Summary. We investigate the generalisation performance of consistent classifiers, i.e. classifiers that are contained in the so-called *version space*, both from a theoretical and experimental angle. In contrast to classical VC analysis—where no single classifier within version space is singled out on grounds of a generalisation error bound—the data dependent structural risk minimisation framework suggests that there exists one *particular* classifier that is to be preferred because it minimises the generalisation error bound. This is usually taken to provide a theoretical justification for learning algorithms such as the well known support vector machine. A reinterpretation of a recent PAC-Bayesian result, however, reveals that given a suitably chosen hypothesis space there exists a large fraction of classifiers with small generalisation error albeit we cannot identify them for a specific learning task. In the particular case of linear classifiers we show that classifiers found by the classical perceptron algorithm have guarantees bounded by the size of version space. These results are complemented with an empirical study for kernel classifiers on the task of handwritten digit recognition which demonstrates that even classifiers with a small margin may exhibit excellent generalisation. In order to perform this analysis we introduce the kernel Gibbs sampler—an algorithm which can be used to sample consistent kernel classifiers.

1

The Structure of Version Space

Ralf Herbrich[1], Thore Graepel[1], and Robert C. Williamson[2]

[1] Microsoft Research Cambridge, Cambridge, UK
[2] RSISE, Australian National University, Australia

1.1 Introduction

Over the last ten years, machine learning has received a boost due to the ground-breaking results on the generalisation error of classifiers (see [26, 30]). Their results build the theoretical basis for the well-known support vector machine (SVM) algorithm. It is now widely accepted that for complex models it is necessary to use regularisation techniques such as margin maximisation in order to find a classifier exhibiting a small generalisation error (see [29, p. 157]). Since for large datasets the SVM algorithm is too time consuming many heuristics to approximate the SVM solution have been put forward (see, e.g. [14, 23, 28]). Recently, it has been demonstrated experimentally that even algorithms with no explicit regularisation perform comparably to SVMs (see [11, 20]). This observation raises an interesting question:

What fraction of classifiers within version space exhibit a small generalisation error?

In this paper we try to answer this question both from a theoretical and experimental point of view. Using a recent result in the PAC-Bayesian framework we are able to show that given a suitably chosen hypothesis space there exists a large fraction of classifiers with small generalisation error. More precisely, *the generalisation error of most of the classifiers in version space is controlled by the size of the version space relative to the size of the hypothesis space.* This result, which we call the *egalitarian* generalisation error bound, is complemented by an experimental study for linear classifiers on the task of handwritten digit recognition using the MNIST database. It is worthwhile mentioning that in a fully Bayesian treatment the size of version space is also called the *evidence* of the model or hypothesis space, respectively (see [18]).

The paper is structured as follows: in the following section we review generalisation error bounds for single classifiers consistent with the whole training sample. We will also introduce the PAC-Bayesian framework and its main result which allows us to give our main theoretical result together with its proof at the end of this section. In the subsequent section we discuss the impact of this result for practical learning theory. We also give a more specific result for the perceptron learning algorithm that points into the same direction. In Section 1.4 we present the kernel Gibbs sampler algorithm which allows us to validate our theoretical result on a benchmark problem in the field of handwritten digit recognition. The paper concludes with a discussion of generalisation error bounds for specific algorithms as opposed to bounds that hold *uniformly* over version space.

We denote a probability measure by P_{X}; random variables are typeset in upper capital sans-serif font. The symbols E and I denote the expectation of a random variable and the indicator function, respectively. We use bold roman font for vectors $\mathbf{x}$ and denote tuples by $\boldsymbol{x}$. Finally, the symbol ℓ_2^n denotes the space of all sequences $\mathbf{x} = (x_1, \ldots, x_n)$ of length n for which $\sum_{i=1}^{n} x_i^2 < \infty$.

1.2 Generalisation Error Bounds for Consistent Classifiers

Suppose we are given a sample $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathcal{X}^m$ together with a sample $\boldsymbol{y} = (y_1, \ldots, y_m) \in \mathcal{Y}^m = \{-1, +1\}^m$ drawn iid from an unknown distribution $\mathsf{P}_{\mathsf{Z}} = \mathsf{P}_{\mathsf{XY}}$. Furthermore, assume we are given a fixed *hypothesis space* $\mathcal{H}$ of functions $h : \mathcal{X} \to \mathcal{Y}$. We consider learning algorithms that aim at finding a function $h^* \in \mathcal{H}$ that minimises the *generalisation error* $R[h]$ given by

$$R[h] = \mathsf{P}_{\mathsf{XY}}(h(\mathsf{X}) \neq \mathsf{Y}) = \mathsf{E}_{\mathsf{XY}}\left[\mathsf{I}_{h(\mathsf{X}) \neq \mathsf{Y}}\right] .$$

A common approach to (approximately) finding h^* based on the training sample $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{Z}^m$ is to select a function $h \in \mathcal{H}$ that minimises the *training error* $R_{\text{emp}}[h, \boldsymbol{z}]$

$$R_{\text{emp}}[h, \boldsymbol{z}] = \frac{1}{m} \sum_{(x_i, y_i) \in \boldsymbol{z}} \mathsf{I}_{h(x_i) \neq y_i} .$$

Let us assume that $\mathsf{P}_{\mathsf{Y}|\mathsf{X}=x}(y) = \mathsf{I}_{h^*(x)=y}$, i.e. h^* deterministically labels all the data and thus has minimal generalisation error. Then we define the *version space* $V(\boldsymbol{z})$ (phrase due to T. Mitchell [21]) as the set of all classifiers $h \in \mathcal{H}$ that are *consistent* with the training sample $\boldsymbol{z}$,

$$V(\boldsymbol{z}) = \{h \in \mathcal{H} \mid R_{\text{emp}}[h, \boldsymbol{z}] = 0\} .$$

Of course, solely based on the training error $R_{\text{emp}}[h, \boldsymbol{z}]$ all classifiers in version space are indistinguishable. Moreover, even if a classifier has zero training error it can happen that its generalisation error is large—an effect known as *overfitting.* In order to cope with this uncertainty a lot of research has been done to obtain probabilistic bounds on the generalisation error of consistent classifiers. The basic idea is to guarantee that for most training trials (random training samples) the generalisation error of a consistent classifier does not exceed a certain value.

Definition 1 (PAC Generalisation Error Bound). *A function* $\varepsilon : \mathbb{N} \times \mathcal{H} \times \cup_{m=1}^{\infty} \mathcal{Z}^m \times [0,1] \to \mathbb{R}$ *such that for all measures* P_{Z}*, for all* $m \in \mathbb{N}$ *and for all* $\delta \in (0,1]$

$$\mathsf{P}_{\mathsf{Z}^m}\left(\forall h \in \mathcal{H} : \left(h \notin V\left(\mathbf{Z}\right)\right) \vee \left(R\left[h\right] < \varepsilon\left(m, h, \mathbf{Z}, \delta\right)\right)\right) \geq 1 - \delta \tag{1.1}$$

is called a PAC generalisation error bound *for the hypothesis space* $\mathcal{H}$*.*

Classical VC theory (see [29,30]) provides the following bound for all $m > d_{\mathcal{H}}$ and for all hypotheses $h \in \mathcal{H}$:

$$\varepsilon_{\mathrm{VC}}\left(m, h, \boldsymbol{z}, \delta\right) = \varepsilon_{\mathrm{VC}}\left(m, \delta\right) = \frac{4}{m}\left(\ln\left(\left(\frac{2em}{d_{\mathcal{H}}}\right)^{d_{\mathcal{H}}}\right) + \ln\left(\frac{2}{\delta}\right)\right) , \tag{1.2}$$

where $d_{\mathcal{H}}$ is known as the *VC dimension* of the hypothesis space $\mathcal{H}$ (see [30] for more details). Obviously, the generalisation error *bound* is *independent* of the particular classifier $h \in V\left(\boldsymbol{z}\right)$ and as such no single classifier $h \in V\left(\boldsymbol{z}\right)$ is singled out on the basis of VC theory.

However, in applied classification learning it is common practice that the classification is carried out by thresholding a real-valued function, i.e. $h\left(x\right) = \operatorname{sign}\left(f\left(x\right)\right)$. It can be shown that the additional information of the real-valued magnitude $\left|f\left(x\right)\right|$ *before* thresholding allows one to obtain a generalisation error bound in terms of the margin $\gamma_{\boldsymbol{z}}\left(h\right) = \min_{(x_i,y_i)\in\boldsymbol{z}} y_i f\left(x_i\right)$ attained on the given sample $\boldsymbol{z}$, i.e., for all hypotheses $h \in \mathcal{H}$ and $m > d_{\mathcal{H}}\left(\tilde{\gamma}_{\boldsymbol{z}}\left(h\right)\right)$, $\tilde{\gamma}_{\boldsymbol{z}}\left(h\right) := \gamma_{\boldsymbol{z}}\left(h\right)/8$

$$\begin{aligned} &\varepsilon_{\mathrm{fat}}\left(m, h, \boldsymbol{z}, \delta\right) = \varepsilon_{\mathrm{fat}}(m, \tilde{\gamma}_{\boldsymbol{z}}\left(h\right), \delta) \\ &= \frac{2}{m}\left(\log_2\left(\left(\frac{8em}{d_{\mathcal{H}}\left(\tilde{\gamma}_{\boldsymbol{z}}\left(h\right)\right)}\right)^{d_{\mathcal{H}}\left(\tilde{\gamma}_{\boldsymbol{z}}\left(h\right)\right)}\right)\log_2\left(32m\right) + \log_2\left(\frac{2m}{\delta}\right)\right) \end{aligned} \tag{1.3}$$

where $d_{\mathcal{H}}\left(\gamma\right)$ is known as the *fat shattering* dimension of the hypothesis space $\mathcal{H}$ at the observed scale γ (see [13,26] for details). The function $d_{\mathcal{H}} : \mathbb{R}^+ \to \mathbb{N}$ is always monotonically non-increasing and is a straightforward generalisation of the VC dimension to sets of real valued functions. An immediate consequence of this result is that the *bound on the* generalisation error $R\left[h\right]$ depends inversely on the margin $\gamma_{\boldsymbol{z}}\left(h\right)$. As such the result singles out *one* classifier within version space — the classifier with maximal margin also known as the support vector solution (see [29]).

Recently, D. McAllester presented “some PAC–Bayesian theorems” [19] which provide a generalisation error bound for the Gibbs classification strategy $\mathrm{Gibbs}_{\boldsymbol{z}}$. Given a prior P_{H} over hypothesis space $\mathcal{H}$ and a training sample $\boldsymbol{z}$, for each test example x the Gibbs classification strategy samples a classifier $h \in V(\boldsymbol{z})$ according to $\mathbf{P}_{\mathsf{H}|\mathsf{H}\in V(\boldsymbol{z})}$ and uses it for classification $\mathrm{Gibbs}_{\boldsymbol{z}}(x)$. Note that $\mathrm{Gibbs}_{\boldsymbol{z}}$ does not correspond to any *single* classifier $h \in V(\boldsymbol{z})$ but to a classification strategy based on $\mathbf{P}_{\mathsf{H}|\mathsf{H}\in V(\boldsymbol{z})}$. For any prior $\mathbf{P}_{\mathsf{H}}$, the PAC bound $\varepsilon_{\mathrm{Gibbs}}$ on the generalisation error $R[\mathrm{Gibbs}_{\boldsymbol{z}}] = \mathbf{E}_{\mathsf{H}|\mathsf{H}\in V(\boldsymbol{z})}[R[\mathsf{H}]]$ of this stochastic classification strategy is given by

$$\varepsilon_{\mathrm{Gibbs}}(m, \mathbf{P}_{\mathsf{H}}, \boldsymbol{z}, \delta) = \frac{1}{m}\left(\ln\left(\frac{1}{\mathbf{P}_{\mathsf{H}}(V(\boldsymbol{z}))}\right) + \ln\left(\frac{em^2}{\delta}\right)\right), \qquad (1.4)$$

hence

$$\mathbf{P}_{\mathsf{Z}^m}\left(R\left[\mathrm{Gibbs}_{\mathbf{Z}}\right] \leq \varepsilon_{\mathrm{Gibbs}}(m, \mathbf{P}_{\mathsf{H}}, \mathbf{Z}, \delta)\right) \geq 1-\delta\,. \qquad (1.5)$$

