Exploring Generalisation Performance through PAC-Bayes

Felix Biggs

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

June 14, 2024

I, Felix Biggs, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Generalisation in machine learning refers to the ability of a predictor learned on some dataset to perform accurately on new, unseen data. Without generalisation, we might be able to memorise the training data perfectly while predicting poorly on new data, a pathology known as over-fitting. Despite its centrality, the generalisation behaviour of many methods remains poorly understood, particularly in complex domains such as deep learning. Indeed, some models that should over-fit according to traditional theoretical bounds do not. This thesis addresses these issues, particularly in the context of classification, and introduces innovative methods for producing non-vacuous generalisation bounds.

The primary thrust is in the development and application of PAC-Bayesian bounds, which are usually used as a method for studying the generalisation of *randomised* predictors. We begin with an introduction to previous work on the problem of generalisation and to PAC-Bayesian ideas, before applying these in work based on a series of five papers (referenced a-e below). Firstly in (a), we provide lower-variance methods for training stochastic neural networks methods, improving the use of these PAC-Bayes bounds as training objectives. Then, we use PAC-Bayes as a stepping stone to provide non-randomised bounds: (b) using margins, both in general and for several different classifiers; (c) for a specific class of deterministic shallow neural networks (where our bounds are the first to be non-vacuous on real-world data using standard training methods); (d) for majority voting on finite ensembles of classifiers, providing state-of-the-art (and sometimes *sharp*) guarantees. Lastly in (e), we introduce a PAC-Bayes bound for a modified excess risk, using information about the relative hardness of data examples to reduce variance and tighten a general bound.

Impact Statement

The importance and ubiquity of machine learning today is truly global and cannot be overstated but its theoretical underpinnings remain poorly understood; we provide a strong contribution towards undoing this shortfall. The research appearing in this thesis is based on first-author publications appearing in the journal Entropy (Biggs and Guedj, 2021), and in the top tier machine learning conferences AISTATS (Biggs and Guedj, 2022a, 2023), ICML (Biggs and Guedj, 2022b) and NeurIPS (Biggs, Zantedeschi, and Guedj, 2022). These all originally appeared on arXiv, ensuring accessibility and collaborative advancement in the field. As of publication, these papers have been over 90 times in follow-on research, with some later publications building directly on ideas introduced in it, including Clerico et al. (2022) and Fortier-Dubois et al. (2023). This influence is expected to continue to grow as the ramifications of the work become clearer in the fast-moving world of machine learning, and it goes on to impact wider academic disciplines.

In the longer term, it is my belief that such work will contribute to the development of more robust, reliable, and transparent machine learning systems, ultimately fostering trust. Such trustworthy systems would have wide-ranging implications for commercial activities like healthcare or finance, and in shaping public policy around AI ethics.

Please find the research paper declarations for included papers appended to the end of this thesis.

Acknowledgements

I'd like to thank my supervisor Benjamin Guedj for setting me onto the topic of learning theory, and for the advice and feedback of the last four years. Thanks to my other amazing coauthors, Valentina Zantedeschi and Antonin Schrab. Thank you to Marios Fournarakis, Anish Dhir and Eric Hambro for making machine learning fun in the first place. Endless thanks to my family and partner, Isabella, for the support that made it possible.

Contents

Notation

Symbol	Description
$\mathcal{A}$	Caligraphic capital letter: a set
$\notin$	Not an element of
$\exists$	Exists
$\forall$	For all
$\Rightarrow$	Implies
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of positive integers
$[n] = \{1, \dots, n\}$	Set of first $n \in \mathbb{N}$ natural numbers
Δ_{d-1}	d-dimensional simplex, $\{\boldsymbol{w} \in [0,1]^d : \sum_i w_i \leq 1\}$
$\mathbf{1}\{\mathcal{A}\}$ or $\mathbf{1}_{\mathcal{A}}$ or $\mathbf{1}_{a \in \mathcal{A}}$	Indicator function on a set $\mathcal{A}$
x_i	i-th index of vector $\boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{R}^d$
I	Identity matrix
$\|\boldsymbol{x}\|_p$	p-norm for a vector
$\|A\|_p$	p-norm for a matrix, $\|A\|_p = \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_p \leq 1} \|Ax\|_p$
$\|A\|_F$	Frobenius norm of a matrix, $\|A\|_F^2 = \sum_{ij} A_{ij}^2$
$\mathbb{E}$	Expectation
$\mathbb{E}_q$	Expectation w.r.t. distribution q
$\mathbb{E}_{X \sim q}$	Expectation w.r.t. $X \sim q$
$\mathbb{P}$	Probability
$\mathscr{P}(\mathcal{A})$	A set of appropriately well-behaved probability measures on $\mathcal{A}$
$\mathcal{N}(\boldsymbol{w}, \Sigma)$	Normal distribution with mean $\boldsymbol{w}$ and covariance Σ
Categ($\boldsymbol{w}$)	Categorical distribution with weights w_i for $\boldsymbol{w} \in \Delta_{d-1}$
$\mathcal{Z}$	Example space (for classification, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$)
$\mathcal{D}$	Data-generating distribution, $\mathcal{D} \in \mathscr{P}(\mathcal{Z})$
S	Sample, $S = (Z_1, \dots, Z_m)$ of m i.i.d. examples
m	Number of training examples, $m \in \mathcal{N}$
$\mathcal{W}$	Parameter space
ℓ	A generic (bounded) loss function, $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$
$\mathcal{L}(w)$	Population Risk of $w \in \mathcal{W}$, $\mathcal{L}(w) := \mathbb{E}_{Z \sim \mathcal{D}}[\ell(w, Z)]$
$\widehat{\mathcal{L}}(w)$	Empirical Risk of $w \in \mathcal{W}$, $\widehat{\mathcal{L}}(w) := m^{-1} \sum_{i=1}^m \ell(w, Z_i)$
$\mathbb{A}$	A machine learning algorithm, `A`
$\mathcal{X}$	Input space
d_{in}	Input dimension when $\mathcal{X} \subset \mathbb{R}^{d_{\text{in}}}$
$\mathcal{Y}$	Label space ($\{1, \dots, d_{\text{out}}\}$ or $\{+1, -1\}$ when $d_{\text{out}} = 2$)

d_{out}	Number of output classes in a classification problem
X_i, Y_i, Z_i	Examples in sample, typically $Z_i = (X_i, Y_i)$
$\ell_0, \mathcal{L}_0, \widehat{\mathcal{L}}_0$	Misclassification loss and associated risks

Chapter 1

Introduction

The evolution from classical statistics to modern machine learning is marked by a vast increase in the size of datasets and their dimensions. Machine learning systems use data to inform their predictions or decisions, reducing reliance on hard-coded rules or explicit programming. These shifts from hard coded rules and small to large datasets has enabled computers to tackle tasks that were previously considered infeasible or exceedingly complex.

Crucial to this transition is the development of methods which *generalise* on real world data: specifically, useful algorithms should output models which perform well not only on their training data, but also on *unseen* data from the same distribution. Without generalisation, datasets are no more useful than lookup tables for previously-seen values, and when new inputs are seen performance is correspondingly poor. Statistical theory shows that this is *in-general* difficult, yet on a huge array of *real-world* tasks many machine learning methods have empirically demonstrated excellent performance and generalisation. Exploring the gap between theory and observation is the topic of this thesis.

This has broader implications extending well beyond the realm of academic curiosity: as machine learning systems increasingly permeate all aspects of our lives, understanding the theoretical underpinnings of their performance becomes essential. It is my belief that such work will contribute to the development of more robust, reliable, and transparent machine learning systems, ultimately fostering trust and facilitating their adoption across a wide range of sectors. The importance and ubiquity of these tools in today's world cannot be overstated: in healthcare, they are used for disease detection, diagnosis, and prognosis; in finance, for credit risk assessment, fraud detection, algorithmic trading, and market trend forecasting; they have enabled technologies like speech recognition, machine translation, sentiment analysis, and chatbots; and they underpin the personalised recommendations we encounter daily in our digital lives, from the movies we watch on streaming platforms, the products we buy online, to the music we listen to. The breakthroughs keep coming, and the remarkably generic nature of machine learning methods means more applications are likely

to be found as tools improve ever further. As applications of machine learning continue to spread, so the case for putting its foundations on firmer theoretical ground grows.

We should note that this lack of theoretical understanding of *why* most machine learning methods work is in in contrast to most statistical methods, where the theory often comes first. Indeed, the predictions of classical statistical learning theory are often directly contradicted by empirical results obtained by modern machine learning tools. The traditional prevailing wisdom was that a model's ability to generalise is inversely linked to some measure of its "capacity", for example the number of parameters, or the (in)-ability to memorise adversarially-chosen data. However, many machine learning algorithms can perform well on real-world tasks far better than their oversized capacity would suggest. In this thesis, we primarily look to prove new *generalisation bounds* for machine learning methods in order to explain this so-called "generalisation mystery", leading to deeper understanding and the potential to obtain guarantees for methods.

Although we draw ideas from a wide variety of sources, the primary tool used to prove our bounds will be the PAC-Bayesian ideas introduced by Catoni (2003, 2004, 2007); McAllester (1998, 1999); