PAC-Bayesian Compression Bounds on the Prediction Error of Learning Algorithms for Classification

Thore Graepel (graepel@inf.ethz.ch)
Institute of Computational Science, ETH Zürich, Switzerland

Ralf Herbrich (rherb@microsoft.com)
Microsoft Research, 7 J J Thomson Avenue, CB3 0FB Cambridge, UK

John Shawe-Taylor (jst@cs.rhul.ac.uk)
Department of Computer Science, Royal Holloway, University of London, UK

Abstract.

We consider bounds on the prediction error of classification algorithms based on sample compression. We refine the notion of a compression scheme to distinguish permutation and repetition invariant and non-permutation and repetition invariant compression schemes leading to different prediction error bounds. Also, we extend known results on compression to the case of non-zero empirical risk.

We provide bounds on the prediction error of classifiers returned by mistake-driven online learning algorithms by interpreting mistake bounds as bounds on the size of the respective compression scheme of the algorithm. This leads to a bound on the prediction error of perceptron solutions that depends on the margin a support vector machine would achieve on the same training sample.

Furthermore, using the property of compression we derive bounds on the average prediction error of kernel classifiers in the PAC-Bayesian framework. These bounds assume a prior measure over the expansion coefficients in the data-dependent kernel expansion and bound the average prediction error uniformly over subsets of the space of expansion coefficients.

1. Introduction

Generalization error bounds based on sample compression are a great example of the intimate relationship between information theory and learning theory. The general relation between compression and prediction has been expressed in different contexts such as Kolmogorov complexity (Vitànyi and Li, 1997), minimum description length (Rissanen, 1978), and information theory (Wyner et al., 1992). As was first pointed out by Littlestone and Warmuth (1986) and later by Floyd and Warmuth (1995), the prediction error of a classifier h can be bounded in terms of the number d of examples to which a training sample of size m can be compressed while still preserving the information necessary for the learning algorithm to identify the classifier h. Intuitively speaking, the remaining $m - d$ examples that are not required for training serve as a test sample on which the classifier is evaluated. Interestingly, the

compression bounds so derived are among the best bounds in existence in the sense that they return low values even for moderately large training sample size. As a consequence, compression arguments have been put forward as a justification for a number of learning algorithms including the support vector machine (Cortes and Vapnik, 1995) whose solution can be reproduced based on the *support vectors,* that constitute a subset of the training sample.

Prediction error bounds based on compression stand in contrast to classical PAC/VC bounds in the sense that PAC/VC bounds assume the existence of a fixed hypothesis space $\mathcal{H}$ (see Cannon et al. (2002) for a relaxation of this assumption) while compression results are independent of this assumption and typically work well for algorithms based on a hypothesis space of infinite VC dimension or even based on a data-dependent hypothesis space, as is the case, for example, in the support vector machine. We systematically review the notion of compression as introduced in Littlestone and Warmuth (1986) and Floyd and Warmuth (1995). In Section 3 we refine the idea of a compression scheme to distinguish between *permutation and repetition invariant* and *non permutation and repetition invariant* compression schemes, leading to different prediction error bounds. Moreover, we extend the known compression results for the zero-error training case to the case of non-zero training error. Note that the results of both Littlestone and Warmuth (1986) and Floyd and Warmuth (1995) implicitly contained this agnostic bound via the notion of side information.

We then review the relation between batch and online learning, which has been a recurrent theme in learning theory (see Littlestone (1989) and Cesa-Bianchi et al. (2002)). The results in Section 4 are based on an interesting relation between online learning and compression: Mistake-driven online learning algorithm constitute non permutation invariant compression schemes. We exploit this fact to obtain PAC type bounds on the prediction error of classifiers resulting from mistake-driven online learning using mistake bounds as bounds on the size d of compression schemes. In particular, we will reconsider the perceptron algorithm and derive a PAC bound for the resulting classifiers from a mistake bound involving the margin a support vector machine would achieve on the same training data. This result went so far largely unnoticed in the study of margin bounds.

Similarly to PAC/VC results, recent bounds in the PAC-Bayesian framework (Shawe-Taylor and Williamson, 1997; McAllester, 1998) assume the existence of a fixed hypothesis space $\mathcal{H}$. Given a prior measure $\mathbf{P}_{\mathsf{H}}$ over $\mathcal{H}$ the PAC-Bayesian framework then provides bounds on the average prediction error of classifiers drawn from a posterior $\mathbf{P}_{\mathsf{H}|\mathbf{Z}=\boldsymbol{z}}$ in terms of the average training error and the KL divergence between prior

and posterior (McAllester, 1999). Interestingly, tight margin bounds for linear classifiers were proved in the PAC-Bayesian framework in Graepel et al. (2000), Herbrich and Graepel (2002) and Langford and Shawe-Taylor (2003). Heavily borrowing from ideas in the compression framework, in Section 5 we prove general PAC-Bayesian results for the case of sparse data-dependent hypothesis spaces such as the class of kernel classifiers on which the support vector machine is based. Instead of assuming a prior $\mathbf{P_H}$ over hypothesis space, we assume a prior $\mathbf{P_A}$ over the space of coefficients in the kernel expansion. As a result, we obtain PAC-Bayesian results on the average prediction error of data-dependent hypotheses.

2. Basic Learning Task and Notation

We consider the problem of binary classification learning, that is, we aim at modeling the underlying dependency between two sets referred to as input space $\mathcal{X}$ and output space $\mathcal{Y}$, which will be jointly referred to as the input-output space $\mathcal{Z}$ according to the following definition:

Definition 1 (Input-Output space). We call

1. $\mathcal{X}$ the *input space*,
2. $\mathcal{Y} := \{-1, +1\}$ the *output space*, and
3. $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ the joint *input-output space*

of the binary classification learning problem.

Learning is based on a training sample $\boldsymbol{z}$ of size m defined as follows:

Definition 2 (Training sample). Given an input-output space $\mathcal{Z}$ and a probability measure $\mathbf{P_Z}$ thereon we call an m-tuple $\boldsymbol{z} \in \mathcal{Z}^m$ drawn IID from $\mathbf{P_Z} := \mathbf{P_{XY}}$ a *training sample* of size m. Given $\boldsymbol{z} = ((x_1, y_1), \ldots, (x_m, y_m))$ we will call the pairs (x_i, y_i) training examples. Also we use the notation $\boldsymbol{x} = (x_1, \ldots, x_m)$ and similarly $\boldsymbol{y} = (y_1, \ldots, y_m)$.

The hypotheses considered in learning are contained in the hypothesis space.

Definition 3 (Hypothesis and hypothesis space). Given an input space $\mathcal{X}$ and an output space $\mathcal{Y}$ we define a *hypothesis* as a function

$$h : \mathcal{X} \to \mathcal{Y}\,,$$

and a *hypothesis space* as a subset

$$\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}} .$$

A hypothesis space is called a *data-dependent hypothesis space* if the set of hypotheses can only be defined for a given training sample and may change with varying training samples.

A learning algorithm takes a training sample and returns a hypothesis according to the following definition:

Definition 4 (Learning algorithm). Given an input space $\mathcal{X}$ and an output space $\mathcal{Y}$ we call a mapping[1]

$$\mathcal{A} : \mathcal{Z}^{(\infty)} \to \mathcal{Y}^{\mathcal{X}}$$

a *learning algorithm*.

In order to assess the quality of solutions to the learning problem, we use the zero-one loss function.

Definition 5 (Loss function). Given an output space $\mathcal{Y}$ we call a function

$$l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{+}$$

a *loss function* on $\mathcal{Y}$ and we define the *zero-one loss function* as

$$l_{0-1}\left(\hat{y}, y\right) := \begin{cases} 0 & \text{for } \hat{y} = y \\ 1 & \text{for } \hat{y} \neq y \end{cases} .$$

Note that this can also be written as $l_{0-1}\left(\hat{y}, y\right) = \mathbb{I}_{\hat{y} \neq y}$, where $\mathbb{I}$ is the indicator function.

A useful measure of success of a given hypothesis h based on a given loss function l is its (true) risk defined as follows:

Definition 6 (True risk). Given a loss function l, a hypothesis space $\mathcal{H}$, and a probability measure $\mathbf{P}_{\mathsf{Z}}$ the functional $R : \mathcal{H} \to \mathbb{R}$ given by

$$R\left[h\right] := \mathbf{E}_{\mathsf{XY}}\left[l\left(h\left(\mathsf{X}\right), \mathsf{Y}\right)\right] ,$$

that is, the expectation of the loss, is called the (true) *risk* on $\mathcal{H}$. Given a hypothesis h we also call $R\left[h\right]$ its *prediction error*. For the zero-one loss l_{0-1} the risk is equal to the probability of error.

The true risk or its average over a subset of hypotheses will be our main quantity of interest. A useful estimator for the true risk is its plug-in estimator, the empirical risk.

[1] Throughout the paper we use the shorthand notation $A^{(i)} := \cup_{j=1}^{i} A^{j}$.

Definition 7 (Empirical risk). Given a training sample $\boldsymbol{z} \sim \mathbf{P}_{\mathsf{Z}^m}$, a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, and an hypothesis $h \in \mathcal{H}$ we call

$$\hat{R}\left[h, \boldsymbol{z}\right] := \frac{1}{|\boldsymbol{z}|} \sum_{i=1}^{|\boldsymbol{z}|} l\left(h\left(x_i\right), y_i\right)$$

the *empirical risk* of h on $\boldsymbol{z}$. An hypothesis h with $\hat{R}\left[h, \boldsymbol{z}\right] = 0$ is called *consistent* with $\boldsymbol{z}$.

Given these preliminaries we are now in a position to consider bounds on the true risk of classifiers based on the property of sample compression.

3. PAC Compression Bounds

In order to relate our new results to the body of existing work, we will review the unpublished work of Littlestone and Warmuth (1986) and the seminal paper Floyd and Warmuth (1995). In addition to these two papers, our introduction of compression schemes carefully distinguishes between permutation and repetition invariance since it leads to different bounds on the prediction error. This distinction will become important when studying online algorithms in Section 4.

3.1. Compression and Reconstruction

In order to be able to bound the prediction error of classifiers in terms of their sample compression it is necessary to consider particular learning algorithms instead of particular hypothesis spaces. In contrast to classical results that constitute bounds on the prediction error which hold uniformly over all hypotheses in $\mathcal{H}$ (PAC/VC framework) or which hold uniformly over all subsets of $\mathcal{H}$ (PAC-Bayesian framework) we are in the following concerned with bounds on the prediction error which hold only for those classifiers that result from particular learning algorithms (see Definition 4). Let us decompose a learning algorithm $\mathcal{A}$ into a compression scheme as follows (Littlestone and Warmuth, 1986).

Theorem 1 (Compression scheme). *We define the set* $I_{d,m} \subset \{1, \ldots, m\}^d$ *as the set containing all index vectors of size* $d \in \mathbb{N}$*,*

$$I_{d,m} := \left\{(i_1, \ldots, i_d) \in \{1, \ldots, m\}^d\right\} .$$

Given a training sample $\boldsymbol{z} \in \mathcal{Z}^m$ and an index vector $\mathbf{i} \in I_{d,m}$ let $\boldsymbol{z}_\mathbf{i}$ be the subsequence indexed by $\mathbf{i}$,

$$\boldsymbol{z}_\mathbf{i} := \left(z_{i_1}, \ldots, z_{i_d}\right) .$$

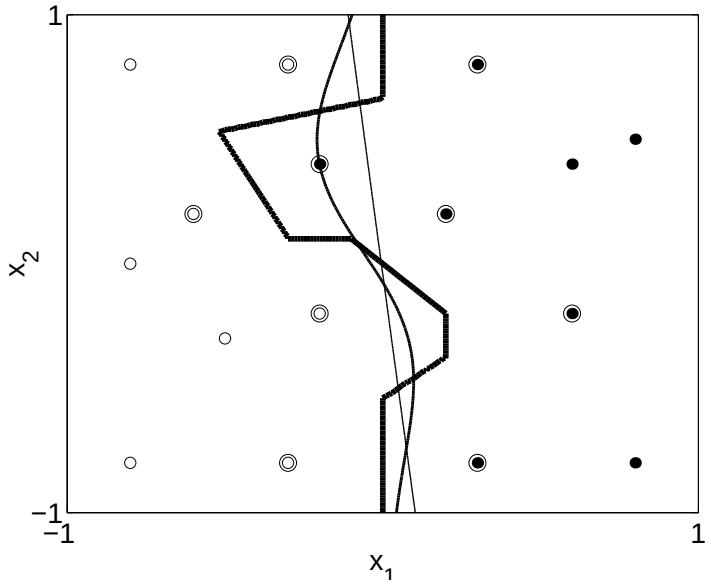

Figure 1. Illustration of the convergence of the kernel classifier based on class-conditional Parzen window density estimation to the nearest neighbor classifier in $\mathcal{X} = [-1, +1]^2 \subset \mathbb{R}^2$. For $\sigma = 5$ the decision surface (thin line) is almost linear, for $\sigma = 0.4$ the curved line (medium line) results, and for very small $\sigma = 0.02$ the piecewise linear decision surface (thick line) of nearest neighbor results. For nearest neighbor only the circled points contribute to the decision surface and form the compression sample.

We call an algorithm $\mathcal{A} : \mathcal{Z}^{(\infty)} \to \mathcal{H}$ a *compression scheme* if and only if there exists a pair $(\mathcal{C}, \mathcal{R})$ of functions $\mathcal{C} : \mathcal{Z}^{(\infty)} \to \bigcup_{m=1}^{\infty} \bigcup_{d=1}^{m} I_{d,m}$ (compression function) and $\mathcal{R} : \mathcal{Z}^{(\infty)} \to \mathcal{H}$ (reconstruction function) such that we have for all training samples $\boldsymbol{z}$,

$$\mathcal{A}(\boldsymbol{z}) = \mathcal{R}\left(\boldsymbol{z}_{\mathcal{C}(\boldsymbol{z})}\right) .$$

We call the compression scheme *permutation and repetition invariant* if and only if the reconstruction function $\mathcal{R}$ is invariant under permutation and repetition of training examples in any training sample $\boldsymbol{z}$. The quantity $|\mathcal{C}(\boldsymbol{z})|$ is called the *size of the compression scheme*.

The definition of a compression scheme is easily illustrated by three well-known algorithms: the perceptron, the support vector machine (SVM), and the K-nearest-neighbors (KNN) classifier, which are all based on the data-dependent hypothesis space of kernel classifiers.

Definition 8 (Kernel classifiers). Given a training sample $\boldsymbol{z} \in (\mathcal{X} \times \mathcal{Y})^m$ and a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ we define the data-dependent hypothesis $\mathcal{H}_k(\boldsymbol{x})$ by

$$\mathcal{H}_k(\boldsymbol{x}) := \left\{ x \mapsto \operatorname{sign}\left(\sum_{i=1}^{m} \alpha_i k(x_i, x) \right) \middle| \; \boldsymbol{\alpha} \in \mathbb{R}^m \right\}. \tag{1}$$

1. The (kernel) perceptron algorithm (Rosenblatt, 1962) is a compression scheme that is not permutation and repetition invariant. Rerunning the perceptron algorithm on a training sample that consists only of those training examples that caused an update in the previous run leads to the same classifier as before. Permuting the order of the examples or omitting repeated examples, however, may lead to a different classifier.

2. The support vector machine (Cortes and Vapnik, 1995) is a permutation and repetition invariant compression scheme. Rerunning the SVM only on the support vectors leads to the same classifier regardless of their order because the expansion coefficients in the optimal solution of the other training examples are zero and the objective function is invariant under permutation of the training examples.

3. The K-nearest-neighbors classifier (Cover and Hart, 1967) can be viewed as a limiting case of kernel classifiers and can be viewed as a permutation and repetition invariant compression scheme as well: Delete those training examples that do not change the majority on any conceivable test input $x \in \mathcal{X}$ (consider Figure 3.1 for an illustration for the case of $K = 1$).

Note that mere sparsity in the expansion coefficients α_i in (1) is not sufficient for an algorithm to qualify as a compression scheme, but it is necessary that the hypothesis found can be reconstructed from the compression sample. The relevance vector machine algorithm presented in Tipping (2001) is an example of an algorithm that does provide solutions that are sparse in the expansion coefficients α_i without constituting a compression scheme. Based on the concept of compression let us consider PAC-style bounds on the prediction error of learning algorithms as described above.

3.2. The Realizable Case

Let us first consider the realizable learning scenario, i.e. for every training sample $\boldsymbol{z}$ there exists a classifier h such that $\hat{R}[h, \boldsymbol{z}] = 0$. Then we have the following compression bound (note that (2) was already proven in Littlestone and Warmuth (1986) but will be repeated here for comparison).

Theorem 2 (PAC compression bound). *Let $\mathcal{A} : \mathcal{Z}^{(\infty)} \to \mathcal{H}$ be a compression scheme. For any probability measure $\mathbf{P}_{\mathsf{Z}}$, any $m \in \mathbb{N}$, and any $\delta \in (0, 1]$, with probability at least $1 - \delta$ over the random draw of the training sample $\boldsymbol{z} \in \mathcal{Z}^m$, if $\hat{R}[\mathcal{A}(\boldsymbol{z}), \boldsymbol{z}] = 0$ and $d := |\mathcal{C}(\boldsymbol{z})|$ then*

$$R[\mathcal{A}(\boldsymbol{z})] \leq \frac{1}{m-d}\left(\log\left(m^d\right) + \log(m) + \log\left(\frac{1}{\delta}\right)\right),$$

and, if $\mathcal{A}$ is a permutation and repetition invariant compression scheme, then

$$R[\mathcal{A}(\boldsymbol{z})] \leq \frac{1}{m-d}\left(\log\binom{m}{d} + \log(m) + \log\left(\frac{1}{\delta}\right)\right). \tag{2}$$