The first term in (1.2)—which is driven by the worst case number of equivalence classes w.r.t. the two classes $y \in \mathcal{Y}$—has been replaced by a *data-dependent* quantity—the prior belief $\mathbf{P}_{\mathsf{H}}$ in consistent classifiers $h \in V(\boldsymbol{z})$. As opposed to classical PAC generalisation error bounds, this result *does not provide any guarantee for single classifiers* $h \in V(\boldsymbol{z})$. The first theoretical result of the present paper is a direct consequence of (1.4) and is stated in the following theorem.

Theorem 1 (Egalitarian Bound). *For all measures $\mathbf{P}_{\mathsf{Z}}$, with probability at least $1-\delta$ over the random draw of the training sample $\boldsymbol{z}$ of size m, for all $\eta > 1$, at least a fraction of $1-\frac{1}{\eta}$ of the classifiers in version space $V(\boldsymbol{z})$ have generalisation error less than*

$$\eta \cdot \varepsilon_{\mathrm{Gibbs}}(m, \mathbf{U}_{\mathsf{H}}, \boldsymbol{z}, \delta)\,,$$

where $\mathbf{U}_{\mathsf{H}}$ is the uniform measure over $\mathcal{H}$.

Proof. The proof is a simple application of Markov's inequality along with the instantiation of $\mathbf{P}_{\mathsf{H}}$ by the uniform measure $\mathbf{U}_{\mathcal{H}}$. Markov's inequality says

$$\forall \eta > 1: \qquad \mathbf{P}_{\mathsf{H}|\mathsf{H}\in V(\mathbf{Z})}\left(R[\mathsf{H}] \geq \eta \cdot \mathbf{E}_{\mathsf{H}|\mathsf{H}\in V(\mathbf{Z})}[R[\mathsf{H}]]\right) < \frac{1}{\eta}\,,$$

because the generalisation error $R: \mathcal{H} \to [0,1]$ as a functional over hypotheses is a positive random variable. Hence, from (1.5) it follows

$$\mathbf{P}_{\mathsf{Z}^m}\left(\forall \eta > 1: \mathbf{P}_{\mathsf{H}|\mathsf{H}\in V(\mathbf{Z})}\left(R[\mathsf{H}] < \eta \cdot \varepsilon_{\mathrm{Gibbs}}(m, \mathbf{U}_{\mathsf{H}}, \mathbf{Z}, \delta)\right) \geq 1 - \frac{1}{\eta}\right) \geq 1-\delta\,.$$

In the following section we shall discuss this results and its impact on the structure of version space. However, one of the most intriguing features of this generalisation error bound is that it holds true regardless of any property of the single classifiers considered. In fact, the only quantity that drives the generalisation error bound is the volume of version space which is a *property of the model $\mathcal{H}$ and the data $\boldsymbol{z}$* but not of single classifiers h.

1.3 Consequences of the Egalitarian Bound

1.3.1 Linear Classifiers

Consider the result of Theorem 1 with $\eta = 2$ and the hypothesis space $\mathcal{H}$ used in SVMs. In this case we know that with high probability ($\geq 1 - \delta$) the generalisation error of at least half of the classifiers in version space $V(\boldsymbol{z})$ are bounded by at most twice the generalisation error of the Gibbs classification strategy. This should be compared with a typical generalisation error bound for linear classifiers in terms of margins (see [10])

$$\frac{2}{m}\left(\ln\left(\frac{2}{\Gamma_{\boldsymbol{z}}^2(h)}\right)^n + \ln\left(\frac{(em)^2}{\delta}\right)\right) \geq 2 \cdot \varepsilon_{\mathrm{Gibbs}}\left(m, \mathbf{U}_{\mathsf{H}}, \boldsymbol{z}, \delta\right) . \tag{1.6}$$

Here, n is the dimensionality of the feature space $\mathcal{K} \subseteq \ell_2^n$ in which the linear classification is carried out. The first term is the inverse of a lower bound on the volume of version space $V(\boldsymbol{z})$ in terms of a *normalised margin* $\Gamma_{\boldsymbol{z}}(h)$ given by

$$\Gamma_{\boldsymbol{z}}(h) \propto \min_{(x_i, y_i) \in \boldsymbol{z}} \frac{y_i f(x_i)}{\|x_i\|} , \tag{1.7}$$

which coincides with $\gamma_{\boldsymbol{z}}(h)$ for normalised data only. Thus we see that *whenever the SVM solution has a small generalisation error bound at least half of the consistent classifiers have the same (or even better) generalisation error bound.* The practical difficulty in exploiting these solutions, however, is that they keep changing over the random draw of the training sample and only the large margin classifier is able to *witness* its small generalisation error by an easy-to-determine quantity—its margin. Nonetheless, randomly drawing a consistent classifier will do as well in at least half of the learning trials *if the hypothesis space (model)* was suited for the task at hand. The result suggests one should not be too dismissive of algorithms such as the perceptron learning algorithm [24] which merely ensure one gets an $h \in V(\boldsymbol{z})$. It appears that the choice of the model $\mathcal{H}$ is more important than the choice of the learning procedure *within a fixed model* $\mathcal{H}$. For kernel based classifiers this means the choice of the kernel (see also Section 1.4).

