Equation (3) and (4)) achieving a margin of $\gamma = \min_S(y_i f(\mathbf{x}_i))$, then with confidence $1-\epsilon$ over the selection of the samples S the generalisation error will be bounded from above by*

$$\frac{2}{\ell}\left(n \ln \kappa + 2 \ln \ell + \ln \frac{1}{\epsilon} + 1\right) ,$$

where $\kappa = \frac{4}{\gamma^2}$.

In contrast to the known result of Theorem 1 we see that our bound is of order $\mathcal{O}(n \ln \kappa)$ whilst the former is of order $\mathcal{O}(\kappa \ln(\ell/\kappa) \ln(\ell))$. Also, the constants in 2 are much smaller than the constants in the classical SVM bounds. Furthermore, we see that the ratio of volumes simplifies to the well known margin complexity. Once again this demonstrates that this ratio serves as an effective VC dimension. In Figure 2 (a) we plotted the value of the upper bound for varying training set size as a function of increasing

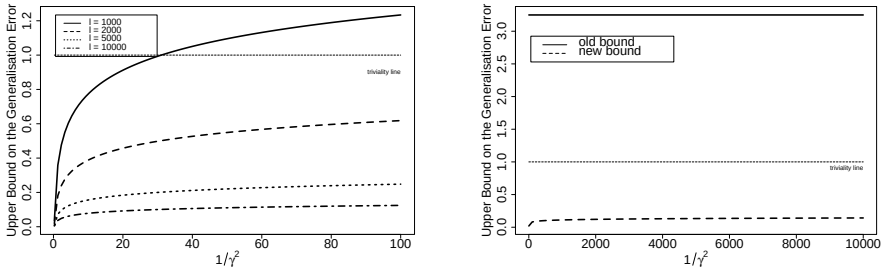


Figure 2: **(a)** Upper bound on the generalisation error (given by Corollary 2) vs. margin complexity $1/\gamma^2$ for varying training set sizes ($\epsilon = 0.05$). As can be seen these bounds grow logarithmically in $1/\gamma^2$. Note, that the corresponding upper bounds given by Theorem 1 are far above the "triviality line". **(b)** The two SVM bounds vs. $1/\gamma^2$ for the NIST task ($n = 400, \ell = 60000, \epsilon = 0.05$).

margin complexity. We chose $n = 100$ input dimensions. As can be seen from these plots, our bound becomes predictive in a large regime as soon as the training set size exhibits a ratio of $\ell/n \geq 20$. Although this heuristic was already given by Vapnik (see [27]) we would like to emphasise that the classical bound is far beyond the triviality line ($R = 1$) in this parameter regime. In Figure 2 (b) we applied our bound to one domain where SVM showed promising generalisation performance — the field of handwritten digit recognition. For the particular task of NIST zip codes we know that $n = 20 \times 20 = 400$ and $\ell = 60000$ [28]. Promisingly, even for very small margins ($\gamma \leq 0.01$) the bound in Corollary 2 could report the superiority of "large" margins while the known result is in no regime non–trivial.

4.3 Application to Bayes Point Machines

In order to apply the result given by Theorem 3 to the Bayes point $\mathbf{w}_{\mathrm{cm}}$ we have to define a region $\mathcal{R}(\mathbf{w}_{\mathrm{cm}}) \subseteq \mathcal{V}(S)$ such that the Bayesian classifier $f_{\mathrm{Bayes}}\left(\cdot; \mathcal{P}_{\mathcal{R}(\mathbf{w}_{\mathrm{cm}})}\right)$ under a uniform distribution over $\mathcal{R}(\mathbf{w}_{\mathrm{cm}})$ always agrees with $\mathbf{w}_{\mathrm{cm}}$. This is easily achieved by constructing an auxiliary *mirrored* version space $\mathcal{V}(S'(\mathbf{w}))$ w.r.t. any $\mathbf{w} \in \mathcal{V}(S)$. Formally, this space is given by the set of all hyperplanes consistent with the auxiliary training set (see Appendix A)

$$S'(\mathbf{w}) = \left\{2\left\langle \mathbf{w}, \phi(\mathbf{x}_i)\right\rangle_{\mathcal{F}} - \phi(\mathbf{x}_i), +1 : (\mathbf{x}_i, y_i) \in S\right\}.$$

The training set $S'(\mathbf{w})$ can be viewed as bounding hyperplanes $y_i\phi(\mathbf{x}_i)$ point–mirrored at point $\mathbf{w}$. The subset $\mathcal{R}(\mathbf{w})$ of version space point symmetric w.r.t. $\mathbf{w}$ is defined as the intersection of $\mathcal{V}(S)$ and $\mathcal{V}(S'(\mathbf{w}))$, i.e.

$$\mathcal{R}(\mathbf{w}) = \mathcal{V}\left(S'(\mathbf{w})\right) \cap \mathcal{V}(S) \ .$$

Note, that the intersection of two convex sets is always convex [10]. Moreover, by construction $\mathbf{w}$ agrees with the Bayesian classifier using a uniform distribution over $\mathcal{R}(\mathbf{w})$: For every data point $\phi(\mathbf{x})$ bisecting $\mathcal{R}(\mathbf{w})$, $\mathbf{w}$ lies in the half of larger volume. Consider only those $\phi(\mathbf{x})$ where[3] $\langle \mathbf{w}, \phi(\mathbf{x})\rangle_{\mathcal{F}} = 0$. Then for each classifier $\mathbf{w} \in \mathcal{W}_{+1} = \{\mathbf{w}' \in \mathcal{R}(\mathbf{w}) : \langle \mathbf{w}', \phi(\mathbf{x})\rangle_{\mathcal{F}} > 0\}$ there exists a corresponding classifier $\mathbf{w} \in \mathcal{W}_{-1} = \{\mathbf{w}' \in \mathcal{R}(\mathbf{w}) : \langle \mathbf{w}', \phi(\mathbf{x})\rangle_{\mathcal{F}} < 0\}$ by mirror–symmetry, and vice versa (see Lemma 4). Hence, the volumes $\mathrm{vol}(\mathcal{W}_{+1})$ and $\mathrm{vol}(\mathcal{W}_{-1})$ are of equal magnitude under the uniform measure over $\mathcal{R}(\mathbf{w})$.

The above argument holds for any classifier $\mathbf{w} \in \mathcal{V}(S)$ and thus allows the application of Theorem 3 to arbitrary version space members. The volume to be considered in the bound is $\mathrm{vol}(\mathcal{R}(\mathbf{w}))$. Although the centre of mass $\mathbf{w}_{\mathrm{cm}}$ does not maximise $\mathcal{R}(\mathbf{w})$ it appears that under quite general circumstances $\mathrm{vol}(\mathcal{R}(\mathbf{w}_{\mathrm{cm}})) > \mathrm{vol}(\mathcal{R}(\mathbf{w}_{\mathrm{SVM}}))$ (for an example see Figure 4). Further investigations aim at constructing the true maximiser of $\mathrm{vol}(\mathcal{R}(\mathbf{w}))$.

5 Bayes Point Estimation with Soft Boundaries

To allow for training errors we will introduce the following version space conditions in place of those in Equation (3).

$$y_j \sum_{i=1}^{\ell} \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) \geq -\lambda y_j \alpha_j k(\mathbf{x}_j, \mathbf{x}_j) \ , \tag{18}$$

where $\lambda \geq 0$ is an adjustable parameter related to the "softness" of version space boundaries.

Clearly, considering this from the billiard viewpoint Equation (18) can be interpreted as allowing penetration of the walls, an idea already hinted at in [19]. Since the decision function based on Equation (1) is invariant under any positive rescaling of the α_i a factor α_j on the right hand side makes λ scale–invariant as well. Although other ways of incorporating training

[3] For a fixed $\phi(\mathbf{x})$ the resulting $\mathbf{w}$ form a so called *Bayes line*.

errors are conceivable our formulation allows for a simple modification of the algorithm described in Section 3. To see this we note that Equation (18) can be re-written as

$$y_j \left[\sum_{i=1}^{\ell} \alpha_i (1 + \lambda \delta_{ij}) k(\mathbf{x}_i, \mathbf{x}_j) \right] \geq 0$$

Hence we can use the above algorithm but with an additive correction to the diagonal terms of the kernel matrix computed at the start of the algorithm $k(\mathbf{x}_j, \mathbf{x}_j) \leftarrow k(\mathbf{x}_j, \mathbf{x}_j) + \lambda$. This additive correction to the kernel diagonals is similar to the L_2 error norm [5] used to introduce a soft margin during training of SVMs which has recently been theoretically motivated [22]. Another insight into the introduction of soft boundaries comes from noting that the distance between two points $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ can be written

$$\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|_{\mathcal{F}}^2 \quad = \quad \|\phi(\mathbf{x}_i)\|_{\mathcal{F}}^2 + \|\phi(\mathbf{x}_j)\|_{\mathcal{F}}^2 - 2\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}} \,,$$

which in the case of soft boundaries becomes $2(1 + \lambda - k(\mathbf{x}_i, \mathbf{x}_j))$. Thus, if we add λ to the diagonal elements of the kernel matrix, the points become equidistant for $\lambda \to \infty$. This would give the resulting version space a more regular shape. As a consequence, the centre of the largest inscribable sphere (SVM solution) would tend towards the centre of mass of the whole of version space.

We want to note that our scheme of incorporating training errors allows us to bound the generalisation error using Theorem 3. Considering that the whole parameter space is given by

$$\left\{ \mathbf{w} = \sum_{i=1}^{\ell} \alpha_i \phi(\mathbf{x}_i) : \|\mathbf{w}\|_{\mathcal{F}}^2 = \sum_i \sum_j \alpha_i \alpha_j \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = 1 \right\} .$$

This can be rewritten as

$$\left\{ \boldsymbol{\alpha} \in \mathbb{R}^{\ell} : \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} = 1 \right\} \qquad \mathbf{K}_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}} = k(\mathbf{x}_i, \mathbf{x}_j) \,.$$

Let us represent the kernel matrix by its spectral decomposition, i.e. $\mathbf{K} = \mathbf{U}^T \mathbf{D} \mathbf{U}$ where $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ and $\mathbf{D} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\ell})$ being the diagonal matrix of eigenvalues σ_i. Thus we know that the parameter space is the set of all coefficients $\widetilde{\boldsymbol{\alpha}} = \mathbf{U} \boldsymbol{\alpha}$ which fulfill

$$\left\{ \widetilde{\boldsymbol{\alpha}} \in \mathbb{R}^{\ell} : \widetilde{\boldsymbol{\alpha}}^T \mathbf{D} \widetilde{\boldsymbol{\alpha}} = 1 \right\} .$$

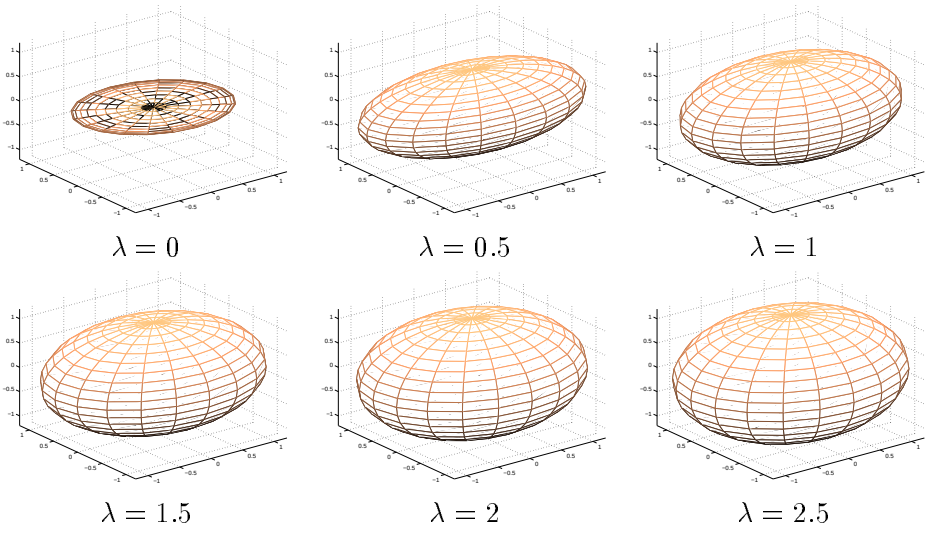


Figure 3: Parameter spaces for a 2D toy problem obtained by introducing training error via an additive correction to the diagonal term of the kernel matrix. In order to visualise the resulting parameter space we fixed $\ell = 3$ and normalised all axes by the product of eigenvalues $\sqrt{\sigma_1\sigma_2\sigma_3}$ (see text for further explanation). This gives the same scaling for all ellipsoids. Clearly, for no training errors ($\lambda = 0$) the parameter space is a 2D ellipse. By introducing an additive correction to the diagonal term of the kernel matrix $\mathbf{K}$ the parameter space expands in the third dimension and finally results in a 3D ball ($\lambda = 2.5$).

This is the defining equation of an ℓ–dimensional axis–parallel ellipsoid. Now adding the term λ to the diagonal of $\mathbf{K}$ makes $\mathbf{K}$ a full rank matrix (see [13]). In Figure 3 we plotted the parameter space for a 2D toy problem using only $\ell = 3$ training points. Although the parameter space is 3–dimensional for all $\lambda > 0$ we obtain a pancake like parameter space for small values of λ. For $\lambda \to \infty$ the set $\widetilde{\boldsymbol{\alpha}}$ of admissible coefficients becomes the ℓ–dimensional ball. As shown in Corollary 1 and 2 this gives a trivial upper bound in the limit of $\lambda \to \infty$. In the range $0 \leq \lambda \leq \infty$ the size of the resulting version space can be evaluated using the approaches outlined in Subsection 4.2 and 4.3. Hence the bound given by Theorem 3 can be applied to learning with training errors admitted. Furthermore, a similar reasoning is applicable to general Mercer kernels.

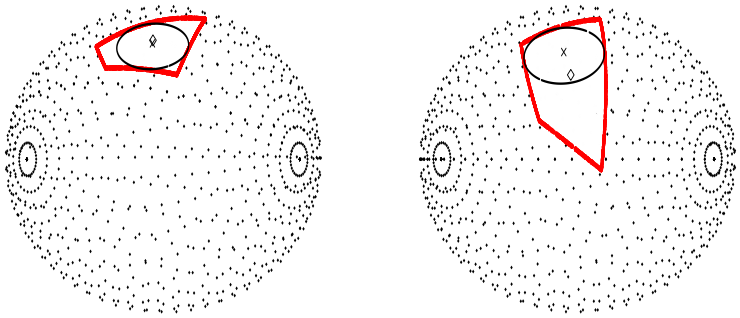


Figure 4: Version spaces $\mathcal{V}(S)$ for two 3D–toy problems. One can see that the approximation of the Bayes point (diamond) by the centre of the largest inscribable sphere (cross) is reasonable if the version space is regularly shaped (left). The situation changes in the case of an elongated and asymmetric version space $\mathcal{V}(S)$ (right).

6 Experiments

In Figures 4 we illustrate the potential benefits of a BPM over a SVM for elongated version spaces. We randomly generated two datasets with 10 training and 10000 test points in $\mathbb{R}^3$. The data points were uniformly generated in $[-1,1]^3$ and labelled by a randomly generated linear decision rule using the kernel $k(\mathbf{x},\mathbf{x}') = \langle \mathbf{x},\mathbf{x}' \rangle_{\mathbb{R}^3}$. By tracking all positions $\mathbf{b}_n$ where the billiard ball hits a version space boundary we can easily visualise the version spaces. For the example illustrated in Figure 4 (right) the SVM and Bayes point solutions with hard margins/boundaries are far apart resulting in a noticeable reduction in generalisation error of the BPM (8.0%) compared to the SVM (15.1%) solution. For another toy example involving RBF kernels $k(\mathbf{x},\mathbf{x}') = \exp\left(\|\mathbf{x}-\mathbf{x}'\|^2/2\sigma^2\right)$ Figure 5 shows the resulting decision functions in the hard margin case. Clearly, the BPM solution appears much smoother than the SVM solution although its minimal margin of 0.020 is significantly smaller.

To investigate the performance on real–world datasets we compared hard margin SVMs to BPMs with hard boundaries ($\lambda = 0$). We studied the performance on 5 standard benchmarking datasets from the UCI Repository

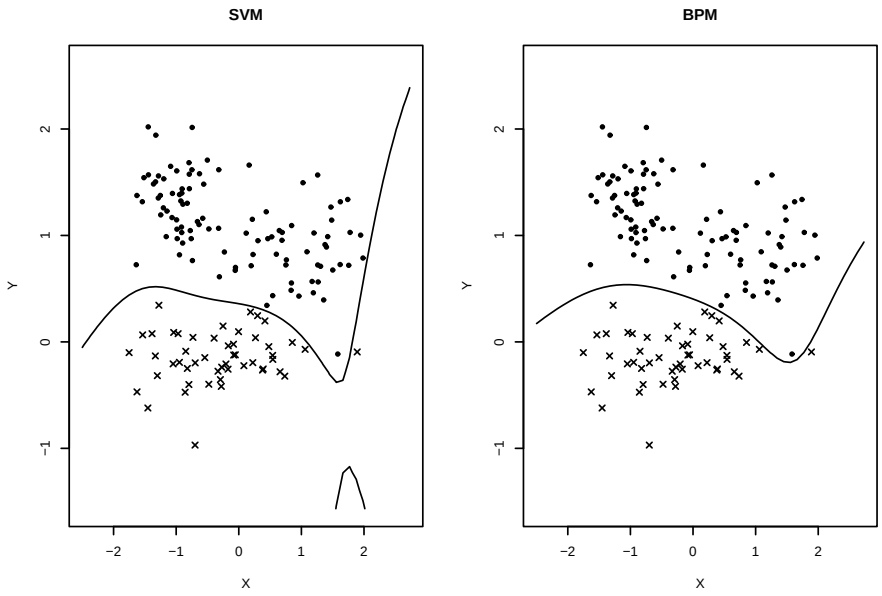


Figure 5: Decision functions for a 2D toy problem of a SVM (left) and BPM (right) using hard margins ($\lambda = 0$) and RBF kernels with the same sigma $\sigma = 1$. Note, that the BPM result in a much "flatter" function sacrificing margin ($\gamma(\mathbf{w}_{SVM}) = 0.036 \rightarrow \gamma(\mathbf{w}_{\mathrm{cm}}) = 0.020$) for smoothness.

[24], and `banana` and `waveform`, two toy datasets[4]. In each case the data was randomly partitioned into 100 training and test sets in the ratio 60%:40%. The means and standard deviations of the average generalisation errors on the test sets are presented as percentages in the columns headed SVM (hard margin) and BPM ($\lambda = 0$) in Table 1. The BPM outperforms SVMs on almost all datasets at a statistically significant level.

In order to demonstrate the effect of positive λ (soft boundaries) we trained a BPM with soft boundaries and compared it to training a SVM with soft margin using the same kernel matrix (see Equation (18)). Figure 6 shows the generalisation error as a function of λ for the toy problem from Figure 4 and the dataset `thyroid` using the same setup as in the previous experiment. We observe that the SVM with an L_2 soft margin achieves a minimum of the generalisation error which is close to, or just above, the minimum error which can be achieved using a BPM with positive λ. This may not be too surprising taking the change of geometry into account (see Section 5). Thus,

[4]Publically available at http://horn.first.gmd.de/~raetsch/data/benchmarks.htm.

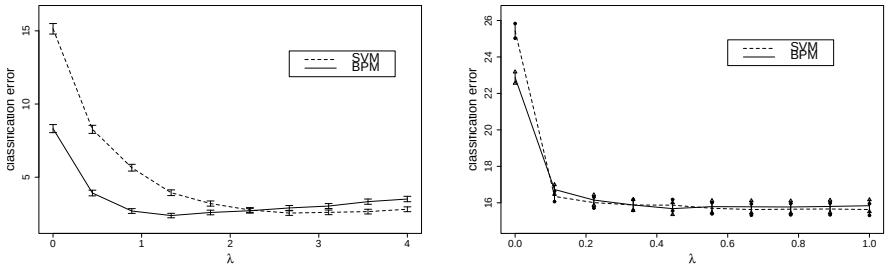


Figure 6: Comparison of soft boundary BPM with soft margin SVM. Plotted is the Generalisation error versus λ for a toy problem using linear kernels (left) and the thyroid dataset using RBF kernels with $\sigma = 3.0$ (right). The error bars indicate one standard deviation of the estimated mean.

also the soft margin SVMs approximates BPMs with soft boundaries.

	SVM (hard margin)	BPM (hard boundary)	σ	p-value
Heart	25.4±0.40	**22.8±0.34**	10.0	1.00
Thyroid	5.3±0.24	**4.4±0.21**	3.00	1.00
Diabetes	33.1±0.24	**32.0±0.25**	5.0	1.00
Waveform	13.0±0.10	**12.1±0.09**	20.0	1.00
Banana	16.2±0.15	**15.1±0.14**	0.5	1.00
Sonar	**15.4±0.37**	15.9±0.38	1.0	0.01
Ionosphere	11.9±0.25	**11.5±0.25**	1.5	0.99

Table 1: Experimental results on seven benchmark datasets. Shown is the estimated generalisation error in percent. The standard deviation was obtained on 100 different runs. The final column gives the p-values of a paired t-test for the hypothesis "*BPM is better than SVM*" indicating that the improvement is statistically significant.

7 Discussion and Conclusion

In this paper we presented an estimation method for the Bayes point for linear functions in Hilbert space. We showed how the SVM can be viewed as an (spherical) approximation method to the Bayes point hyperplane. By randomly generating consistent hyperplanes playing billiards in version space we showed how to stochastically approximate this point. In the field of Markov Chain Monte Carlo methods such approaches are known as *reflective*

slice sampling [14]. Current investigations in this field include the question of ergodicity of such methods.

We presented theoretical results which indicate that the fraction of the volume of parameter space to the volume of version space plays a crucial role in the generalisation error of Bayesian classifiers. The analysis presented exploits the idea of representing a classifier by its posterior distribution. The results motivate the centre of mass as a classifier with good volume ratio and thus good generalisation. The results also indicate that under circumstances where the shape of the version space is almost spherical the classical SVM gives the best result. All these results were supported by experiments indicating that the centre of mass has excellent generalisation behaviour.

Bayes points in kernel space constitute an interesting bridge between the Bayesian approach to machine learning and statistical learning theory. In this paper we showed that they outperform hard margin SVMs. We could also improve on further bounds by casting the SVM classifier into a Bayesian framework. However, it is well known that introduction of a soft margin improves the generalisation performance of SVMs on most datasets by allowing for training errors. Consequently we introduced a mechanism for Bayesian learning with training errors admitted. A comparison of the generalisation performance of the two types of systems shows they exhibit a much closer generalisation performance than in the hard boundary/margin case.

Although our approach has an impressive generalisation performance further work is required to improve the usability and performance. For example, it may be possible to introduce simpler algorithms for approximating the Bayes Point in kernel space [30].

Acknowledgements

This work was partially done during a research stay of Ralf Herbrich at University of Bristol and Royal Holloway University London. He would like to thank Colin Campbell and John Shawe–Taylor for the excellent research environment and also for the warm hospitality during that stay. We are also greatly indebted to Matthias Burger, Søren Fiig Jarner, Klaus Obermayer, Craig Saunders, Matthias Seeger, John Shawe–Taylor, Alex Smola, and Jason Weston for fruitful discussions. In particular we would like to thank Ulrich Kockelkorn and Peter Bollmann–Sdorra for their great support in proving Lemmata 2 and 3.

A Geometry in an RKHS

This appendix gives a detailed derivation of geometrical results in Reproducing Kernel Hilbert spaces (some of them are extensively used in the kernel billiard algorithm). Most of the results are well known from linear algebra and can be found in many textbooks, e.g. [8].

A.1 Flight times in Kernel Space

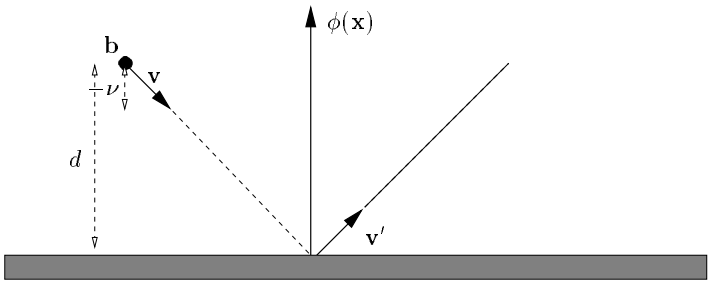


Figure 7: Bouncing the ball in an RKHS. See text for further details.

Assume $\mathbf{v}, \mathbf{b} \in \mathcal{F}$ are normalised and $\phi(\mathbf{x})$ is the normal vector (not necessarily of unit length) of the hyperplane (see Figure 7). Here, ϕ is a mapping from $\mathcal{X}$ into the RKHS $\mathcal{F}$ endowed with the reproducing kernel $k(\cdot,\cdot)$, i.e. $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}} = k(\mathbf{x}_i, \mathbf{x}_j)$. Clearly

$$\frac{\langle \mathbf{b}, \phi(\mathbf{x}) \rangle_{\mathcal{F}}}{\|\phi(\mathbf{x})\|_{\mathcal{F}}}$$

is the distance of the ball $\mathbf{b}$ from the hyperplane. Moreover, for direction vectors pointing *towards* the hyperplane,

$$\frac{\langle \mathbf{v}, \phi(\mathbf{x}) \rangle_{\mathcal{F}}}{\|\phi(\mathbf{x})\|_{\mathcal{F}}}$$

is negative and its absolute value is the distance of $\mathbf{v}$ from the hyperplane. Thus, the negative fraction τ of both terms, i.e.

$$\tau \quad = \quad -\frac{\langle \mathbf{b}, \phi(\mathbf{x}) \rangle_{\mathcal{F}}}{\langle \mathbf{v}, \phi(\mathbf{x}) \rangle_{\mathcal{F}}}$$

gives the flight time for a ball at position $\mathbf{b}$ when $\mathbf{v}$ is the direction vector pointing *towards* the hyperplane. This justifies Equation (11).

One can easily check the validity of the update rule (13)

$$\mathbf{b}' \;=\; \mathbf{b} + \tau \cdot \mathbf{v}$$

by calculating the distance of $\mathbf{b}'$ to the hyperplane with normal vector $\phi(\mathbf{x})$, i.e.

$$\begin{aligned}
\frac{\langle \mathbf{b}', \phi(\mathbf{x})\rangle_{\mathcal{F}}}{\sqrt{\langle \phi(\mathbf{x}), \phi(\mathbf{x})\rangle_{\mathcal{F}}}} &= \frac{\langle \mathbf{b} + \tau \cdot \mathbf{v}, \phi(\mathbf{x})\rangle_{\mathcal{F}}}{\|\phi(\mathbf{x})\|_{\mathcal{F}}} \\
&= \frac{\langle \mathbf{b}, \phi(\mathbf{x})\rangle_{\mathcal{F}} - \frac{\langle \mathbf{b}, \phi(\mathbf{x})\rangle_{\mathcal{F}}}{\langle \mathbf{v}, \phi(\mathbf{x})\rangle_{\mathcal{F}}} \langle \mathbf{v}, \phi(\mathbf{x})\rangle_{\mathcal{F}}}{\|\phi(\mathbf{x})\|_{\mathcal{F}}} \\
&= \frac{0}{\|\phi(\mathbf{x})\|_{\mathcal{F}}} = 0\,.
\end{aligned}$$

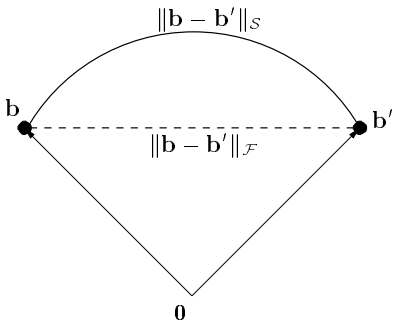


Figure 8: The relation between the flight time $\|\mathbf{b} - \mathbf{b}'\|_{\mathcal{F}}$ in the Euclidean span and the flight time $\|\mathbf{b} - \mathbf{b}'\|_{\mathcal{S}}$ on the unit hypersphere.

Let us assume that the vectors $\mathbf{b}'$ and $\mathbf{b}$ are normalised to unit length. Then we know the following relation between $\|\mathbf{b} - \mathbf{b}'\|_{\mathcal{F}}$ (the length of the shortest line between these two points in $\mathcal{F}$) and $\|\mathbf{b} - \mathbf{b}'\|_{\mathcal{S}}$ (the length of the shortest path on the unit hypersphere in $\mathcal{F}$) (see Figure 8)

$$\left\|\mathbf{b} - \mathbf{b}'\right\|_{\mathcal{S}} = \arccos\left(1 - \left(\frac{|\mathbf{b} - \mathbf{b}'|_{\mathcal{F}}}{\sqrt{2}}\right)^2\right)\,.$$

This follows from the fact that

$$\begin{aligned}
\left\|\mathbf{b}-\mathbf{b}'\right\|_{\mathcal{S}} &= \arccos(\langle \mathbf{b},\mathbf{b}'\rangle_{\mathcal{F}})\,, \\
\left\|\mathbf{b}-\mathbf{b}'\right\|_{\mathcal{F}} &= \sqrt{\langle \mathbf{b}-\mathbf{b}',\mathbf{b}-\mathbf{b}'\rangle_{\mathcal{F}}} \\
&= \sqrt{\langle \mathbf{b},\mathbf{b}\rangle_{\mathcal{F}} - 2\langle \mathbf{b},\mathbf{b}'\rangle_{\mathcal{F}} + \langle \mathbf{b}',\mathbf{b}'\rangle_{\mathcal{F}}} \\
&= \sqrt{1-2\langle \mathbf{b},\mathbf{b}'\rangle_{\mathcal{F}}+1} \\
&= \sqrt{2\left(1-\langle \mathbf{b},\mathbf{b}'\rangle_{\mathcal{F}}\right)}\,.
\end{aligned}$$

A.2 Reflections in Kernel Space

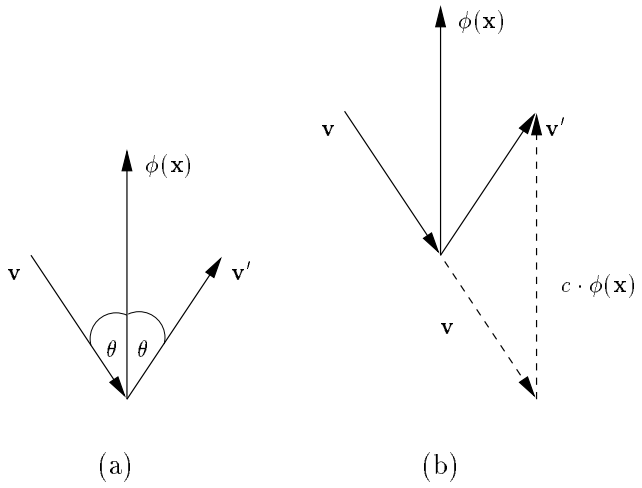


Figure 9: (**a**): A reflection of $\mathbf{v}$ at the hyperplane with normal vector $\phi(\mathbf{x})$. Note, that $\cos(\theta) = \langle \mathbf{v},\phi(\mathbf{x})\rangle_{\mathcal{F}} / \left(\|\mathbf{v}\|_{\mathcal{F}}\cdot\|\phi(\mathbf{x})\|_{\mathcal{F}}\right) = -\langle \mathbf{v}\prime,\phi(\mathbf{x})\rangle_{\mathcal{F}} / \left(\|\mathbf{v}'\|_{\mathcal{F}}\cdot\|\phi(\mathbf{x})\|_{\mathcal{F}}\right)$. (**b**): Shifting the vector $\phi(\mathbf{x})$ shows, that for the reflection vector $\mathbf{v}'$ the equality $\mathbf{v}' = \mathbf{v} + c\cdot\phi(\mathbf{x})$ has to hold. For calculation of c see the text.

Again, assume $\mathbf{v},\mathbf{v}' \in \mathcal{F}$ are normalised and $\phi(\mathbf{x})$ is the normal of the hyperplane where the reflection takes place. Then for a reflection (see Figure 9 (a)) the following equality holds,

$$\langle \mathbf{v},\phi(\mathbf{x})\rangle_{\mathcal{F}} \;=\; -\langle \mathbf{v}',\phi(\mathbf{x})\rangle_{\mathcal{F}}\,. \tag{19}$$

It is also easy to see (Figure 9 (b)) that the reflected vector $\mathbf{v}'$ has to be of the form

$$\mathbf{v}' \;=\; \mathbf{v} + c\cdot\phi(\mathbf{x})\,, \tag{20}$$

where c has to chosen such that Equation (20) is fulfilled. Thus, inserting Equation (20) into Equation (19) gives

$$\begin{aligned}\langle \mathbf{v}, \phi(\mathbf{x})\rangle_{\mathcal{F}} &= -\langle \mathbf{v} + c \cdot \phi(\mathbf{x}), \phi(\mathbf{x})\rangle_{\mathcal{F}} \\ \langle \mathbf{v}, \phi(\mathbf{x})\rangle_{\mathcal{F}} &= -\langle \mathbf{v}, \phi(\mathbf{x})\rangle_{\mathcal{F}} - c\langle \phi(\mathbf{x}), \phi(\mathbf{x})\rangle_{\mathcal{F}} \\ c &= -2\frac{\langle \mathbf{v}, \phi(\mathbf{x})\rangle_{\mathcal{F}}}{\|\phi(\mathbf{x})\|^2_{\mathcal{F}}} = -2\nu\frac{1}{k(\mathbf{x},\mathbf{x})}\,,\end{aligned}$$

which justify the usage of Equation (14). Here we used $\nu = \langle \mathbf{v}, \phi(\mathbf{x})\rangle_{\mathcal{F}}$.

A.3 Point–Mirroring in Kernel Space

Given a point $\mathbf{w}' \in \mathcal{F}$ and a hyperplane $\{\mathbf{w} : \langle \mathbf{w}, \phi(\mathbf{x})\rangle_{\mathcal{F}} = 0\}$ characterised by its normal vector $\phi(\mathbf{x})$ the task is to find the normal $\phi(\mathbf{x}')$ of the point–mirror image of the hyperplane with respect to $\mathbf{w}'$. This situation is depicted in Figure 10. Clearly, this task is equivalent to the reflection of the vector $\phi(\mathbf{x})$ at the normal to the hyperplane where $\mathbf{w}'$ lives on. Thus, we have the

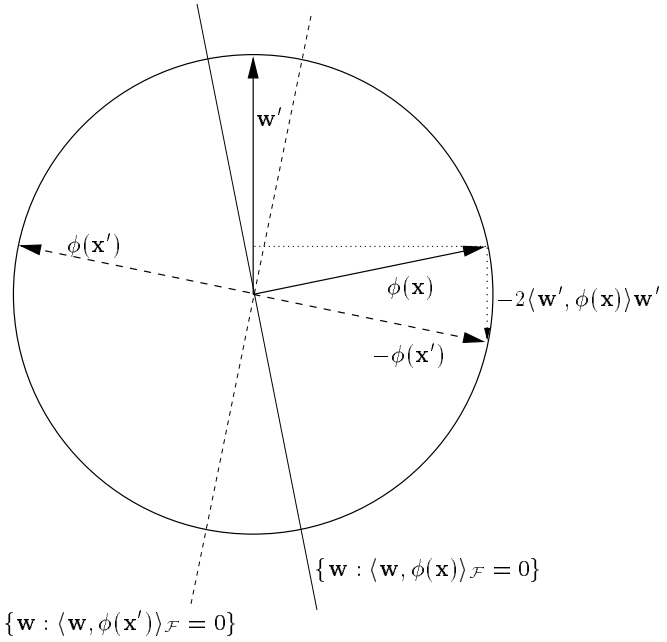


Figure 10: The reflection of the hyperplane with normal vector $\phi(\mathbf{x})$ at the point $\mathbf{w}'$. Note the close similarity to Figure 9.

following relationship for $\phi(\mathbf{x})$

$$\begin{aligned} -\phi(\mathbf{x}') &= \phi(\mathbf{x}) - 2\left\langle \mathbf{w}', \phi(\mathbf{x})\right\rangle_{\mathcal{F}} \\ \phi(\mathbf{x}') &= 2\left\langle \mathbf{w}', \phi(\mathbf{x})\right\rangle_{\mathcal{F}} - \phi(\mathbf{x}) \ . \end{aligned}$$

If $\|\mathbf{w}'\| = \|\phi(\mathbf{x})\| = 1$ we automatically obtain a normal $\phi(\mathbf{x}')$ of unit length.

$$\begin{aligned} \left\|\phi(\mathbf{x}')\right\|^2 &= \left\langle 2\left\langle \mathbf{w}', \phi(\mathbf{x})\right\rangle_{\mathcal{F}} - \phi(\mathbf{x}), 2\left\langle \mathbf{w}', \phi(\mathbf{x})\right\rangle_{\mathcal{F}} - \phi(\mathbf{x})\right\rangle_{\mathcal{F}} \\ &= 4\left(\left\langle \mathbf{w}', \phi(\mathbf{x})\right\rangle_{\mathcal{F}}\right)^2 - 4\left(\left\langle \mathbf{w}', \phi(\mathbf{x})\right\rangle_{\mathcal{F}}\right)^2 + \|\phi(\mathbf{x})\|^2 = 1 \ . \end{aligned}$$