Proof. First we bound the probability

$$\begin{aligned}
&\mathbf{P}_{\mathsf{Z}^m}\left(\hat{R}[\mathcal{A}(\mathbf{Z}), \mathbf{Z}] = 0 \ \wedge \ R[\mathcal{A}(\mathbf{Z})] > \varepsilon \ \wedge \ |\mathcal{C}(\mathbf{Z})| = d\right) \\
&\quad \leq \mathbf{P}_{\mathsf{Z}^m}\left(\exists \mathbf{i} \in I_{d,m} : \left(\hat{R}[\mathcal{R}(\mathbf{Z}_{\mathbf{i}}), \mathbf{Z}] = 0 \ \wedge \ R[\mathcal{R}(\mathbf{Z}_{\mathbf{i}})] > \varepsilon\right)\right) \\
&\quad \leq \sum_{\mathbf{i} \in I_{d,m}} \mathbf{P}_{\mathsf{Z}^m}\left(\hat{R}[\mathcal{R}(\mathbf{Z}_{\mathbf{i}}), \mathbf{Z}] = 0 \ \wedge \ R[\mathcal{R}(\mathbf{Z}_{\mathbf{i}})] > \varepsilon\right).
\end{aligned} \tag{3}$$

The second line follows from the property $\mathcal{A}(\boldsymbol{z}) = \mathcal{R}\left(\boldsymbol{z}_{\mathcal{C}(\boldsymbol{z})}\right)$ and the fact that the event in the second line is implied by the event of the first line. The third line follows from the union bound, Lemma 1 in Appendix A. Each summand in (3)—being a product measure—is further bounded by

$$\mathbf{E}_{\mathsf{Z}^d}\left[\mathbf{P}_{\mathsf{Z}^{m-d}|\mathsf{Z}^d = \boldsymbol{z}_{\mathbf{i}}}\left(\hat{R}[\mathcal{R}(\boldsymbol{z}_{\mathbf{i}}), \mathbf{Z}] = 0 \ \wedge \ R[\mathcal{R}(\boldsymbol{z}_{\mathbf{i}})] > \varepsilon\right)\right] \tag{4}$$

where we used the fact that correct classification of the whole training sample $\boldsymbol{z}$ implies correct classification of any subset $\tilde{\boldsymbol{z}} \subseteq \boldsymbol{z}$ of it. Since the $m - d$ remaining training examples are drawn IID from $\mathbf{P}_{\mathsf{Z}}$ we can apply the binomial tail bound, Theorem 7 in Appendix A, thus bounding the probability in (4) by $\exp(-(m-d)\varepsilon)$. The number of different index vectors $\mathbf{i} \in I_{d,m}$ is given by $m^d = |I_{d,m}|$ for the case that $\mathcal{R}$ is not permutation and repetition invariant and $\binom{m}{d}$ in the case that

$\mathcal{R}$ is permutation and repetition invariant. As a result, the probability in (3) is strictly less than $m^d \exp\left(-\left(m-d\right)\varepsilon\right)$ or $\binom{m}{d} \exp\left(-\left(m-d\right)\varepsilon\right)$, respectively.

We have with probability at least $1-\delta$ over the random draw of the training sample $\boldsymbol{z} \in \mathcal{Z}^m$ that the proposition $\Upsilon_d\left(\boldsymbol{z}, \delta\right)$ defined by

$$\hat{R}\left[\mathcal{A}\left(\boldsymbol{z}\right), \boldsymbol{z}\right] = 0 \ \wedge \ \left|\mathcal{C}\left(\boldsymbol{z}\right)\right| = d \ \Rightarrow \ R\left[\mathcal{A}\left(\boldsymbol{z}\right)\right] \leq \frac{\log\left(m^d\right) + \log\left(\frac{1}{\delta}\right)}{m-d}$$

holds true (with m^d replaced by $\binom{m}{d}$ for the permutation and repetition invariant case). Finally, we apply the stratification lemma, Lemma 1 in Appendix A, to the sequence of propositions Υ_d with $\mathbf{P}_{\mathsf{D}}\left(d\right) = \frac{1}{m}$ for all $d \in \{1, \ldots, m\}$. □

The bound (2) in Theorem 2 is easily interpreted if we consider the bound on the binomial coefficient, $\binom{m}{d} < \left(\frac{em}{d}\right)^d$, thus obtaining[2]

$$R\left[\mathcal{A}\left(\boldsymbol{z}\right)\right] \leq \frac{2}{m}\left(d \log\left(\frac{em}{d}\right) + \log\left(m\right) + \log\left(\frac{1}{\delta}\right)\right) . \tag{5}$$

This result should be compared to the simple VC bound (see, e.g., Cristianini and Shawe-Taylor (2000)),

$$\varepsilon\left(m, d_{\mathrm{VC}}, \delta\right) = \frac{2}{m}\left(d_{\mathrm{VC}} \log_2\left(\frac{2em}{d_{\mathrm{VC}}}\right) + \log_2\left(\frac{2}{\delta}\right)\right) . \tag{6}$$

Ignoring constants that are worse in the VC bound, these two bounds almost look alike. The (data-dependent) number d of examples needed by the compression scheme replaces the VC dimension $d_{\mathrm{VC}} := \mathrm{VCdim}\left(\mathcal{H}\right)$ of the underlying hypothesis space. Compression bounds can thus provide bounds on the prediction error of classifiers even if the classifier is chosen from an hypothesis space $\mathcal{H}$ of infinite VC dimension. The relation between VC bounds and compression schemes—motivated by equations such as (5) and (6)—is still not fully explored (see Floyd and Warmuth (1995) and recently Warmuth (2003)). We observe an interesting analogy between the ghost sample argument in VC theory (see Herbrich (2001) for an overview) and the use of the remaining $m-d$ examples from the sample. While the uniform convergence requirement in VC theory forces us to assume an extra ghost sample to be able to bound the true risk, the $m-d$ training examples serve the same purpose in the compression framework: To measure an empirical risk that serves to bound the true risk.

The second interesting observation about Theorem 2 is that the bound for a permutation and repetition invariant compression scheme

[2] Note that the bound is trivially true for $d > \frac{m}{2}$; otherwise $\frac{1}{m-d} \leq \frac{2}{m}$.

is slightly better than its counterpart without this invariance. This difference can be understood from a coding point of view: It requires more bits to encode a sequence of indices (where order and repetition matter) as compared to a set of indices (where order does *not* matter and there are no repetitions).

In the proof of the PAC compression bound, Theorem 2, the stratification over the number d of training examples used was carried out using a uniform (prior) measure $\mathbf{P}_{\mathsf{D}}(1) = \cdots = \mathbf{P}_{\mathsf{D}}(m) = \frac{1}{m}$ indicating complete ignorance about the sparseness to be expected. In a PAC-Bayesian spirit, however, we may choose a more "natural" prior that expresses our prior belief about the sparseness to be achieved. To this end we assume that given a training sample $\boldsymbol{z} \in \mathcal{Z}^m$ the probability p that any given example $z_i \in \boldsymbol{z}$ will be in the compression sample $\boldsymbol{z}_{\mathcal{C}(\boldsymbol{z})}$ is constant and independent of $\boldsymbol{z}$. This induces a distribution over $d = |\mathcal{C}(\boldsymbol{z})|$ given for all $d \in \{1, \ldots, m\}$ by

$$\mathbf{P}_{\mathsf{D}}(d) = \binom{m}{d} p^d (1-p)^{m-d} ,$$

for which we have $\sum_{i=1}^{m} \mathbf{P}_{\mathsf{D}}(i) \leq 1$ as required for the stratification lemma, Lemma 1 in Appendix A. The value p thus serves as an a-priori belief about the value of the observed compression coefficient $\hat{p} := \frac{d}{m}$. This alternative sequence leads to the following bound for permutation and repetition invariant compression schemes,

$$R[\mathcal{A}(\boldsymbol{z})] \leq 2 \cdot \left(\hat{p} \log\left(\frac{1}{p}\right) + (1-\hat{p}) \log\left(\frac{1}{1-p}\right) + \frac{1}{m} \log\left(\frac{1}{\delta}\right) \right) . \quad (7)$$

Note that the term $\hat{p} \log\left(\frac{1}{p}\right) + (1-\hat{p}) \log\left(\frac{1}{1-p}\right)$ can be interpreted as the cross entropy between two random variables that are Bernoulli-distributed with success probabilities p and $\hat{p}$, respectively. For an illustration of how a suitably chosen value p of the expected compression ratio can decrease the bound value for a given value $\hat{p}$ of the compression ratio consider Figure 3.2.

3.3. The Unrealizable Case

The previous compression bound indicates an interesting relation between PAC/VC theory and data compression. Of course, data compression schemes come in two flavors, lossy and non-lossy. Thus it comes as no surprise that we can derive bounds on the prediction error of compression schemes also for the unrealisable case with non-zero empirical risk (Graepel et al., 2000). Note that these results are implicitly contained in Floyd and Warmuth (1995) where the authors consider the

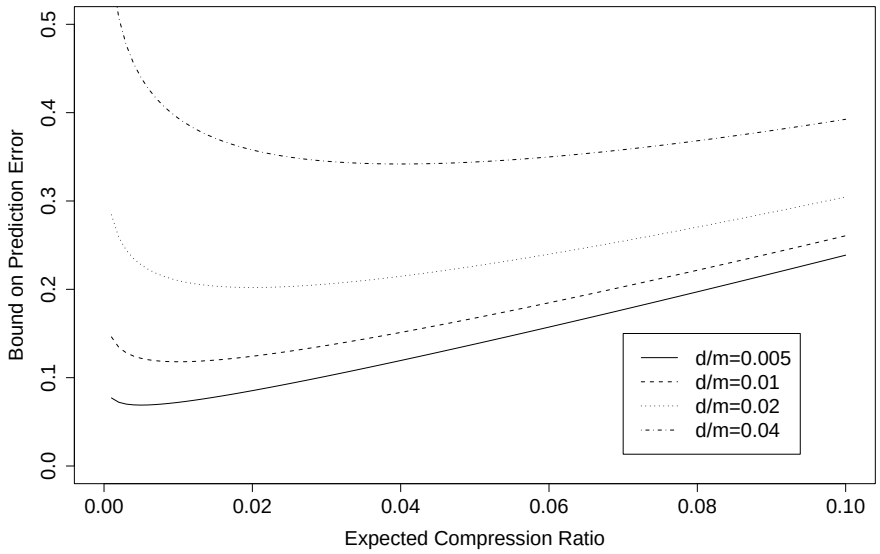

Figure 2. Dependency of the PAC-Bayesian compression bound (7) on the expected value p and the observed value $\hat{p}$ of the compression coefficient. For increasing values $\hat{p} := \frac{d}{m}$ the optimal choice of the expected compression ratio p increases as indicated by the shifted minima of the family of curves ($m = 1000$, $\delta = 0.05$).

more general scenario that the reconstruction function $\mathcal{R}$ also gets r bits of side-information.

Theorem 3 (Lossy compression bound). *Let $\mathcal{A} : \mathcal{Z}^{(m\infty)} \to \mathcal{H}$ be a compression scheme. For any probability measure $\mathbf{P}_{\mathsf{Z}}$, any $m \in \mathbb{N}$, and any $\delta \in (0, 1]$, with probability at least $1 - \delta$ over the random draw of the training sample $\boldsymbol{z} \in \mathcal{Z}^m$, if $d = |\mathcal{C}(\boldsymbol{z})|$ the prediction error of $\mathcal{A}(\boldsymbol{z})$ is bounded from above by*

$$R[\mathcal{A}(\boldsymbol{z})] \leq \frac{m}{m-d} \hat{R}[\mathcal{A}(\boldsymbol{z}), \boldsymbol{z}] + \sqrt{\frac{\log(m^d) + 2\log(m) + \log\left(\frac{1}{\delta}\right)}{2(m-d)}},$$

and, if $\mathcal{A}$ is a permutation and repetition invariant compression scheme, then by

$$R[\mathcal{A}(\boldsymbol{z})] \leq \frac{m}{m-d} \hat{R}[\mathcal{A}(\boldsymbol{z}), \boldsymbol{z}] + \sqrt{\frac{\log\binom{m}{d} + 2\log(m) + \log\left(\frac{1}{\delta}\right)}{2(m-d)}}.$$

Proof. Fixing the number of training errors $q \in \{1, \ldots, m\}$ and $|\mathcal{C}(\boldsymbol{z})|$ we bound—in analogy to the proof of Theorem 2—the probability

$$\begin{aligned}\mathbf{P}_{\mathbf{Z}^m}\left(\hat{R}\left[\mathcal{A}\left(\mathbf{Z}\right),\mathbf{Z}\right] \leq \frac{q}{m} \ \wedge \ R\left[\mathcal{A}\left(\mathbf{Z}\right)\right] > \varepsilon \ \wedge \ \left|\mathcal{C}\left(\mathbf{Z}\right)\right| = d\right) \\ \leq \sum_{\mathbf{i}\in I_{d,m}} \mathbf{P}_{\mathbf{Z}^m}\left(\hat{R}\left[\mathcal{R}\left(\mathbf{Z}_{\mathbf{i}}\right),\mathbf{Z}\right] \leq \frac{q}{m} \ \wedge \ R\left[\mathcal{R}\left(\mathbf{Z}_{\mathbf{i}}\right)\right] > \varepsilon\right). \quad (8)\end{aligned}$$

We have that $m \cdot \hat{R}\left[\mathcal{A}\left(\boldsymbol{z}\right),\boldsymbol{z}\right] \leq q$ implies $(m-d) \cdot \hat{R}\left[\mathcal{A}\left(\boldsymbol{z}\right),\boldsymbol{z}_{\bar{\mathbf{i}}}\right] \leq q$ for all $\mathbf{i} \in I_{m,d}$ and $\bar{\mathbf{i}} := \{1, \ldots, m\} \setminus \mathbf{i}$ leading to an upper bound,

$$\mathbf{E}_{\mathbf{Z}^d}\left[\mathbf{P}_{\mathbf{Z}^{m-d}|\mathbf{Z}^d=\boldsymbol{z}_{\mathbf{i}}}\left(\hat{R}\left[\mathcal{R}\left(\boldsymbol{z}_{\mathbf{i}}\right),\mathbf{Z}\right] \leq \frac{q}{m-d} \ \wedge \ R\left[\mathcal{R}\left(\boldsymbol{z}_{\mathbf{i}}\right)\right] > \varepsilon\right)\right], \quad (9)$$

on the probability in (8). From Hoeffding's inequality, Theorem 8 in Appendix A, we know for a given sample $\boldsymbol{z}_{\mathbf{i}}$ that the probability in (9) is bounded by

$$\exp\left(-2\left(m-d\right)\left(\varepsilon - \frac{q}{m-d}\right)^2\right).$$

The number of different index vectors $\mathbf{i} \in I_{d,m}$ is again given by m^d for the case that $\mathcal{R}$ is not permutation and repetition invariant and $\binom{m}{d}$ in the case that $\mathcal{R}$ is permutation and repetition invariant.

Thus we have with probability at least $1-\delta$ over the random draw of the training sample $\boldsymbol{z} \in \mathcal{Z}^m$ for all compression schemes $\mathcal{A}$ and maximal number of training errors q that the proposition $\Upsilon_{d,q}\left(\boldsymbol{z},\delta\right)$ given by

$$\begin{gathered}\hat{R}\left[\mathcal{A}\left(\boldsymbol{z}\right),\boldsymbol{z}\right] \leq \tfrac{q}{m} \ \wedge \ \left|\mathcal{C}\left(\boldsymbol{z}\right)\right| = d \\ \Rightarrow \\ R\left[\mathcal{A}\left(\boldsymbol{z}\right)\right] \leq \tfrac{q}{m-d} + \sqrt{\tfrac{\log\left(m^d\right)+\log\left(\frac{1}{\delta}\right)}{2(m-d)}}\end{gathered}$$

holds true (with m^d replaced by $\binom{m}{d}$ for the permutation invariant case). Finally, we apply the stratification lemma, Lemma 1 in Appendix A, to the sequence of propositions $\Upsilon_{d,q}$ with $\mathbf{P}_{\mathsf{DQ}}\left((d,q)\right) = m^{-2}$ for all $(d,q) \in \{1, \ldots, m\}^2$. □

The above theorem is proved using a simple combination of Hoeffding's inequality and a double stratification about the number d of non-zero coefficients and the number of empirical errors, q. From an information theoretic point of view the first term of the right hand side of the inequalities represents the number of bits required to explicitly

transfer the labels of the misclassified examples—this establishes the link to the more general results of Floyd and Warmuth (1995). Note also that Marchand and Shawe-Taylor (2001) prove a similar result to Theorem 3, avoiding the square root in the bound at the cost of a less straight-forward argument and worse constants.

4. PAC Bounds for Online Learning

In this section we will review the relation between PAC bounds and mistake bounds for online learning algorithms. This relation has been studied before and Theorem 4 is a direct consequence of Theorem 3 in Floyd and Warmuth (1995).

In light of the relationship, we will reconsider the perceptron algorithm and derive a PAC bound for the resulting classifiers from a mistake bound involving the margin a support vector machine would achieve on the same training data. We will argue that a large potential margin is sufficient to obtain good bounds on the prediction error of all the classifiers found by the perceptron on permuted training sequences $\boldsymbol{z}_{\mathbf{j}}$. Although this result is a straightforward application of Theorem 4 it went unnoticed and is, so far, missing in any comparative study of margin bounds—which form the theoretical basis of all margin based algorithms including the support vector machine algorithm.

4.1. Online-Learning and Mistake Bounds

In order to be able to discuss the perceptron convergence theorem and the relation between mistake bounds and PAC bounds in more depth let us introduce formally the notion of an online algorithm (Littlestone, 1988).

Definition 9 (Online learning algorithm). Consider an update function $\mathcal{U} : \mathcal{Z} \times \mathcal{H} \to \mathcal{H}$ and an initial hypothesis $h_0 \in \mathcal{H}$. An *online learning algorithm* is a function $\mathcal{A} : \mathcal{Z}^{(\infty)} \times \bigcup_{m=1}^{\infty} \{1, \ldots, m\}^{(\infty)} \times \mathcal{H} \to \mathcal{H}$ that takes a training sample $\boldsymbol{z} \in \mathcal{Z}^m$, a *training sequence* $\mathbf{j} \in \bigcup_{m=1}^{\infty} \{1, \ldots, m\}^{(\infty)}$, and an *initial hypothesis* $h_0 \in \mathcal{H}$, and produces the final hypothesis $\mathcal{A}_{\mathcal{U}}(\boldsymbol{z}) := h_{|\mathbf{j}|}$ of the $|\mathbf{j}|$-fold recursion of the update function $\mathcal{U}$,

$$h_i := \mathcal{U}(z_{j_i}, h_{i-1}) \ .$$

Mistake-driven learning algorithms are a particular class of online algorithms that only change their current hypothesis if it causes an error on the current training example.