1.3.2 From Margin To Sparsity—A Revival of the Perceptron

Theorem 1 tells us that whenever the training sample $\boldsymbol{z}$ observed and the hypothesis space $\mathcal{H}$ chosen lead to a large version space, there *exists* a large fraction of classifiers $h \in V(\boldsymbol{z})$ with a small generalisation error. In the special case of linear classifiers there is also an efficient algorithm for finding some of these classifiers — the perceptron algorithm [24]. In particular, we can prove the following theorem (see [5, 9] for more details).

Theorem 2 (Margin Bound). *For any measure P_Z, with probability at least $1-\delta$ over the random draw of the training set $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in (\mathcal{X} \times \{-1,+1\})^m$ of size m, if there exists a linear classifier $h^* \in \mathcal{H}$ such that*

$$\kappa^* = \left\lceil \frac{1}{\Gamma_{\boldsymbol{z}}^2 (h^*)} \right\rceil \leq m$$

then the generalisation error $R[h]$ of the classifier $h \in V(\boldsymbol{z})$ found by the perceptron algorithm is less than

$$\frac{1}{m-\kappa^*} \left(\ln \left(\binom{m}{\kappa^*} \right) + \ln (m) + \ln \left(\frac{1}{\delta} \right) \right) . \qquad (1.8)$$

Proof. The proof is a combination of a results of Novikoff [22] on the number of mistakes of the perceptron learning algorithm and a compression bound (see [3, 8, 16]). At first, Novikoff's theorem tells us that for normalised data $\boldsymbol{x} \in \mathcal{X}^m$ the perceptron learning algorithm is guaranteed to make at most κ^* mistakes. At each mistake, it adds (or subtracts) the current data point x_i to the weight vector which was initially set to $\mathbf{0}$. As a consequence thereof, the number of training samples (x_i, y_i) used to construct the final hypothesis is always less than or equal to κ^*. Since there are at most $\binom{m}{\kappa^*}$ different subsets of training samples of size κ^* the effective number of different hypotheses $h \in V(\boldsymbol{z})$ is this number. A combination of the binomial tail bound on the $m - \kappa^*$ left-out training points, i.e.

$$\forall h \in \mathcal{H}: \qquad \mathsf{P}_{\mathsf{Z}^{m-\kappa^*}} \left((h \notin V(\mathbf{Z})) \vee \left(R[h] \leq \frac{\ln\left(\frac{1}{\delta}\right)}{m-\kappa^*} \right) \right) \geq 1-\delta \,,$$

with the union bound over the number of different subsets proves the theorem. Note that the additional $\ln(m)$ term is due to the fact that the value of κ^* is not fixed. This requires us to share the confidence of $1-\delta$ among all its at most m different values.

Similar to the egalitarian bound this result is somewhat surprising as the generalisation error of the classifier learned by the perceptron learning algorithm is controlled by the potential margin $\Gamma_{\boldsymbol{z}}(h^*)$ a SVM *would have achieved* on the same training sample $\boldsymbol{z}$. Combining this result with the fact that margin bounds for support vector machines just witness the good choice of a model $\mathcal{H}$ (see (1.6)) we conclude that the simple perceptron algorithm is theoretically well justified because *whenever the SVM solution has a small generalisation error bound all the up to $m!$ different classifiers learned with the perceptron learning algorithm have the same (or even better) generalisation error bound.* This has also found some empirical evidence in the binary classification problems of handwritten digit recognition (see [4]).

1.3.3 Bayes Classification Strategy

Another consequence of Theorem 1 is that half of the classifiers within version space $V(\boldsymbol{z})$ have a generalisation error bound as good as that of the Bayes classification strategy. The Bayes classification strategy—also known as Bayesian transduction (see [7,30])—assigns a test example x to the class y by majority voting under the measure $\mathbf{P}_{\mathsf{H}|\mathsf{H}\in V(\boldsymbol{z})}$,

$$Bayes_{\boldsymbol{z}}(x) = \operatorname{argmax}_{y\in\mathcal{Y}} \mathbf{P}_{\mathsf{H}|\mathsf{H}\in V(\boldsymbol{z})}\left(\mathsf{H}(x) = y\right) .$$

In contrast to the Gibbs classification strategy, the Bayes classification strategy *deterministically* assigns a new test example to a class. For $|\mathcal{Y}| = 2$, whenever the Bayes classification strategy is wrong at x, at least half of the classifiers in version space misclassify x, too. By this argument, the generalisation error bound of the Bayes classification strategy fulfils

$$\forall \mathbf{P}_{\mathsf{H}}: \qquad \varepsilon_{\text{Bayes}}\left(m, \mathbf{P}_{\mathsf{H}}, \boldsymbol{z}, \delta\right) \leq 2 \cdot \varepsilon_{\text{Gibbs}}\left(m, \mathbf{P}_{\mathsf{H}}, \boldsymbol{z}, \delta\right) . \tag{1.9}$$

This equivalence of generalisation error bounds finds empirical support in [7,11]. Note that the “averaging” and “voting” feature of the Gibbs and Bayes strategies, respectively, safeguards them against domination by a minority of inferior members of the version space $V(\boldsymbol{z})$.

1.3.4 Have we Thrown the Baby out with the Bath Water?

At first glance the egalitarian bound seems to imply that we are hopeless in the search for *the* quantity controlling generalisation error (bounds) because it gives a good generalisation error bound for a huge number of consistent classifiers $h \in V(\boldsymbol{z})$ not referring to any property other than the choice of the model $\mathcal{H}$. This result, however, comes at no surprise taking into account what we investigated theoretically (see Definition 1). Although one is typically only interested in the performance of the one classifier h learned using a fixed learning algorithm $\mathcal{A} : \cup_{m=1}^{\infty} \mathcal{Z}^m \to \mathcal{H}$ traditional learning theory claims to need guarantees on the generalisation error that hold *uniformly* over the whole hypothesis space $\mathcal{H}$ or version space $V(\boldsymbol{z})$, respectively. This is much too demanding and can therefore only lead to bounds that indicate whether we have chosen an appropriate model or not. A much more promising approach seems to investigate the question of generalisation error bounds for specific algorithms. In fact, the proof of Theorem 2 uses a compression bound which requires the specification of the algorithm $\mathcal{A}$ in advance, i.e., the bounds apply only to a small subset of learning algorithms (so called *compression schemes*). A related idea is studied in [1] where the VC dimension as a complexity measure of an hypothesis space $\mathcal{H}$ is replaced by the *robustness* of the learning algorithm $\mathcal{A}$ used. The robustness of an algorithm $\mathcal{A}$ measures by how much the training error of the learned classifier $\mathcal{A}(\boldsymbol{z})$ is changing when adding one additional observation, i.e. $\max_{z=(x,y)} \left|R_{\text{emp}}\left[\mathcal{A}(\boldsymbol{z}), \boldsymbol{z}\right] - R_{\text{emp}}\left[\mathcal{A}(\boldsymbol{z} \cup z), \boldsymbol{z} \cup z\right]\right|$.

According to intuition, whenever a learning algorithm is very robust we have small deviation between generalisation and training error for the classifiers learned although the VC dimension of the hypothesis class used might have been infinite.

Finally, it is worthwhile noticing that this result does not deny the importance of *inductive principles*. Although we know that within a good model $\mathcal{H}$ there are many classifiers with a provably small generalisation error, there might exist procedures (the maximum margin algorithm is one such procedure) that single out classifiers with small generalisation error bounds for most random draws of the training sample $\boldsymbol{z}$. A potential candidate for formulating such inductive principles is the *luckiness framework* [26], which was recently extended to include an explicit dependency on the learning algorithm [12].

1.4 Experimental Results for Linear Classifiers

In order to complement the above theoretical analysis let us empirically evaluate the distribution of generalisation errors over version space members. Let us consider the hypothesis class $\mathcal{H}$ provided by linear classifiers in feature space $\mathcal{K} \subseteq \ell_2^n$ as used in SVMs. Each hypothesis is given by

$$h_{\mathbf{w}}(x) = \operatorname{sign}(\langle \boldsymbol{\phi}(x), \mathbf{w} \rangle) = \operatorname{sign}(\langle \mathbf{x}, \mathbf{w} \rangle) , \tag{1.10}$$

where $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{K} \subseteq \ell_2^n$ is a mapping[3] from the input space $\mathcal{X}$ to the feature space $\mathcal{K}$. Note that it is sufficient to consider weight vectors $\mathbf{w} \in \mathcal{K}$ of unit length, i.e. $\mathbf{w} \in \mathcal{W}$, $\mathcal{W} = \{\mathbf{w} \in \mathcal{K} \mid \|\mathbf{w}\| = 1\}$, because for any positive constant

$$\forall \lambda > 0 : \qquad h_{\mathbf{w}} = \operatorname{sign}(\langle \mathbf{x}, \mathbf{w} \rangle) = \operatorname{sign}(\langle \mathbf{x}, \lambda\mathbf{w} \rangle) = h_{\lambda\mathbf{w}} .$$

Ergo, the hypothesis space $\mathcal{H}$ is isomorphic to the unit sphere $\mathcal{W} \subset \ell_2^n$ (see also Figure 1.1). If the objective function optimised by the learning algorithm depends only on the inner products of the weight vector $\mathbf{w}$ with all the mapped training points it can be shown that it is sufficient to consider normal vectors $\mathbf{w} \in \mathcal{W}$ that are linearly expandable in the training points [15, 25],

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i .$$

As a consequence, each hypothesis h can be written in terms of $\boldsymbol{\alpha} \in \mathbb{R}^m$, i.e.

$$h_{\boldsymbol{\alpha}}(x) = \operatorname{sign}\left(\sum_{i=1}^{m} \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle\right) = \operatorname{sign}\left(\sum_{i=1}^{m} \alpha_i k(x_i, x)\right) ,$$

where the inner product function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is also known as the *kernel* (see, e.g. [29]). In practical application, it is often more convenient to select the kernel than the feature mapping $\boldsymbol{\phi}$.

[3] We abbreviate (x) by $\mathbf{x}$ always assuming to be fixed. This, however, should not be confused with the training sample $\boldsymbol{x} \in \mathcal{X}^m$.

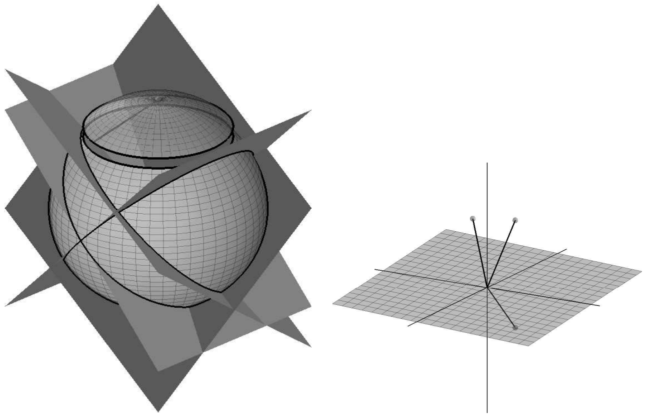


Fig. 1.1. (Left) The hypothesis space $\mathcal{H}$ of linear classifiers for a 3–dimensional feature space $\mathcal{K}$. Each point on the unit sphere is the weight vector $\mathbf{w} \in \mathcal{W}$ of a linear classifier $h_{\mathbf{w}}$ (see (1.10)). The convex polyhedron on top is a version space $V(\boldsymbol{z})$; the length of the gray line is proportional to the normalised margin $\Gamma_{\boldsymbol{z}}(h_{\mathbf{w}})$ of the classifier on top of the sphere. **(Right)** Three data points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ in a 3–dimensional feature space $\mathcal{K} \subseteq \ell_2^3$. Note that the planes in the left picture are incurred by each of the three training points by $\{\mathbf{w} \in \mathcal{K} \mid \langle \mathbf{x}, \mathbf{w} \rangle = 0\}$. Using exactly the same rule, each point $\mathbf{w} \in \mathcal{W}$ on the unit sphere in the left picture induces a decision plane $\{\mathbf{x} \in \mathcal{K} \mid \langle \mathbf{x}, \mathbf{w} \rangle = 0\}$ in feature space.

1.4.1 The Kernel Gibbs Sampler

In order to sample consistent classifiers uniformly from $V(\boldsymbol{z})$ we suggest a Markov Chain sampling method known as the *kernel Gibbs*[4] *sampler* [6]. It is a variant of the well-known hit-and-run sampling algorithm [27], which was recently shown to exhibit a fast mixing time of $\mathcal{O}(n^3)$, where n is the dimensionality of the space [17]. The kernel Gibbs sampler is applicable whenever $\mathbf{P}_{\mathsf{H}|\mathsf{Z}^m=\boldsymbol{z}}$ is a piecewise constant density proportional to

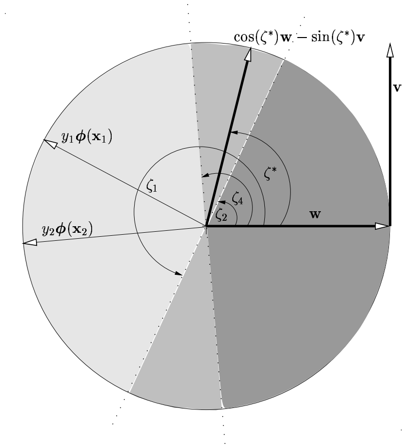
$$\mathcal{L}[h,\boldsymbol{z}] = \theta^{m\cdot R_{\mathrm{emp}}[h,\boldsymbol{z}]}(1-\theta)^{m(1-R_{\mathrm{emp}}[h,\boldsymbol{z}])}\ , \quad \text{for some } \theta \in [0,1]\ . \tag{1.11}$$

Note that this density arises from a Bayesian consideration of learning when assuming that the classification is corrupted by label noise of level $\theta \in [0,1]$, i.e.

$$\mathbf{P}_{\mathsf{Y}|\mathsf{X}=x,\mathsf{H}=h}(y) = \theta \cdot \mathbf{I}_{y\neq h(x)} + (1-\theta)\,\mathbf{I}_{y=h(x)}\ . \tag{1.12}$$

For a given value of the noise level θ and an arbitrary starting point $\mathbf{w}_0 \in \mathcal{W}$, the sampling scheme can be decomposed into the following steps (see also Figure 1.2):

[4] This should not be confused with the *Gibbs classification strategy.*


$\cos(\zeta^*)\mathbf{w} - \sin(\zeta^*)\mathbf{v}$
$\mathbf{v}$
$y_1\boldsymbol{\phi}(\mathbf{x}_1)$
ζ_1
ζ^*
ζ_4
ζ_2
$\mathbf{w}$
$y_2\boldsymbol{\phi}(\mathbf{x}_2)$

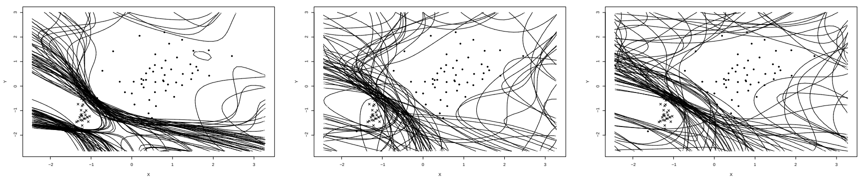
Two data points $y_1\mathbf{x}_1$ and $y_2\mathbf{x}_2$ divide the space of normalised weight vectors $\mathbf{w} \in \mathcal{W}$ into four equivalence classes with different posterior density indicated by the gray shading. In each iteration, starting from $\mathbf{w}_{j-1}$ a random direction $\mathbf{v}$ with $\mathbf{v}\perp\mathbf{w}_{j-1}$ is generated. We sample from the piecewise constant density on the great circle determined by the plane defined by $\mathbf{w}_{j-1}$ and $\mathbf{v}$. In order to obtain ζ^*, we calculate the $2m$ angles ζ_i where the training samples intersect with the circle and keep track of the number $m \cdot e_i$ of training errors for each region i.

Fig. 1.2. Schematic view of the kernel Gibbs sampling procedure.

1. Choose a direction $\mathbf{v} \in \mathcal{W}$ in the tangent space $\{\tilde{\mathbf{v}} \in \mathcal{W} \mid \langle \tilde{\mathbf{v}}, \mathbf{w}_j \rangle = 0\}$.
2. Calculate all m hit points $\mathbf{b}_i \in \mathcal{W}$ from $\mathbf{w}$ in direction $\mathbf{v}$ with the hyperplane having normal $y_i\mathbf{x}_i$. Before normalisation, this is achieved by [11]
$$\mathbf{b}_i = \mathbf{w}_j - \frac{\langle \mathbf{w}_j, \mathbf{x}_i \rangle}{\langle \mathbf{v}, \mathbf{x}_i \rangle}\mathbf{v}\,.$$
3. Calculate the $2m$ angular distances ζ_i from the current position $\mathbf{w}_j$.
4. Sort the ζ_i in ascending order (resulting in a permutation $\Pi : \{1, \ldots, 2m\} \to \{1, \ldots, 2m\}$ and calculate the training errors $e_i = R_{\text{emp}}\left[h_{\mathbf{m}_i}, \boldsymbol{z}\right]$ of the $2m$ intervals $\left[\zeta_{\Pi(i-1)}, \zeta_{\Pi(i)}\right]$ by evaluating
$$\mathbf{m}_i = \cos\left(\frac{\zeta_{\Pi(i+1)} - \zeta_{\Pi(i)}}{2}\right)\mathbf{w}_j - \sin\left(\frac{\zeta_{\Pi(i+1)} - \zeta_{\Pi(i)}}{2}\right)\mathbf{v}\,.$$
Here, we have defined $\zeta_{\Pi(2m+1)} = \zeta_{\Pi(1)}$.
5. Sample an angle ζ^* using the piecewise uniform distribution and (1.11). Calculate a new sample $\mathbf{w}_{j+1}$ by $\mathbf{w}_{j+1} = \cos(\zeta^*)\,\mathbf{w}_j - \sin(\zeta^*)\,\mathbf{v}$.
6. Set $j \leftarrow j+1$ and go back to step 1.

Since the algorithm is carried out in feature space $\mathcal{K}$ we use
$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i\,, \quad \mathbf{v} = \sum_{i=1}^{m} \nu_i \mathbf{x}_i\,, \quad \mathbf{b} = \sum_{i=1}^{m} \beta_i \mathbf{x}_i\,.$$

For the inner products and norms it follows that $\langle \mathbf{w}, \mathbf{v} \rangle = \boldsymbol{\alpha}'\mathbf{G}\boldsymbol{\nu}$, $\|\mathbf{w}\|^2 = \boldsymbol{\alpha}'\mathbf{G}\boldsymbol{\alpha}$, where the $m \times m$ matrix $\mathbf{G}$ is known as the *kernel* or *Gram matrix* and is given by $\mathbf{G}_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = k(x_i, x_j)$. In Figure 1.3 we have shown



$\theta = 0.0$ $\theta = 0.1$ $\theta = 0.2$

Fig. 1.3. A set of 50 samples $\mathbf{w}_j$ for various noise levels θ. Shown are the resulting decision boundaries in input space $\mathcal{X} = \mathbb{R}^2$.

an application of the kernel Gibbs sampler to some toy data in $\mathbb{R}^2$. As can be seen from these plots, increasing the noise level θ leads to more diverse classifiers on the training sample $\boldsymbol{z}$. In the following we will fix the noise level θ to zero in order to sample version space classifiers only. Other applications of this sampling algorithm are active learning, transduction and confidence estimation with kernel classifiers.

1.4.2 Distribution of Generalisation Errors and Margins

Based on the MNIST dataset[5] for images of “1” and “2” we generated well-balanced training and test samples of size 118 and 453, respectively. In order to explore the structure of version space we were interested in the distribution of generalisation errors (estimated on the given test sample) *and* its relation to the attained margin $\mathit{\Gamma}_{\boldsymbol{z}}(h)$. In Figure 1.4 (left) we plotted the distribution of generalisation errors for $l = 10000$ samples $\mathbf{w}$ using different degrees of the polynomial kernel

$$k(x_i, x_j) = \left(\langle x_i, x_j \rangle_{\mathcal{X}} + 1\right)^p \, , \tag{1.13}$$

which produced excellent classifiers when used in SVM learning ($p_{\text{opt}} = 5$). In order to reduce dependencies between successive samples $\mathbf{w}$ of the Markov chain we used only one in ten samples thus effectively having $l_{\text{eff}} = 1000$ samples. For any value of p considered there are at least 50% of consistent classifiers whose generalisation error is smaller than the one found by the SVM ($\triangle$) in accordance with (1.6) and the egalitarian bound of Theorem 1. Surprisingly, with increasing polynomial degree p the variance of the distribution keeps decreasing while only a small increase of its mean can be observed beyond degree 5. Furthermore, using the Bayes point machine algorithm that returns the “centre of mass” of version space $V(\boldsymbol{z})$ (see [11]) or the SVM on the normalised training sample in feature space $\mathcal{K}$ we seem to be able to find classifiers always within the best 50% ($\circ$ and $\times$). Both these algorithms aim at finding a solution at the “centre” of version space $V(\boldsymbol{z})$ in the sense of $\mathit{\Gamma}_{\boldsymbol{z}}$ (see (1.7)).

[5] publicly available at `http://www.research.att.com/~yann/ocr/mnist/`.

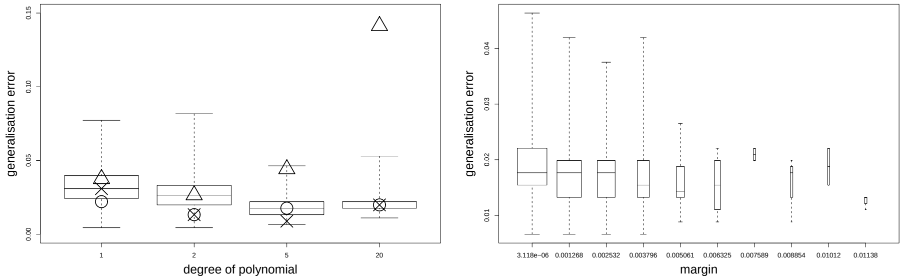


Fig. 1.4. **(Left)** Box-plots of distributions of generalisation errors for $l = 1000$ samples $\mathbf{w}$ using different degrees in the polynomial kernel (1.13). The $\triangle$, $\times$ and $\circ$ depict the generalisation errors of the SVM solution, the SVM solution when normalising in feature space $\mathcal{K}$ and the Bayes point machine solution (see text), respectively. **(Right)** Box-plots of distributions of generalisation for different attained margins (1.7) when using a polynomial kernel of degree 5. The width of each box-plot is proportional to the number of samples on which it is based.

In Figure 1.4 (right) we additionally provide the distributions of generalisation error for given attained margins $\Gamma_{\boldsymbol{z}}(h)$. As expected, *almost all of the classifiers h with a large margin $\Gamma_{\boldsymbol{z}}(h)$ do have a small generalisation error $R[h]$*. The plot also clarifies that large margins are only (probabilistically) a *sufficient condition* for good generalisation ability and that there exist many consistent classifiers with good generalisation error despite of their small margins. This is again in accordance with the egalitarian bound of Theorem 1 keeping in mind that in high-dimensional feature spaces $\mathcal{K}$ the uniform measure over volumes is concentrated near the edges. Hence, most of the classifiers in version space $V(\boldsymbol{z})$ *do have* a small margin (see width of the box-plots) albeit exhibiting good generalisation.

1.5 Conclusion

The notion of version space plays a crucial rule both in the theoretical analysis of learning algorithms and in their practical implementation. We have presented a theorem which shows that within a wisely chosen hypothesis space many consistent classifiers show good generalisation irrespective of the maximisation of a pre-specified complexity measure (luckiness) such as margin. Our empirical results strongly support this conclusion and give an intuition for the structure of version space.

While the restriction to zero training-error classifiers may appear to be severe at first glance, for linear classifiers this limitation is easily overcome by modifying the kernel as follows:

$$k_{\lambda}\left(x_i, x_j\right) = k\left(x_i, x_j\right) + \lambda \mathbf{I}_{x_i = x_j} \,.$$

This trick—well known in SVMs as the quadratic soft-margin technique [2]—gradually (with increasing λ) decouples the training examples $\boldsymbol{\phi}\left(x_i\right)$ for learning and thus serves to create a version space even if the training examples were not separable under the original kernel k. Furthermore, it is straightforward to exploit Theorem 2 of [19] so as to generalise the egalitarian bound to *any* subset H of hypothesis space $\mathcal{H}$. The difference to the present result is that in this case for many classifiers the generalisation error is effectively bounded by the training error plus the penalty $-\ln\left(\mathbf{P}_{\mathsf{H}}\left(H\right)\right)$. In case most of the classifiers in hypothesis space exhibit a small training error $\left(\mathbf{P}_{\mathsf{H}}\left(H\right) \approx 1\right)$ we see that we get a conceptually similar result to Theorem 1. Hence, our results also cover certain cases of inconsistent classifiers deemed so important in practice.

It is worthwhile mentioning that a consequence of the above mentioned generalisation of Theorem 1 is that with high probability over the random draw of the training sample for many classifiers in *hypothesis space* the deviation between generalisation and training error is small. This result holds regardless of the VC dimension of hypothesis space $\mathcal{H}$ used. The challenge is to find generalisation error bounds that indicate if this result also holds for the *single* classifier we learned from the observed training sample.

Acknowledgements

We would like to thank Olivier Bousquet and Matthias Seeger for careful proofreading and many useful suggestions. Furthermore, we are greatly indebted to John Shawe-Taylor, Peter Bartlett, Jonathan Baxter and Martin Anthony for helpful discussions and comments. This work was supported by the Australian Research Council.

References

1. O. Bousquet and A. Elisseeff. Algorithmic stability and generalization performance. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 196–202. MIT Press, 2001.
2. C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.
3. S. Floyd and M. Warmuth. Sample compression, learnability, and the Vapnik Chervonenkis dimension. *Machine Learning*, 27:1–36, 1995.
4. Y. Freund. An adaptive version of the boost by majority algorithm. In *Proceedings of the Annual Conference on Computational Learning Theory*, 1999.
5. Y. Gat. A learning generalization bound with an application to sparse-representation classifiers. *Machine Learning*, 42(3):233–240, 2001.

6. T. Graepel and R. Herbrich. The kernel Gibbs sampler. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 514–520, Cambridge, MA, 2001. MIT Press.
7. T. Graepel, R. Herbrich, and K. Obermayer. Bayesian Transduction. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 456–462, Cambridge, MA, 2000. MIT Press.
8. T. Graepel, R. Herbrich, and J. Shawe-Taylor. Generalisation error bounds for sparse linear classifiers. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 298–303, 2000.
9. T. Graepel, R. Herbrich, and R. C. Williamson. From margin to sparsity. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 210–216, Cambridge, MA, 2001. MIT Press.
10. R. Herbrich and T. Graepel. A PAC-Bayesian margin bound for linear classifiers. *IEEE Transactions on Information Theory*, 2002.
11. R. Herbrich, T. Graepel, and C. Campbell. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279, 2001.
12. R. Herbrich and R. C. Williamson. Algorithmic luckiness. *Journal of Machine Learning Research*, 3:175–212, 2002.
13. M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.
14. S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. Technical Report Technical Report TR-ISL-99-03, Indian Institute of Science, Bangalore, 1999.
15. G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41:495–502, 1970.
16. N. Littlestone and M. Warmuth. Relating data compression and learnability. Technical report, University of California Santa Cruz, 1986.
17. L. Lovasz. Hit-And-Run mixes fast. *Mathematical Programming A*, 86:443–461, 1999.
18. D. J. C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.
19. D. A. McAllester. Some PAC Bayesian theorems. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 230–234, Madison, Wisconsin, 1998. ACM Press.
20. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.
21. T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):202–226, 1982.
22. A. B. J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622. Polytechnic Institute of Brooklyn, 1962.
23. J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods—Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

24. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
25. B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 416–426, 2001.
26. J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.
27. R. L. Smith. Efficient Monte-Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32:1296–1308, 1984.
28. A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In P. Langley, editor, *Proceedings of the International Conference on Machine Learning*, pages 911–918, San Francisco, 2000. Morgan Kaufmann Publishers.
29. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
30. V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, Berlin, 1982.