A.4 A derivation of the operation $\oplus_\mu$

Let us derive operation $\oplus_\mu$ acting on vectors of unit length. This function has to have the following properties (see Section 3)

$$\begin{aligned} \|\mathbf{s} \oplus_\mu \mathbf{t}\|_{\mathcal{F}}^2 &= 1 \ , & (21) \\ \|\mathbf{t} - \mathbf{s} \oplus_\mu \mathbf{t}\|_{\mathcal{F}} &= \mu \|\mathbf{t} - \mathbf{s}\|_{\mathcal{F}} \ , & (22) \\ \mathbf{s} \oplus_\mu \mathbf{t} &= \rho_1 \mathbf{s} + \rho_2 \mathbf{t} \ , & (23) \\ \rho_1 \geq 0 \quad &, \quad \rho_2 \geq 0 \ . & (24) \end{aligned}$$

Here we assume that $\|\mathbf{s}\|_{\mathcal{F}}^2 = \|\mathbf{t}\|_{\mathcal{F}}^2 = 1$. Inserting Equation (23) into (21) results in

$$\begin{aligned} \|\rho_1\mathbf{s} + \rho_2\mathbf{t}\|_{\mathcal{F}}^2 &= \langle \rho_1\mathbf{s} + \rho_2\mathbf{t}, \rho_1\mathbf{s} + \rho_2\mathbf{t}\rangle_{\mathcal{F}} \\ &= \rho_1^2 \|\mathbf{s}\|_{\mathcal{F}}^2 + \rho_2^2 \|\mathbf{t}\|_{\mathcal{F}}^2 + 2\rho_1\rho_2\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}} \\ &= \rho_1^2 + \rho_2^2 + 2\rho_1\rho_2\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}} = 1 \ . \qquad (25) \end{aligned}$$

In a similar fashion combining Equation (23) and (22) gives

$$\begin{aligned} \|\mathbf{t} - \mathbf{s} \oplus_\mu \mathbf{t}\|_{\mathcal{F}}^2 &= \mu^2 \|\mathbf{t} - \mathbf{s}\|_{\mathcal{F}}^2 \\ \|(1-\rho_2)\mathbf{t} - \rho_1\mathbf{s}\|_{\mathcal{F}}^2 &= \mu^2 \|\mathbf{t} - \mathbf{s}\|_{\mathcal{F}}^2 \\ (1-\rho_2^2)\|\mathbf{t}\|_{\mathcal{F}}^2 - 2(1-\rho_2)\rho_1\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}} + \rho_1^2 \|\mathbf{s}\|_{\mathcal{F}}^2 &= \mu^2(\|\mathbf{t}\|_{\mathcal{F}}^2 - 2\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}} + \|\mathbf{s}\|_{\mathcal{F}}^2) \\ (1-\rho_2)^2 - 2(1-\rho_2)\rho_1\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}} + \rho_1^2 &= \mu^2(2 - 2\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}}) \ . \qquad (26) \end{aligned}$$

Note that Equation (25) is quadratic in ρ_2 and has the following solution

$$\rho_2 = -\rho_1\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}} \pm \sqrt{\rho_1^2(\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}})^2 - \rho_1^2 + 1} = -\rho_1\langle \mathbf{s}, \mathbf{t}\rangle_{\mathcal{F}} + A \ . \quad (27)$$

Let us insert Equation (27) into the r.h.s. of Equation (26). This gives the following quadratic equation in ρ_1

$$\begin{aligned}
(1-\rho_2)^2 - 2(1-\rho_2)\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + \rho_1^2 &= \\
1 - 2\rho_2 + \rho_2^2 - 2\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + 2\rho_2\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + \rho_1^2 &= \\
1 + 2\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + 2A + \rho_2^2 - 2\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + 2\rho_2\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + \rho_1^2 &= \\
1 + 2A + \rho_1^2 + (\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + A)^2 - 2\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}(\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + A) &= \\
1 + 2A + \rho_1^2 + \rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 + 2\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}A + A^2 & \\
-2\rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - 2\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}A &= \\
1 + 2A + \rho_1^2 - \rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 + A^2 &= \\
1 + 2A + \rho_1^2 - \rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 + \rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - \rho_1^2 + 1 &= \\
2 + 2A &= 2\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})\,.
\end{aligned}$$

Rearranging terms then gives the following

$$\begin{aligned}
1 \pm \sqrt{\rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - \rho_1^2 + 1} &= -\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}) \\
\pm\sqrt{\rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - \rho_1^2 + 1} &= \mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}) - 1 \\
\rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - \rho_1^2 + 1 &= (\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}) - 1)^2 \\
\rho_1^2((\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - 1) &= (\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}) - 1)^2 - 1 \\
\rho_1^2 &= \frac{(\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}) - 1)^2 - 1}{(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - 1} \qquad (28) \\
&= \frac{\mu^4(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - 2\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})}{(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - 1} \\
&= \frac{\mu^2[\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - 2 + 2\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}]}{(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} - 1)(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + 1)}\,.
\end{aligned}$$

Making use of the identity

$$\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - 2 + 2\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} = -(\mu^2 - \mu^2\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} - 2)(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} - 1)\,,$$

finally gives the desired result

$$\begin{aligned}
\rho_1^2 &= \frac{-\mu^2(\mu^2 - \mu^2\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} - 2)(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} - 1)}{(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} - 1)(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + 1)} \\
&= \frac{\mu^2(\mu^2 - \mu^2\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} - 2)}{\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + 1} \\
\rho_1 &= \sqrt{-\frac{\mu^2 - \mu^2\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} - 2}{\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} + 1}}\mu\,. \qquad (29)
\end{aligned}$$

Inserting this formula back into Equation (27) and making use of the identity (28) we obtain for ρ_2

$$\begin{aligned}
\rho_2 &= -\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} \pm \sqrt{\rho_1^2(\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - \rho_1^2 + 1} \\
&= -\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} \pm \sqrt{\rho_1^2((\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}})^2 - 1) + 1} \\
&= -\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} \pm \sqrt{(\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}) - 1)^2 - 1 + 1} \\
&= -\rho_1\langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}} \pm (\mu^2(1 - \langle \mathbf{s},\mathbf{t}\rangle_{\mathcal{F}}) - 1)\,. \qquad (30)
\end{aligned}$$

B Proofs

B.1 Volume Ratio in Terms of Margins

In this section we explicitly derive the volume ratio between the largest inscribable ball in version space and the whole parameter space for the special case of linear kernels in $\mathbb{R}^n$. According to definition (3) we know that the whole parameter space is given by

$$\mathcal{S}_n = \left\{\mathbf{w} : \|\mathbf{w}\|^2 = 1\right\}\,,$$

where in the following $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ denote the classical inner product and norm in $\mathbb{R}^n$, i.e.

$$\langle \mathbf{a},\mathbf{b}\rangle = \sum_{i=1}^{n} a_i b_i, \quad \|\mathbf{w}\|^2 = \sum_{i=1}^{n} w_i^2\,.$$

Given a point $\mathbf{w}' \in \mathcal{S}_n$ and a positive number $0 \leq \gamma \leq r$ we can characterise the ball of radius γ in the parameter space by

$$\begin{aligned}
\mathcal{S}_n(\mathbf{w}',\gamma) &= \left\{\mathbf{w} : \mathbf{w} \in \mathcal{S}_n, \left\|\mathbf{w} - \mathbf{w}'\right\|^2 \leq \gamma^2\right\} \\
&= \left\{\mathbf{w} : \mathbf{w} \in \mathcal{S}_n, \|\mathbf{w}\|^2 - 2\left\langle \mathbf{w},\mathbf{w}'\right\rangle + \left\|\mathbf{w}'\right\|^2 \leq \gamma^2\right\} \\
&= \left\{\mathbf{w} : \mathbf{w} \in \mathcal{S}_n, 2 - 2\left\langle \mathbf{w},\mathbf{w}'\right\rangle \leq \gamma^2\right\} \\
&= \left\{\mathbf{w} : \mathbf{w} \in \mathcal{S}_n, \left\langle \mathbf{w},\mathbf{w}'\right\rangle \geq 1 - \frac{\gamma^2}{2}\right\}
\end{aligned}$$

In the following we will calculate the exact value of the ratio $\frac{\mathrm{vol}(\mathcal{S}_n)}{\mathrm{vol}(\mathcal{S}_n(\mathbf{w}',\gamma))}$ where $\mathbf{w}'$ can be chosen arbitrarily (due to the symmetry of the sphere) and γ equals the observed margin.

Lemma 1. *For linear kernels in* $\mathbb{R}^n$ *the fraction of the whole surface* $\mathrm{vol}\,(\mathcal{S}_n)$ *of the unit sphere to the surface* $\mathrm{vol}\,(\mathcal{S}_n\,(\mathbf{w}',\gamma))$ *with Euclidean distance less than* γ *from any point* $\mathbf{w}' \in \mathcal{S}_n$ *is given by*

$$\frac{\mathrm{vol}\,(\mathcal{S}_n)}{\mathrm{vol}\,(\mathcal{S}_n(\mathbf{w}',\gamma))} = \frac{\int_0^{2\pi} [\sin\,(\theta)]^{n-2}\; d\theta}{\int_0^{\arccos(1-\gamma^2/2)} [\sin\,(\theta)]^{n-2}\; d\theta}\,.$$

Proof. As the derivation requires the calculation of surface integrals on the hypersphere in $\mathbb{R}^n$ we define each admissible $\mathbf{w}$ by its polar coordinates and carry out the integration over the angles. Thus we specify the coordinate transformation $\mathfrak{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ from polar coordinates into Cartesian coordinates, i.e. every $\mathbf{w} \in \mathcal{S}_n \subset \mathbb{R}^n$ is expressed via $n-2$ angles $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n-2})$ ranging from 0 to π, one angle $0 \leq \varphi \leq 2\pi$, and the radius r. This transformation reads

$$\begin{aligned}
w_1 = \mathfrak{f}_1\,(r,\varphi,\boldsymbol{\theta}) &= r\cdot\sin(\varphi)\sin(\theta_1)\cdots\sin(\theta_{n-2}) && (31)\\
w_2 = \mathfrak{f}_2\,(r,\varphi,\boldsymbol{\theta}) &= r\cdot\cos(\varphi)\sin(\theta_1)\cdots\sin(\theta_{n-2}) && (32)\\
\vdots\;\;\vdots &\;\;\vdots && (33)\\
w_{n-1} = \mathfrak{f}_{n-1}\,(r,\varphi,\boldsymbol{\theta}) &= r\cdot\cos(\theta_{n-3})\sin(\theta_{n-2}) && (34)\\
w_n = \mathfrak{f}_n\,(r,\varphi,\boldsymbol{\theta}) &= r\cdot\cos(\theta_{n-2})\,. && (35)
\end{aligned}$$

Without loss of generality we choose $\mathbf{w}'$ to be $\boldsymbol{\theta}' = \mathbf{0}, \varphi' = 0$. Hence the ball of radius γ can be expressed as

$$\begin{aligned}
\mathcal{S}_n\,(\gamma) &= \left\{r,\varphi,\boldsymbol{\theta} : \left\langle \mathfrak{f}\,(r,\varphi,\boldsymbol{\theta})\,, \mathfrak{f}\,\left(r,\varphi',\boldsymbol{\theta}'\right)\right\rangle \geq 1-\frac{\gamma^2}{2}\right\}\\
&= \left\{r,\varphi,\boldsymbol{\theta} : r\cdot\cos(\theta_{n-2}) r\cdot\cos\left(\theta'_{n-2}\right) + \cdots \geq 1-\frac{\gamma^2}{2}\right\}\\
&= \left\{r,\varphi,\boldsymbol{\theta} : \cos(\theta_{n-2}) \geq 1-\frac{\gamma^2}{2}\right\}\\
&= \left\{r,\varphi,\boldsymbol{\theta} : \theta_{n-2} \leq \arccos\left(1-\frac{\gamma^2}{2}\right)\right\}\,,
\end{aligned}$$

using $\sin\,(0) = 0$ and $\cos\,(0) = 1$ in the third line. As can be seen from this expression the margin γ characterising the ball simply possesses a restriction on the angle θ_{n-2} in the integration. Thus, the quantity of interest is given

by

$$\frac{\mathrm{vol}\left(\mathcal{S}_n(2)\right)}{\mathrm{vol}\left(\mathcal{S}_n(\gamma)\right)} = \frac{\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi}\left|J_n\left(r,\varphi,\theta_1,\ldots,\theta_{n-2}\right)\right| d\theta_{n-2}\cdots d\theta_1\, d\varphi}{\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\Psi}\left|J_n\left(r,\varphi,\theta_1,\ldots,\theta_{n-2}\right)\right| d\theta_{n-2}\cdots d\theta_1\, d\varphi}, \tag{36}$$

where $\Psi = \arccos\left(1-\frac{\gamma^2}{2}\right)$ and J_n is the functional determinant of $\mathfrak{f}$ given by Equation (31) to (35). J_n is given by

$$J_n(r,\varphi,\theta_1,\ldots,\theta_{n-2}) = \det \mathbf{J}_n\,, \tag{37}$$

$$\mathbf{J}_n = \begin{pmatrix} \frac{\partial \mathfrak{f}_1(r,\varphi,\boldsymbol{\theta})}{\partial r} & \frac{\partial \mathfrak{f}_1(r,\varphi,\boldsymbol{\theta})}{\partial \varphi} & \cdots & \frac{\partial \mathfrak{f}_1(r,\varphi,\boldsymbol{\theta})}{\partial \theta_{n-2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathfrak{f}_n(r,\varphi,\boldsymbol{\theta})}{\partial r} & \frac{\partial \mathfrak{f}_n(r,\varphi,\boldsymbol{\theta})}{\partial \varphi} & \cdots & \frac{\partial \mathfrak{f}_n(r,\varphi,\boldsymbol{\theta})}{\partial \theta_{n-2}} \end{pmatrix} \tag{38}$$

Hence the n–th row of this matrix contains only two nonzero elements

$$\frac{\partial \mathfrak{f}_n\left(r,\varphi,\boldsymbol{\theta}\right)}{\partial r} = \cos(\theta_{n-2}) \qquad \frac{\partial \mathfrak{f}_n\left(r,\varphi,\boldsymbol{\theta}\right)}{\partial \theta_{n-2}} = -r\cdot\sin\left(\theta_{n-2}\right)\,.$$

Now using the Laplace–expansion of (37) in the n–th row we obtain

$$J_n\left(r,\varphi,\theta_1,\ldots,\theta_{n-2}\right) = \cos\left(\theta_{n-2}\right) A\left(x_n,r\right) - r\sin\left(\theta_{n-2}\right) A\left(x_n,\theta_{n-2}\right)\,,$$

where $A(x_n,r)$ is the algebraic complement of the element $\left(\mathbf{J}_n\right)_{x_n,r}$ of the Jacobian, similarly $A(x_n,\theta_{n-2})$. Let us decompose $A(x_n,r)$ and $A(x_n,\theta_{n-2})$ into

$$\begin{aligned} A(x_n,r) &= J_{n-1}\left(r,\varphi,\theta_1,\ldots,\theta_{n-3}\right)\cdot(-1)^{n+1}\cdot(-1)^{n-2}\cdot r\left[\sin\left(\theta_{n-2}\right)\right]^{n-2}\cos\left(\theta_{n-2}\right)\,, \\ A(x_n,\theta_{n-2}) &= J_{n-1}\left(r,\varphi,\theta_1,\ldots,\theta_{n-3}\right)\cdot(-1)^{2n}\cdot\left[\sin\left(\theta_{n-2}\right)\right]^{n-1}\,. \end{aligned}$$

The first factor is the determinant of the submatrix of $\mathbf{J}_{n-1}$ obtained by deletion of the n–th row and n–th column of $\mathbf{J}$. The second factor is the checkerboard term of the algebraic complement whereas the third factor in the term $A(x_n,r)$ specifies the number of column flips to transform the column order of the matrix obtained by deletion of the first column and n–th row into the column order of $\mathbf{J}_{n-1}$. The last factor gives the factor which is missing in $J_{n-1}\left(r,\varphi,\theta_1,\ldots,\theta_{n-3}\right)$. As an example consider the special case

of $n = 3$ and $n = 4$:

$$J_3(r,\varphi,\theta_1) = \det\begin{pmatrix} \cos(\varphi)\sin(\theta_1) & -r\sin(\varphi)\sin(\theta_1) & r\cos(\varphi)\cos(\theta_1) \\ \sin(\varphi)\sin(\theta_1) & r\cos(\varphi)\sin(\theta_1) & r\sin(\varphi)\cos(\theta_1) \\ \underbrace{\cos(\theta_1)}_{\mathbf{j}_1} & \underbrace{0}_{\mathbf{j}_2} & \underbrace{-r\sin(\theta_1)}_{\mathbf{j}_3} \end{pmatrix},$$

$$J_4(r,\varphi,\theta_1,\theta_2) = \det\begin{pmatrix} \sin(\theta_2)\,\mathbf{j}_1 & \sin(\theta_2)\,\mathbf{j}_2 & \sin(\theta_2)\,\mathbf{j}_3 & r\cos(\theta_2)\,\mathbf{j}_1 \\ \cos(\theta_2) & 0 & 0 & -r\sin(\theta_2) \end{pmatrix}.$$

Hence, $J_n(r,\varphi,\theta_1,\ldots,\theta_{n-2})$ is given by

$$\begin{aligned} J_n(r,\varphi,\theta_1,\ldots,\theta_{n-2}) &= J_{n-1}(r,\varphi,\theta_1,\ldots,\theta_{n-3})\cdot \\ &\quad -r\left([\sin(\theta_{n-2})]^{n-2}[\cos(\theta_{n-2})]^2 + [\sin(\theta_{n-2})]^{n-2}[\sin(\theta_{n-2})]^2\right) \end{aligned}$$

As a result, we obtain

$$J_n(r,\varphi,\theta_1,\ldots,\theta_{n-2}) = -r\,[\sin(\theta_{n-2})]^{n-2}\cdot J_{n-1}(r,\varphi,\theta_1,\ldots,\theta_{n-3}),$$

which back–inserted into Equation (36) gives

$$\frac{\mathrm{vol}(\mathcal{S}_n)}{\mathrm{vol}(\mathcal{S}_n(\gamma))} = \frac{\int_0^{\pi}[\sin(\theta_{n-2})]^{n-2}\,d\theta_{n-2}}{\int_0^{\Psi}[\sin(\theta_{n-2})]^{n-2}\,d\theta_{n-2}},$$

where $\Psi = \arccos\left(1-\frac{\gamma^2}{2}\right)$. The lemma is proven. □

Now we prove a lemma which can be used to bound the ratio $\frac{\mathrm{vol}(\mathcal{S}_n(2))}{\mathrm{vol}(\mathcal{S}_n(\gamma))}$ from above. Note that for tighter bounds on the volume ratio one only needs to evaluate the previous expression. Here, one can make use of an expansion of the fraction in terms of the binomial coefficients.

Lemma 2. *For all $k \in \mathbb{N}$ and all $0 < x < \frac{1}{2}$*

$$\ln\left(\frac{\int_0^{\pi}[\sin(\theta)]^{2k+1}\,d\theta}{\int_0^{\arccos(1-2x)}[\sin(\theta)]^{2k+1}\,d\theta}\right) \leq -(2k+1)\ln(x). \qquad (39)$$

Proof. From [3] we know that for all $k \in \mathbb{N}$

$$\int [\sin(\theta)]^{2k+1}\, d\theta = -\frac{\cos(\theta)}{2k+1}\left(B(k,0) + \sum_{i=1}^{k} [\sin(\theta)]^{2i} B(k,i)\right), \quad (40)$$

$$B(k,i) = \frac{2(i+1)\cdot 2(i+2)\cdots 2k}{(2i+1)\cdot(2i+3)\cdots(2k-1)} \quad (41)$$

$$= \frac{2\cdot 4\cdots 2k}{1\cdot 3\cdots(2k-1)} \cdot \frac{1\cdot 3\cdots(2i-1)}{2\cdot 4\cdots(2i)} \quad (42)$$

$$= \frac{4^k (k!)^2 (2i)!}{(2k)!\,(i!)^2\, 4^i} = \frac{4^k}{4^i}\frac{\binom{2i}{i}}{\binom{2k}{k}}. \quad (43)$$

Let us introduce the abbreviation

$$S(k,x) = \int_0^{\arccos(1-2x)} [\sin(\theta)]^{2k+1}\, d\theta\,.$$

Then the numerator of (39) is given by $S(k,1)$ whereas the denominator of (39) is simply $S(k,x)$. From Equation (40) we see

$$\begin{aligned} S(k,x) &= -\frac{\cos(\theta)}{2k+1}\left(B(k,0) + \sum_{i=1}^{k} [\sin(\theta)]^{2i} B(k,i)\right)\Bigg|_0^{\arccos(1-2x)} \\ &= \frac{1}{2k+1}\left(B(k,0) - (1-2x)\,B(k,0) - (1-2x)\sum_{i=1}^{k}\left(4x-4x^2\right)^i B(k,i)\right) \\ &= \frac{1}{2k+1}\frac{4^k}{\binom{2k}{k}}\left(1 + (2x-1) + (2x-1)\sum_{i=1}^{k}\binom{2i}{i} x^i (1-x)^i\right). \end{aligned}$$

where we have used

$$[\sin(\theta)]^{2i} = \left[\sin^2(\theta)\right]^i = \left[1-\cos(\theta)^2\right]^i = \left(1-(1-2x)^2\right)^i = \left(4x-4x^2\right)^i.$$

For the fraction we obtain

$$\ln\left(\frac{S(k,1)}{S(k,x)}\right) = \ln\left(\frac{2}{2x + (2x-1)\sum_{i=1}^{k}\binom{2i}{i} x^i (1-x)^i}\right)$$

In Lemma 3 we show that for any $k \in \mathbb{N}^+$ and $0 < x < \frac{1}{2}$

$$\sum_{i=1}^{k} \left(\begin{array}{c} 2i \\ i \end{array} \right) x^i \left(1-x\right)^i \leq \frac{2\left(x^{2k+1}-x\right)}{2x-1}\,.$$

Back–inserted into the last expression we obtain

$$\begin{aligned} \ln\left(\frac{S(k,1)}{S(k,x)}\right) &\leq \ln\left(\frac{2}{2x+(2x-1)\,\frac{2\left(x^{2k+1}-x\right)}{(2x-1)}}\right) = \ln\left(\frac{2}{2x^{2k+1}}\right) \\ &= \ln\left(\frac{1}{x^{2k+1}}\right) = -\left(2k+1\right)\ln\left(x\right)\,, \end{aligned}$$

which proves the lemma. □

Lemma 3. *For any $k \in \mathbb{N}^+$ and $0 < x < \frac{1}{2}$*

$$\sum_{i=1}^{k} \left(\begin{array}{c} 2i \\ i \end{array} \right) x^i \left(1-x\right)^i \leq \frac{2\left(x^{2k+1}-x\right)}{2x-1}\,.$$

Proof. In order to prove the lemma we note that

$$\sum_{i=1}^{\infty} \left(\begin{array}{c} 2i \\ i \end{array} \right) x^i (1-x)^i = \frac{2x}{1-2x}\,. \tag{44}$$

This can be seen by considering

$$\begin{aligned} \arcsin\left(u\right) &= u + \sum_{i=1}^{\infty} \left(\begin{array}{c} 2i \\ i \end{array} \right) \frac{1}{4^i} \frac{u^{2i+1}}{2i+1} \\ \frac{d\arcsin\left(u\right)}{du} &= 1 + \sum_{i=1}^{\infty} \left(\begin{array}{c} 2i \\ i \end{array} \right) \frac{1}{4^i} u^{2i} = \frac{1}{\sqrt{1-u^2}}\,, \end{aligned}$$

Using $u = 2\sqrt{x\left(1-x\right)}$ we obtain Equation (44). In the next step we show the following lower bound

$$\sum_{i=k+1}^{\infty} \left(\begin{array}{c} 2i \\ i \end{array} \right) x^i (1-x)^i \geq \frac{2x^{2k+1}}{1-2x}\,.$$

This can be achieved by renumbering and componentwise comparison of the resulting sequence, i.e.

$$\begin{aligned}\sum_{i=k+1}^{\infty}\left(\begin{array}{c}2i\\ i\end{array}\right)x^i(1-x)^i &= \sum_{j=1}^{\infty}\left(\begin{array}{c}2\,(k+j)\\ k+j\end{array}\right)x^{k+j}(1-x)^{k+j}\\ &\geq \sum_{j=1}^{\infty}\left(\begin{array}{c}2j\\ j\end{array}\right)x^{2k+j}(1-x)^j = x^{2k}\frac{2x}{1-2x}\,,\end{aligned}$$

where we used $\left(\begin{array}{c}2\,(k+j)\\ k+j\end{array}\right) \geq \left(\begin{array}{c}2j\\ j\end{array}\right)$ which holds for all $k \in \mathbb{N}^+$, and $x^{k+j}\,(1-x)^{k+j} \geq x^{2k+j}\,(1-x)^j \Leftrightarrow (1-x)^k \geq x^k$ which holds for all $0 < x < \frac{1}{2}$. Finally, we combine the two statements to prove the lemma. Thus we see

$$\begin{aligned}\sum_{i=1}^{k}\left(\begin{array}{c}2i\\ i\end{array}\right)x^i(1-x)^i &= \sum_{i=1}^{\infty}\left(\begin{array}{c}2i\\ i\end{array}\right)x^i(1-x)^i - \sum_{i=k+1}^{\infty}\left(\begin{array}{c}2i\\ i\end{array}\right)x^i(1-x)^i\\ &\leq \frac{2x}{1-2x} - \frac{2x^{2k+1}}{1-2x} = \frac{2\left(x-x^{2k+1}\right)}{1-2x}\\ &= \frac{2\left(x^{2k+1}-x\right)}{2x-1}\,.\end{aligned}$$

□

B.2 Bayes–Admissibility and Point–Symmetry

Let us formally introduce the property of *point–symmetry* and *Bayes–admissibility* of a compact convex set V in any metric space.

Definition 3. A compact convex set V in a vector space is said to be *point–symmetric* iff

$$\exists \mathbf{w} \in V : \forall \mathbf{v} \in V\ \mathbf{v} + 2\,(\mathbf{w}-\mathbf{v}) \in V\,.$$

A compact convex set V in a metric space is said to be *Bayes–admissible* iff there exists a $\mathbf{w}$ such that

$$\begin{aligned}
\bigcap_{\mathbf{n}} H(\mathbf{n},\mathbf{w}) &= \mathbf{w}\,, \\
H(\mathbf{n},\mathbf{w}) &= \{\mathbf{v} : \langle \mathbf{v}-\mathbf{w},\mathbf{n}\rangle_{\mathcal{F}} = 0 \wedge \mathrm{vol}(\mathcal{W}_{+1}(\mathbf{w},\mathbf{n})) = \mathrm{vol}(\mathcal{W}_{-1}(\mathbf{w},\mathbf{n}))\}\,, \\
\mathcal{W}_y(\mathbf{w},\mathbf{n}) &= \{\mathbf{v}' \in V : \mathrm{sign}(\langle \mathbf{v}'-\mathbf{w},\mathbf{n}\rangle_{\mathcal{F}}) = y\}\,.
\end{aligned}$$

The following lemma is of interest for the construction of Bayes–admissible sets given a point $\mathbf{w}$.

Lemma 4. *The following two statements are equivalent*

- A compact convex set V is point–symmetric.
- A compact convex set V is Bayes–admissible.

Proof. First we show that any point–symmetric set is Bayes–admissible. Consider any normal vector $\mathbf{n}$, the symmetry centre $\mathbf{w} \in V$, and any $\mathbf{v} \in \mathcal{W}_{+1}(\mathbf{w},\mathbf{n})$. Then we know from the property of point–symmetry that there exists a unique vector $\mathbf{v}' = \mathbf{v} + 2(\mathbf{w}-\mathbf{v}) \in V$ and $\mathbf{v}' \in \mathcal{W}_{-1}(\mathbf{w},\mathbf{n})$. Since this holds for any $\mathbf{n}$ and any $\mathbf{v} \in \mathcal{W}_{+1}(\mathbf{w},\mathbf{n})$ it follows that $\mathrm{vol}(\mathcal{W}_{+1}(\mathbf{w},\mathbf{n})) = \mathrm{vol}(\mathcal{W}_{-1}(\mathbf{w},\mathbf{n}))$. Hence, point–symmetry implies Bayes–admissibility.

Now we prove that any Bayes–admissible set V is point–symmetric. Let us represent the convex set V by polar coordinates. Hence, we represent the convex set V by a boundary function $f(\varphi,\boldsymbol{\theta}) \geq 0$. Without loss of generality we assume that the Bayes–point is located at the origin of the coordinate system. Hence, points are represented by one angle $0 \leq \varphi \leq 2\pi$, n angles $0 \leq \theta_i \leq \pi$, and a radius $0 \leq r \leq f(\varphi,\boldsymbol{\theta})$. Then, V is point–symmetric w.r.t. to the origin iff

$$\forall\varphi\forall\theta_1 \cdots \forall\theta_n \qquad f(\varphi,\boldsymbol{\theta}) = f(\varphi-\pi, \pi\mathbf{1}-\boldsymbol{\theta})\,.$$

Without loss of generality we can assume that the two subvolumes of equal volume (induced by the intersection of $n+2$ Bayes–lines) are given by $0 \leq \varphi \leq \Delta_0, 0 \leq \theta_i \leq \Delta_n$, $i = 1 \ldots n$ and $\pi \leq \varphi' \leq \pi + \Delta_0, \pi \leq \theta_i' \leq \pi - \Delta_n$, $i =$

$1 \ldots n$. Hence by the Bayes–admissibility we have

$$\begin{aligned}
\int_0^{\Delta_0} \int_0^{\Delta_1} \cdots \int_0^{\Delta_n} \int_0^{f(\varphi,\boldsymbol{\theta})} J_{n+2}\,(r,\varphi,\boldsymbol{\theta})\; dr\, d\theta_n \cdots d\theta_1\, d\varphi \quad &= \\
\int_\pi^{\pi+\Delta_0} \int_\pi^{\pi-\Delta_1} \cdots \int_\pi^{\pi-\Delta_n} \int_0^{f(\varphi,\boldsymbol{\theta})} J_{n+2}\,(r,\varphi,\boldsymbol{\theta})\; dr\, d\theta_n \cdots d\theta_1\, d\varphi \quad &= \\
\int_0^{\Delta_0} \int_0^{\Delta_1} \cdots \int_0^{\Delta_n} \int_0^{f(\varphi-\pi,\pi\mathbf{1}-\boldsymbol{\theta})} J_{n+2}\,(r,\varphi-\pi,\pi\mathbf{1}-\boldsymbol{\theta})\; dr\, d\theta_n \cdots d\theta_1\, d\varphi \quad &.
\end{aligned}$$

Now using the following two properties of the functional determinant J_{n+2}

$$\begin{aligned}
|J_{n+2}\,(r,\varphi,\boldsymbol{\theta})\,| &= r^{n+1} |K_{n+2}\,(\varphi,\boldsymbol{\theta})\,|\,, \\
|J_{n+2}\,(r,\varphi,\boldsymbol{\theta})\,| &= |J_{n+2}\,(r,\varphi-\pi,\pi\mathbf{1}-\boldsymbol{\theta})\,|\,,
\end{aligned}$$

we see that by Bayes–admissibility

$$\int_0^{f(\varphi,\boldsymbol{\theta})} r^{n+1}\, dr = \int_0^{f(\varphi-\pi,\pi\mathbf{1}-\boldsymbol{\theta})} r^{n+1}\, dr\,,$$

which gives

$$f\,(\varphi,\boldsymbol{\theta}) = f\,(\varphi-\pi,\pi\mathbf{1}-\boldsymbol{\theta})\;.$$

This is the defining property of point–symmetry and proves the lemma.

□

Algorithm 1 Pseudocode of the Billiard algorithm

Require: TOL < 1
Require: $\tau_{\max} \in \mathbb{R}^+$
Ensure: $y_j \sum_i \gamma_i k(\mathbf{x}_i, \mathbf{x}_j) > 0 \qquad j = 1, \ldots, \ell$
  $\boldsymbol{\alpha} = \mathbf{0}$
  $\boldsymbol{\beta}$ = random; normalise $\boldsymbol{\beta}$ using Equation (10)
  $\Lambda = p_{\max} = \lambda_{\max} = 0$
  while $\rho_2 \left(p_{\max}, \lambda_{\max}/\Lambda\right) >$ TOL **do**
    repeat
      for $i = 1, \ldots, \ell$ **do**
        $d_i = y_i \sum_j \gamma_j k(\mathbf{x}_j, \mathbf{x}_i)$
        $\nu_i = y_i \sum_j k(\mathbf{x}_j, \mathbf{x}_i)$
        $\tau_i = -d_i/\nu_i$
      end for
      $m^{'} = \min_{i:\tau_i > 0} \tau_i$
      if $\tau_{m'} \geq \tau_{\max}$ **then**
        $\boldsymbol{\beta}$ = random, but fulfils Equation (12)
        normalise $\boldsymbol{\beta}$ using Equation (10)
      else
        $m = m^{'}$
      end if
    until $\tau_{m'} < \tau_{\max}$
    $\boldsymbol{\gamma}' = \boldsymbol{\gamma} + \tau_m \boldsymbol{\beta}$; normalise $\boldsymbol{\beta}'$ using Equation (10)
    $\beta_m = \beta_m - 2\nu_m y_m / k(\mathbf{x}_m, \mathbf{x}_m)$; normalise $\boldsymbol{\beta}$ using Equation (10)
    $\boldsymbol{\zeta} = \boldsymbol{\gamma} + \boldsymbol{\gamma}'$; normalise $\boldsymbol{\zeta}$ using Equation (10)
    $\lambda = \sqrt{\sum_{i,j} \left(\gamma_i - \gamma_i'\right) \left(\gamma_j - \gamma_j'\right) k(\mathbf{x}_i, \mathbf{x}_j)}$
    $p = \sum_{i,j} \zeta_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$
    $\boldsymbol{\alpha} = \rho_1 \left(p, \frac{\Lambda}{\Lambda+\lambda}\right) \boldsymbol{\alpha} + \rho_2 \left(p, \frac{\Lambda}{\Lambda+\lambda}\right) \boldsymbol{\zeta}$
    $p_{\max} = \max\left(p, p_{\max}\right)$
    $\lambda_{\max} = \max\left(\lambda, \lambda_{\max}\right)$
    $\Lambda = \Lambda + \lambda$
    $\boldsymbol{\gamma} = \boldsymbol{\gamma}'$
  end while

References

[1] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999. to appear.

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

[3] G. P. Bois. *Tables of Indefinite Integrals*. Dover Publications, 1961.

[4] I. Cornfeld, S. Fomin, and Y. Sinai. *Ergodic Theory*. Springer Verlag, 1982.

[5] C. Cortes and V. Vapnik. Support Vector Networks. *Machine Learning*, 20:273–297, 1995.

[6] N. Cristianini, J. Shawe-Taylor, and P. Sykacek. Bayesian classifiers are large margin hyperplanes in a hilbert space. Technical report, Royal Holloway, University of London, 1998. NC2–TR–1998–008.

[7] W. Feller. *An Introduction To Probability Theory and Its Application*, volume 1. Jon Wiley, New York and Sons, 1950.

[8] D. A. Harville. *Matrix Algebra From a Statistican's Perspective*. Springer–Verlag, New York, 1997.

[9] D. Haussler, M. Kearns, and R. Shapire. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning*, 14:88–113, 1994.

[10] A. Kolmogorov and S.V.Fomin. *Functional Analysis*. Graylock Press, 1957.

[11] D. A. McAllester. Some PAC Bayesian theorems. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 230–234, Madison, Wisconsin, 1998.

[12] T. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Transaction of London Philosophy Society (A)*, 209:415–446, 1909.

[13] C. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

[14] R. M. Neal. Markov chain monte carlo method based on 'slicing' the density function. Technical report, Department of Statistics, University of Toronto, 1997. TR–9722.

[15] M. Opper and D. Haussler. Generalization performance of bayes optimal classification algorithm for learning a perceptron. *Physical Review Letters*, 66:2677, 1991.

[16] M. Opper and W. Kinzel. *Statistical Mechanics of Generalisation.* Springer, 1995.

[17] M. Opper, W. Kinzel, J. Kleinz, and R. Nehl. On the ability of the optimal perceptron to generalise. *Journal of Physics*, 23:581–586, 1990.

[18] M. Rosenblatt. *Principles of neurodynamics: Perceptron and Theory of Brain Mechanisms.* Spartan–Books, Washington D.C., 1962.

[19] P. Rujàn. Playing billiard in version space. *Neural Computation*, 9:99–122, 1997.

[20] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proccedings of the 14–th International Conference in Machine Learning*, 1997.

[21] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data–dependent hierarchies. Technical report, Royal Holloway, University of London, 1996. NC–TR–1996–053.

[22] J. Shawe-Taylor and N. Cristianini. Robust bounds on generalization from the margin distribution. Technical report, Royal Holloway, University of London, 1998. NC2–TR–1998–029.

[23] J. Shawe-Taylor and R. C. Williamson. A PAC analysis of a Bayesian estimator. Technical report, Royal Holloway, University of London, 1997. NC2–TR–1997–013.

[24] UCI. University of California Irvine: Machine Learning Repository, 1990.

[25] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

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

[27] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[28] V. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, New York, 1998.

[29] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Application*, 16(2):264–281, 1971.

[30] T. Watkin. Optimal learning with a neural network. *Europhysics Letters*, 21:871–877, 1993.

[31] G. Whaba. *Spline Models for Observational Data.* Society for Industrial and Applied Mathematics, Philadelphia, 1990.