Definition 10 (Mistake-driven learning algorithm). An online algorithm $\mathcal{A}_{\mathcal{U}}$ is called *mistake-driven* if the update function satisfies for all $x \in \mathcal{X}$, for all $y \in \mathcal{Y}$, and for all $h \in \mathcal{H}$ that

$$y = h(x) \quad \Rightarrow \quad \mathcal{U}((x,y),h) = h\,.$$

In the PAC framework we focus on the error of the final hypothesis $\mathcal{A}(\boldsymbol{z})$ an algorithm produces after considering the whole training sample $\boldsymbol{z}$. In the analysis of online-algorithms one takes a slightly different view: The number of updates until convergence is considered the quantity of interest.

Definition 11 (Mistake bound). Consider an hypothesis space $\mathcal{H}$, a training sample $\boldsymbol{z} \in \mathcal{Z}^m$ labeled by a hypothesis $h \in \mathcal{H}$ and a sequence $\mathbf{j} \in \{1,\ldots,m\}^{(\infty)}$. Denote by $\tilde{\mathbf{j}} \subseteq \mathbf{j}$ the sequence of mistakes, i.e., the subsequence of $\mathbf{j}$ containing the indices $j_i \in \{1,\ldots,m\}$ for which $h_{i-1} \neq h_i$. We call a function $M_{\mathcal{U}} : \mathcal{Z}^{(\infty)} \to \mathbb{N}$ a *mistake bound* for the online algorithm $\mathcal{A}_{\mathcal{U}}$ if it bounds the number $|\tilde{\mathbf{j}}|$ of mistakes $\mathcal{A}_{\mathcal{U}}$ makes on $\boldsymbol{z} \in \mathcal{Z}^m$,

$$|\tilde{\mathbf{j}}| \leq M_{\mathcal{U}}\,,$$

for any ordering $\mathbf{j} \in \{1,\ldots,m\}^{(\infty)}$.

In a sense, this is a very practical measure of error assuming that a learning machine is learning "on the job".

4.2. From Online to Batch Learning

Interestingly, we can relate any mistake bound for a mistake-driven algorithm to a PAC style bound on the prediction error:

Theorem 4 (Mistake bound to PAC bound). *Consider a mistake-driven online learning algorithm $\mathcal{A}_{\mathcal{U}}$ for $\mathcal{H}$ with a mistake bound $M_{\mathcal{U}} : \mathcal{Z}^{(\infty)} \to \mathbb{N}$. For any probability measure P_{Z}, any $m \in \mathbb{N}$, and any $\delta \in (0,1]$, with probability at least $1-\delta$ over the random draw of the training sample $\boldsymbol{z} \in \mathcal{Z}^m$ we have that the true risk $R[\mathcal{A}_{\mathcal{U}}(\boldsymbol{z})]$ of the hypothesis $\mathcal{A}_{\mathcal{U}}(\boldsymbol{z})$ is bounded from above by*

$$R[\mathcal{A}_{\mathcal{U}}(\boldsymbol{z})] \leq \frac{2}{m}\left(\left(M_{\mathcal{U}}(\boldsymbol{z})+1\right)\log(m) + \log\left(\frac{1}{\delta}\right)\right)\,. \tag{10}$$

Proof. The proof is based on the fact that a mistake-driven algorithm constitutes a (non permutation and repetition invariant) compression scheme. Assume we run $\mathcal{A}_{\mathcal{U}}$ twice on the same training sample $\boldsymbol{z}$ and training sequence $\mathbf{j}$. From the first run we obtain the sequence of mistakes $\tilde{\mathbf{j}}$. Thus we have for the compression function $\mathcal{C}$,

$$\mathcal{C}(\boldsymbol{z}_{\mathbf{j}}) := \tilde{\mathbf{j}}\,.$$

Running $\mathcal{A}_\mathcal{U}$ only on $\boldsymbol{z}_{\tilde{\mathbf{j}}}$ then leads to the same hypothesis as before,

$$\mathcal{A}_\mathcal{U}(\boldsymbol{z}, \mathbf{j}) = \mathcal{A}_\mathcal{U}\left(\boldsymbol{z}, \tilde{\mathbf{j}}\right)$$

showing that the reconstruction function $\mathcal{R}$ is given by the algorithm $\mathcal{A}_\mathcal{U}$ itself. The compression scheme is in general not permutation and repetition invariant because $\mathcal{A}_\mathcal{U}$ and hence $\mathcal{R}$ is not. We can thus apply Theorem 2, where we bound d from above by $M_\mathcal{U}$ and use $\frac{1}{m-d} \leq \frac{2}{m}$ for all $d \leq \frac{m}{2}$. □

Let us consider two examples for the application of this theorem. The first example illustrates the relation between PAC/VC theory and the mistake bound framework:

Example 1 (Halving algorithm). For finite hypothesis spaces $\mathcal{H}$, $|\mathcal{H}| < \infty$, the so-called halving algorithm $\mathcal{A}_{1/2}$ (Littlestone, 1988) achieves a minimal mistake bound of

$$M_{1/2}(\boldsymbol{z}) = \lceil \log_2(|\mathcal{H}|) \rceil \ .$$

The algorithm proceeds as follows:

1. Initialize the set $V_0 := \mathcal{H}$ and $t = 0$.
2. For a given input $x_i \in \mathcal{X}$ predict the class $\hat{y}_i \in \mathcal{Y}$ that receives the majority of votes from classifiers $h \in V_t$,
$$\hat{y}_i = \operatorname*{argmax}_{y \in \mathcal{Y}} |\{h \in V_t : h(x_i) = y\}| \ .$$
3. If a mistake occurs, that is $y_i \neq \hat{y}_i$, all classifiers $h \in V_t$ that are inconsistent with x_i are removed,
$$V_{t+1} := V_t \setminus \{h \in V_t : h(x_i) \neq y_i\} \ .$$
4. If no more mistakes occur, return any classifier $h \in V_t$; otherwise goto 2.

Plugging the value $M_{1/2}(\boldsymbol{z})$ into the bound (10) gives

$$R\left[\mathcal{A}_{1/2}(\boldsymbol{z})\right] \leq \frac{2}{m}\left((\lceil \log_2(|\mathcal{H}|) \rceil + 1)\log(m) + \log\left(\frac{1}{\delta}\right)\right) \ ,$$

which holds uniformly over version space V_M and up to a factor of $2\log(m)$ recovers what is known as the cardinality bound in PAC/VC theory.

The second example provides a surprising way of proving bounds for linear classifiers based on the well-known margin γ by a combination of mistake bounds and compression bounds:

Example 2 (Perceptron algorithm). The perceptron algorithm $\mathcal{A}_{\text{perc}}$ is possibly the best-known mistake-driven online algorithm (Rosenblatt, 1962). The perceptron convergence theorem provides a mistake bound for the perceptron algorithm given by

$$M_{\text{perc}}\left(\boldsymbol{z}\right) = \left(\frac{\varsigma\left(\boldsymbol{x}\right)}{\gamma^{*}\left(\boldsymbol{z}\right)}\right)^{2},$$

with $\varsigma^{2}\left(\boldsymbol{x}\right) := \max_{x_i \in \boldsymbol{x}} \left\|x_i\right\|$ being the data radius[3] and

$$\gamma^{*}\left(\boldsymbol{z}\right) := \max_{w} \min_{(x_i, y_i) \in \boldsymbol{z}} y_i \left\langle x_i, w\right\rangle / \left\|w\right\| ,$$

being the maximum margin that can be achieved on $\boldsymbol{z}$. Plugging the value $M_{\text{perc}}\left(\boldsymbol{z}\right)$ into the bound (10) gives

$$R\left[\mathcal{A}_{\text{perc}}\left(\boldsymbol{z}\right)\right] \leq \frac{2}{m}\left(\left(\left(\frac{\varsigma\left(\boldsymbol{x}\right)}{\gamma^{*}\left(\boldsymbol{z}\right)}\right)^{2} + 1\right)\log\left(m\right) + \log\left(\frac{1}{\delta}\right)\right) .$$

This result bounds the prediction error of any solution found by the perceptron algorithm in terms of the quantity $\varsigma\left(\boldsymbol{x}\right)/\gamma^{*}\left(\boldsymbol{z}\right)$, that is, in terms of the margin $\gamma^{*}\left(\boldsymbol{z}\right)$ a support vector machine (SVM) would achieve on the same data sample $\boldsymbol{z}$. Remarkably, the above bound gives lower values than typical margin bounds (Vapnik, 1998; Bartlett and Shawe-Taylor, 1998; Shawe-Taylor et al., 1998) for classifiers w in terms of their individual margins $\gamma\left(w, \boldsymbol{z}\right)$ that have been put forward as justifications of large margin algorithms. As a consequence, whenever the SVM appears to be theoretically justified by a large observed margin $\gamma^{*}\left(\boldsymbol{z}\right)$, every solution found by the perceptron algorithm has a small guaranteed prediction error—mostly bounded more tightly than current bounds on the prediction error of SVMs.

5. PAC-Bayesian Compression Bounds

In the proofs of the compression results, Theorem 2 and Theorem 3, we made use of the fact that $m - d$ of the m training examples had not been used for constructing the classifier and could thus be used to bound the true risk with high probability. In this section, we will make

[3] Note that in this example we assume $x_i \in \mathbb{R}^N$ and $w \in \mathbb{R}^N$.

use of similar arguments in order to deal with data-dependent hypothesis spaces such as those parameterized by $\boldsymbol{\alpha}$ in kernel classifiers. This function class constitutes the basis of support vector machines, Bayes point machines, and other kernel classifiers (see Herbrich (2001) for an overview). Note that our results neither rely on the kernel function k to be positive definite or even symmetric nor is is relevant which algorithm is used to construct the final kernel classifiers. For example, these bounds also apply to kernel classifiers learned with the relevance vector machine. Obviously, typical VC results cannot be applied to this type of data-dependent hypothesis class, because the hypothesis class is not fixed in advance. Hence, its complexity cannot be determined before learning[4]. In this section we will proceed similarly to McAllester (1998): First we prove a PAC-Bayesian "folk" theorem, then we proceed with a PAC-Bayesian subset bound.

5.1. The PAC-Bayesian Folk Theorem for Data-Dependent Hypotheses

Suppose instead of a PAC-Bayesian prior P_{H} over a fixed hypothesis space we define a prior P_{A} over the sequence $\boldsymbol{\alpha}$ of expansion coefficients α_i in (1). Relying on a sparse representation with $\|\boldsymbol{\alpha}\|_0 < m$ we can then prove the following theorem:

Theorem 5 (PAC-Bayesian bound for single data-dependent classifiers). *For any prior probability distribution P_{A} on a countable subset $A \subset \mathbb{R}^m$ satisfying $\mathsf{P}_{\mathsf{A}}(\boldsymbol{\alpha}) > 0$ for all $\boldsymbol{\alpha} \in A$, for any probability measure P_{Z}, any $m \in \mathbb{N}$, and for all $\delta \in (0,1]$ we have with probability at least $1-\delta$ over the random draw of the training sample $\boldsymbol{z} \in \mathcal{Z}^m$ that for any hypothesis $h_{(\ ,\boldsymbol{x})} \in \mathcal{H}_k(\boldsymbol{x})$ the prediction error $R\left[h_{(\ ,\boldsymbol{x})}\right]$ is bounded by*

$$R\left[h_{(\ ,\boldsymbol{x})}\right] \leq \frac{1}{m - \|\boldsymbol{\alpha}\|_0} \left(\log\left(\frac{1}{\mathsf{P}_{\mathsf{A}}(\boldsymbol{\alpha})} \right) + \log\left(\frac{1}{\delta} \right) \right) .$$

Proof. First we show that the proposition $\Upsilon\ \ (\boldsymbol{z}, \|\boldsymbol{\alpha}\|_0, \delta)$,

$$\Upsilon\ \ (\boldsymbol{z}, \|\boldsymbol{\alpha}\|_0, \delta) := \left(\hat{R}\left[h_{(\ ,\boldsymbol{x})}, \boldsymbol{z}\right] = 0 \ \Rightarrow\ R\left[h_{(\ ,\boldsymbol{x})}\right] \leq \frac{\log\left(\frac{1}{\delta}\right)}{m - \|\boldsymbol{\alpha}\|_0} \right) , \tag{11}$$

[4] A fixed hypothesis space is a pre-requisite in the VC analysis because it appeals to the union bound over all hypotheses which are distinguishable by their predictions on a double sample (see Herbrich (2001) for more details).

holds for all $\boldsymbol{\alpha} \in A$ with probability at least $1-\delta$ over the random draw of $\boldsymbol{z} \in \mathcal{Z}^m$. Let $\mathbf{i} \in I_{d,m}$, $d := \|\boldsymbol{\alpha}\|_0$, be the index vector with entries at which $\alpha_i \neq 0$. Then we have for all $\boldsymbol{\alpha} \in A$ that

$$\begin{aligned}
&\mathsf{P}_{\mathsf{Z}^m}\left(\hat{R}\left[h_{(\ ,\mathbf{X})}, \mathbf{Z}\right] = 0 \ \wedge \ R\left[h_{(\ ,\mathbf{X})}\right] > \varepsilon\right) \\
&\quad \leq \mathsf{P}_{\mathsf{Z}^m}\left(\hat{R}\left[h_{(\ ,\mathbf{X}_\mathbf{i})}, \mathbf{Z}\right] = 0 \ \wedge \ R\left[h_{(\ ,\mathbf{X}_\mathbf{i})}\right] > \varepsilon\right) \\
&\quad \leq \mathsf{E}_{\mathsf{Z}^d}\left[\mathsf{P}_{\mathsf{Z}^{m-d}|\mathsf{Z}^d=\boldsymbol{z}_\mathbf{i}}\left(\hat{R}\left[h_{(\ ,\boldsymbol{x}_\mathbf{i})}, \mathbf{Z}\right] = 0 \ \wedge \ R\left[h_{(\ ,\boldsymbol{x}_\mathbf{i})}\right] > \varepsilon\right)\right] \\
&\quad < (1-\varepsilon)^{m-d} \leq \exp\left(-\varepsilon\left(m-d\right)\right) .
\end{aligned}$$

The key is that the classifier $h_{(\ ,\boldsymbol{x})}$ does not change over the random draw of the $m-d$ examples not used in its expansion. Finally, apply the stratification lemma, Lemma 1 in Appendix A, to the proposition $\Upsilon \ \ (\boldsymbol{z}, \|\boldsymbol{\alpha}\|_0, \delta)$ with $\mathsf{P}_\mathsf{A}(\boldsymbol{\alpha})$. □

Obviously, replacing the binomial tail bound with Hoeffding's inequality, Theorem 8, allows us to derive a result for the unrealisable case with non-zero empirical risk. This bound then reads

$$R\left[h_{(\ ,\boldsymbol{x})}\right] \leq \frac{m}{m-\|\boldsymbol{\alpha}\|_0}\hat{R}\left[h_{(\ ,\boldsymbol{x})}\right] + \sqrt{\frac{\log\left(\frac{1}{\mathsf{P}_\mathsf{A}(\ )}\right) + \log\left(\frac{m}{\delta}\right)}{2\left(m-\|\boldsymbol{\alpha}\|_0\right)}} .$$

Remark 1. Note that both these results are not direct consequences of Theorem 2 and 3 since in these new results the bound depends on *both* the sparsity $\|\boldsymbol{\alpha}\|_0$ and the prior $\mathsf{P}_\mathsf{A}(\boldsymbol{\alpha})$ of the particular hypothesis $h_{(\ ,\boldsymbol{x})}$ as opposed to only the sparsity d of the compression scheme that produced h in Theorem 2 and 3. Note that any prior P_A over a finite subset of $\boldsymbol{\alpha}$'s is effectively encoding a prior over infinitely many hypotheses $\left\{h_{(\ ,\boldsymbol{x})} \,|\boldsymbol{x} \in \mathcal{X}^m, \mathsf{P}_\mathsf{A}(\boldsymbol{\alpha}) > 0\right\}$. It is not possible to incorporate such a prior into both Theorem 2 or 3 using the union bound.

Example 3 (1-norm soft margin perceptron). Suppose we run the (kernel) perceptron algorithm with box-constraints $0 \leq \alpha_i \leq C$ (see, e.g., Herbrich (2001)) and obtain a classifier $h_{(\ ,\boldsymbol{x})}$ with d non-zero coefficients α_i. For a prior

$$\mathsf{P}_\mathsf{A}(\boldsymbol{\alpha}) := \frac{1}{m\binom{m}{\|\ \|_0}(2C+1)^{\|\ \|_0}} \tag{12}$$

over the set $\{\boldsymbol{\alpha} \in \mathbb{R}^m \,|\, \boldsymbol{\alpha} \in \{-C, \ldots, 0, \ldots, C\}^m\}$ we get the bound

$$R\left[h_{(\ ,\boldsymbol{x})}\right] \leq \frac{m}{m-d}\hat{R}\left[h_{(\ ,\boldsymbol{x})}\right] + \sqrt{\frac{\log\binom{m}{d} + d\log\left(2C+1\right) + \log\left(\frac{m^2}{\delta}\right)}{2\left(m-d\right)}} ,$$

which yields lower values than the compression bound, Theorem 3, for non-permutation and repetition invariant compression schemes if $(2C+1) < d$. This can be seen by bounding $\log(d!)$ by $d\log(d)$ in Theorem 3 using Stirling's formula, Theorem 10 in Appendix A.

5.2. The PAC-Bayesian Subset Bound for Data-Dependent Hypotheses

Let us now consider a PAC-Bayesian subset bound for the data-dependent hypothesis space of kernel classifiers (1). In order to make the result more digestible we consider it for a fixed number d of non-zero coefficients.

Theorem 6 (PAC-Bayesian bound for subsets of data-dependent classifiers). *For any prior probability distribution $\mathbf{P}_{\mathbf{A}}$, for any probability measure $\mathbf{P}_{\mathbf{Z}}$, for any $m \in \mathbb{N}$, for any $d \in \{1,\ldots,m\}$, and for all $\delta \in (0,1]$ we have with probability at least $1-\delta$ over the random draw of the training sample $\boldsymbol{z} \in \mathcal{Z}^m$ that for any subset A, $\mathbf{P}_{\mathbf{A}}(A) > 0$, with constant sparsity d and zero empirical risk, $\forall \boldsymbol{\alpha} \in A : \|\boldsymbol{\alpha}\|_0 = d \wedge \hat{R}\left[h_{(\ ,\boldsymbol{x})}(\boldsymbol{z})\right] = 0$, the average prediction error $\mathbf{E}_{\mathbf{A}|\mathbf{A}\in A}\left[R\left[h_{(\mathbf{A},\boldsymbol{x})}\right]\right]$ is bounded by*

$$\mathbf{E}_{\mathbf{A}|\mathbf{A}\in A}\left[R\left[h_{(\mathbf{A},\boldsymbol{x})}\right]\right] \leq \frac{\log\left(\frac{1}{\mathbf{P}_{\mathbf{A}}(A)}\right) + 2\log(m) + \log\left(\frac{1}{\delta}\right) + 1}{m-d}\,.$$

Proof. Using the fact that the loss function l_{0-1} is bounded from above by 1, we decompose the expectation at some point $\varepsilon \in \mathbb{R}$ by

$$\begin{aligned}&\mathbf{E}_{\mathbf{A}|\mathbf{A}\in A}\left[R\left[h_{(\mathbf{A},\boldsymbol{x})}\right]\right] \leq \\ &\quad \varepsilon \cdot \mathbf{P}_{\mathbf{A}|\mathbf{A}\in A}\left(R\left[h_{(\mathbf{A},\boldsymbol{x})}\right] \leq \varepsilon\right) + 1 \cdot \mathbf{P}_{\mathbf{A}|\mathbf{A}\in A}\left(R\left[h_{(\mathbf{A},\boldsymbol{x})}\right] > \varepsilon\right). \quad (13)\end{aligned}$$

As in the proof of Theorem 5 we have that for all $\boldsymbol{\alpha} \in A$ and for all $\delta \in (0,1]$,

$$\mathbf{P}_{\mathbf{Z}^m|\mathbf{A}=}\left(\Upsilon\ \ (\mathbf{Z},d,\delta)\right) \geq 1-\delta\,,$$

where the proposition $\Upsilon\ \ (\boldsymbol{z},d,\delta)$ is given by (11). By the quantifier reversal lemma, Lemma 2 in Appendix A, this implies that for all $\beta \in (0,1)$ with probability at least $1-\delta$ over the random draw of the training sample $\boldsymbol{z} \in \mathcal{Z}^m$ for all $\gamma \in (0,1]$,

$$\begin{aligned}\mathbf{P}_{\mathbf{A}|\mathbf{Z}^m=\boldsymbol{z}}\left(\neg\Upsilon_{\mathbf{A}}\left(\boldsymbol{z},d,(\gamma\beta\delta)^{\frac{1}{1-\beta}}\right)\right) &< \gamma \\ \mathbf{P}_{\mathbf{A}|\mathbf{Z}^m=\boldsymbol{z}}\left(\hat{R}\left[h_{(\mathbf{A},\boldsymbol{x})},\boldsymbol{z}\right] = 0 \ \wedge\ R\left[h_{(\mathbf{A},\boldsymbol{x})}\right] > \varepsilon(\gamma,\beta)\right) &< \gamma\end{aligned}$$

with

$$\varepsilon(\gamma,\beta) := \frac{\log\left(\frac{1}{\delta\gamma\beta}\right)}{(1-\beta)(m-d)}.$$

Since the distribution over $\boldsymbol{\alpha}$ is by assumption a prior and thus independent of the data, we have $\mathbf{P}_{\mathbf{A}|\mathbf{Z}^m=\boldsymbol{z}} = \mathbf{P}_{\mathbf{A}}$ and hence

$$\begin{aligned}\mathbf{P}_{\mathbf{A}|\mathbf{A}\in A}\left[R\left[h_{(\mathbf{A},\boldsymbol{x})}\right] > \varepsilon(\gamma,\beta)\right] &= \frac{\mathbf{P}_{\mathbf{A}}\left(\mathbf{A}\in A \wedge R\left[h_{(\mathbf{A},\boldsymbol{x})}\right] > \varepsilon(\gamma,\beta)\right)}{\mathbf{P}_{\mathbf{A}}(A)} \\ &\leq \frac{\gamma}{\mathbf{P}_{\mathbf{A}}(A)},\end{aligned}$$

because by assumption $\boldsymbol{\alpha}\in A$ implies $\hat{R}\left[h_{(\ ,\boldsymbol{x})},\boldsymbol{z}\right]=0$. Now choosing $\gamma = \frac{\mathbf{P}_{\mathbf{A}}(A)}{m}$ and $\beta=\frac{1}{m}$ we obtain from (13)

$$\begin{aligned}\mathbf{E}_{\mathbf{A}|\mathbf{A}\in A}\left[R\left[h_{(\mathbf{A},\boldsymbol{x})}\right]\right] &\leq \varepsilon(\gamma,\beta)\cdot\left(1-\frac{\gamma}{\mathbf{P}_{\mathbf{A}}(A)}\right)+\frac{\gamma}{\mathbf{P}_{\mathbf{A}}(A)} \\ &= \frac{\log\left(\frac{1}{\mathbf{P}_{\mathbf{A}}(A)}\right)+2\log(m)+\log\left(\frac{1}{\delta}\right)}{m-d}+\frac{1}{m}.\end{aligned}$$

Exploiting that $\frac{1}{m}\leq\frac{1}{m-d}$ completes the proof. □

Again, replacing the binomial tail bound with Hoeffding's inequality, Theorem 8, allows us to derive a result for the unrealisable case with non-zero empirical risk.

Example 4 (1-norm soft margin permutational perceptron sampling). Continuing the discussion of Example 3 with the same prior distribution (12) consider the following procedure: Learn a 1-norm soft margin perceptron with box constraints $0\leq\alpha_i\leq C$ for all $i\in\{1,\ldots,m\}$ and assume linear separability. Permute the compression sample $\boldsymbol{z}_{\mathbf{i}_{\mathrm{sv}}}$ and retrain to obtain an ensemble $A:=\{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_N\}$ of N different coefficient vectors $\boldsymbol{\alpha}_j$. Then the PAC-Bayesian subset bound for data dependent hypotheses, Theorem 6, bounds the average prediction error of the ensemble of classifiers $\left\{h_{(\ ,\boldsymbol{x})}\,|\,\boldsymbol{\alpha}\in A\right\}$ corresponding to the ensemble A of coefficient vectors.

6. Conclusions

We derived various bounds on the prediction error of sparse classifiers based on the idea of sample compression. Essentially, the results rely on

the fact that a classifier $h_{(\ ,\boldsymbol{x})}$ resulting from a compression scheme (of size d) is independent of the random draw of $m-d$ training examples, which—if classified with low or zero empirical risk by $h_{(\ ,\boldsymbol{x})}$—serve to ensure a low prediction error with high probability.

Our results in Section 4 relied on an interpretation of mistake-driven online learning algorithms as compression schemes. The mistake bound was then used as an upper bound on the size of the compression sample and thus lead to bounds on the prediction error of the final hypothesis returned by the algorithm. This procedure emphasizes the conceptual difference between our results and typical PAC/VC results: PAC/VC theory makes statements about uniform convergence within particular hypothesis classes $\mathcal{H}$. In contrast, compression results rely on assumptions about particular learning algorithms $\mathcal{A}$. This idea (which is carried further in Herbrich and Williamson (2002)) is promising in that it leads to bounds on the prediction error that are closer to the observed values and that take into account the actual learning algorithm used.

We extended the PAC-Bayesian results of McAllester (1998) to data-dependent hypotheses that are represented as linear expansions in terms of training inputs. The theorems are thus applicable to the class of kernel classifiers as defined in Definition 8, ranging from support vector to K-nearest-neighbors classifiers. Empirically, the bounds given yield rather low bound values and have low constants in comparison to VC bounds or bounds based on the observed margin. In summary, they are widely applicable and rather tight. The formulation of a prior over expansion coefficients $\boldsymbol{\alpha}$ that parameterize data-dependent hypotheses appears rather unusual. No contradiction, however, arises because the prior cannot be used to "cheat" by adjusting it in such a way as to manipulate the bound values. The reason is that the expansion (1) does not contain the labels y_i. Instead the prior serves to incorporate a-priori knowledge about the representation of classifiers in terms of training inputs. Of course, there exist many non-sparse classifiers with a low prediction error as well. It remains a challenging open question how we can formulate and prove PAC-Bayesian bounds for data-dependent hypotheses that are dense, i.e., that have few or no non-zero coefficients. Note that the PAC-Bayesian results in Langford and Shawe-Taylor (2003) only apply to a fixed hypothesis space by the assumption of a positive definite and symmetric kernel ensuring a fixed feature space.

Appendix

A. Basic Results

As a service to the reader we provide some basic results in the appendix for reference. Proofs using a rigorous and unified notation consistent with this paper can be found in Herbrich (2001).

A.1. Tail Bounds

At several points we require bounds on the probability mass in the tails of distributions. Assuming the zero-one loss, the simplest such bound is the binomial tail bound.

Theorem 7 (Binomial tail bound). *Let* $\mathsf{X}_1, \ldots, \mathsf{X}_n$ *be independent random variables distributed* Bernoulli (μ). *Then we have that*

$$\mathbf{P}_{\mathsf{X}^n}\left(\sum_{i=1}^{n} \mathsf{X}_i = 0\right) = (1-\mu)^n \leq \exp\left(-n\mu\right) .$$

For the case of non-zero empirical risk, we use Hoeffding's inequality (Hoeffding, 1963) that bounds the deviation between mean and expectation for bounded IID random variables.

Theorem 8 (Hoeffding's inequality). *Given* n *independent bounded random variables* $\mathsf{X}_1, \ldots, \mathsf{X}_n$ *such that for all* i $\mathbf{P}_{\mathsf{X}_i}\left(\mathsf{X}_i \in [a,b]\right) = 1$, *then we have for all* $\varepsilon > 0$

$$\mathbf{P}_{\mathsf{X}^n}\left(\frac{1}{n}\sum_{i=1}^{n} \mathsf{X}_i - \mathbf{E}_{\mathsf{X}}\left[\mathsf{X}\right] > \varepsilon\right) < \exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right) .$$

A.2. Binomial Coefficient and Factorial

For bounding combinatorial quantities the following two results are useful.

Theorem 9 (Bound on binomial coefficient). *For all* $m, d \in \mathbb{N}$ *with* $m \geq d$ *we have*

$$\log\binom{m}{d} \leq d \log\left(\frac{em}{d}\right)$$

Theorem 10 (Simple Stirling's approximation). *For all* $n \in \mathbb{N}$ *we have*

$$n\left(\log\left(n\right) - 1\right) < \log\left(n!\right) < n\log\left(n\right)$$

A.3. Stratification

In order to be able to make probabilistic statements uniformly over a given set we use a generalization of the so-called union bound, which we refer to as the stratification or multiple testing lemma.

Lemma 1 (Stratification). *Suppose we are given a set* $\{\Upsilon_1, \ldots, \Upsilon_s\}$ *of* s *measurable logic formulae* $\Upsilon : \mathcal{Z}^{(m)} \times (0,1] \to \{\text{true}, \text{false}\}$ *and a discrete probability measure* $\mathbf{P}_{\mathsf{I}}$ *over the sample space* $\{1, \ldots, s\}$*. Let us assume that*

$$\forall i \in \{1, \ldots, s\} : \forall m \in \mathbb{N} : \forall \delta \in (0,1] : \quad \mathbf{P}_{\mathsf{Z}^m} \left(\Upsilon_i \left(\mathbf{Z}, \delta \right) \right) \geq 1 - \delta \, .$$

Then, for all $m \in \mathbb{N}$ *and* $\delta \in (0,1]$*,*

$$\mathbf{P}_{\mathsf{Z}^m} \left(\bigwedge_{i=1}^{s} \Upsilon_i \left(\mathbf{Z}, \delta \mathbf{P}_{\mathsf{I}} \left(1 \right) \right) \right) \geq 1 - \delta \, .$$

A.4. Quantifier Reversal

The quantifier reversal lemma is an important building block for some PAC-Bayesian theorems (McAllester, 1998).

Lemma 2 (Quantifier reversal). *Let* X *and* Y *be random variables with associated probability spaces* $(\mathcal{X}, \mathcal{X}, \mathbf{P}_{\mathsf{X}})$ *and* $(\mathcal{Y}, \mathcal{Y}, \mathbf{P}_{\mathsf{Y}})$*, respectively, and let* $\delta \in (0,1]$*. Let* $\Upsilon : \mathcal{X} \times \mathcal{Y} \times (0,1] \to \{\text{true}, \text{false}\}$ *be any measurable formula such that for any* x *and* y *we have*

$$\{ \delta \in (0,1] \,|\, \Upsilon \left(x, y, \delta \right) \} = (0, \delta_{\max}]$$

for some $\delta_{\max} \in (0,1]$*. If*

$$\forall x \in \mathcal{X} : \forall \delta \in (0,1] : \ \mathbf{P}_{\mathsf{Y}|\mathsf{X}=x} \left(\Upsilon \left(x, \mathsf{Y}, \delta \right) \right) \geq 1 - \delta \, ,$$

then for any $\beta \in (0,1)$ *we have* $\forall \delta \in (0,1]$ *that*

$$\mathbf{P}_{\mathsf{Y}} \left(\forall \alpha \in (0,1] : \ \mathbf{P}_{\mathsf{X}|\mathsf{Y}=y} \left(\Upsilon \left(\mathsf{X}, y, (\alpha\beta\delta)^{\frac{1}{1-\beta}} \right) \right) \geq 1 - \alpha \right) \geq 1 - \delta \, .$$

References

Bartlett, P. and J. Shawe-Taylor: 1998, 'Generalization Performance of Support Vector Machines and other Pattern Classifiers'. In: *Advances in Kernel Methods — Support Vector Learning*. MIT Press, pp. 43–54.

Cannon, A., J. M. Ettinger, D. Hush, and C. Scovel: 2002, 'Machine Learning with Data Dependent Hypothesis Classes'. *Journal of Machine Learning Research* **2**, 335–358.

Cesa-Bianchi, N., A. Conconi, and C. Gentile: 2002, 'On the generalization ability of on-line learning algorithms'. In: *Advances in Neural Information Processing Systems 14*. Cambridge, MA, MIT Press.

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

Cover, T. M. and P. E. Hart: 1967, 'Nearest neighbor pattern classifications'. *IEEE Transactions on Information Theory* **13**(1), 21–27.

Cristianini, N. and J. Shawe-Taylor: 2000, *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge University Press.

Floyd, S. and M. Warmuth: 1995, 'Sample Compression, learnability, and the Vapnik Chervonenkis dimension'. *Machine Learning* **27**, 1–36.

Graepel, T., R. Herbrich, and J. Shawe-Taylor: 2000, 'Generalisation error bounds for sparse linear classifiers'. In: *Proceedings of the Annual Conference on Computational Learning Theory*. pp. 298–303.

Herbrich, R.: 2001, *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press.

Herbrich, R. and T. Graepel: 2002, 'A PAC-Bayesian margin bound for linear classifiers'. submitted to IEEE Transactions on Information Theory.

Herbrich, R. and R. C. Williamson: 2002, 'Algorithmic Luckiness'. *Journal of Machine Learning Research* **3**, 175–212.

Hoeffding, W.: 1963, 'Probability Inequalities for Sums of Bounded Random Variables'. *Journal of the American Statistical Association* **58**, 13–30.

Langford, J. and J. Shawe-Taylor: 2003, 'PAC-Bayes and Margins'. In: *Advances in Neural Information Processing Systems 15*. Cambridge, MA, pp. 439–446, MIT Press.

Littlestone, N.: 1988, 'Learning Quickly When Irrelevant Attributes Abound: A New Linear-treshold Algorithm'. *Machine Learning* **2**, 285–318.

Littlestone, N.: 1989, 'From on-line to batch learning'. In: *Proceedings of the second Annual Conference on Computational Learning Theory*. pp. 269–284.

Littlestone, N. and M. Warmuth: 1986, 'Relating Data Compression and Learnability'. Technical report, University of California Santa Cruz.

Marchand, M. and J. Shawe-Taylor: 2001, 'Learning with the Set Covering Machine'. In: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML'2001)*. San Francisco CA, pp. 345–352, Morgan Kaufmann.

McAllester, D. A.: 1998, 'Some PAC Bayesian Theorems'. In: *Proceedings of the Annual Conference on Computational Learning Theory*. Madison, Wisconsin, pp. 230–234, ACM Press.

McAllester, D. A.: 1999, 'PAC-Bayesian Model Averaging'. In: *Proceedings of the Annual Conference on Computational Learning Theory*. Santa Cruz, USA, pp. 164–170.

Rissanen, J.: 1978, 'Modeling by Shortest Data Description'. *Automatica* **14**, 465–471.

Rosenblatt, F.: 1962, *Principles of Neurodynamics: Perceptron and Theory of Brain Mechanisms*. Washington D.C.: Spartan–Books.

Shawe-Taylor, J., P. L. Bartlett, R. C. Williamson, and M. Anthony: 1998, 'Structural Risk Minimization over Data-Dependent Hierarchies'. *IEEE Transactions on Information Theory* **44**(5), 1926–1940.

Shawe-Taylor, J. and R. C. Williamson: 1997, 'A PAC Analysis of a Bayesian Estimator'. Technical report, Royal Holloway, University of London. NC2–TR–1997–013.

Tipping, M.: 2001, 'Sparse Bayesian Learning and the Relevance Vector Machine'. *Journal of Machine Learning Research* **1**, 211–244.

Vapnik, V.: 1998, *Statistical Learning Theory*. New York: John Wiley and Sons.

Vitànyi, P. and M. Li: 1997, 'On Prediction by Data Compression'. In: *Proceedings of the European Conference on Machine Learning*. pp. 14–30.

Warmuth, M.: 2003, 'Open Problems: Compressing to VC Dimension Many Points'. In: *Proceedings of the Annual Conference on Computational Learning Theory*.

Wyner, A. D., J. Ziv, and A. J. Wyner: 1992, 'On the role of pattern matching in information theory'. *IEEE Transactions on Information Theory* **4**(6), 415–447.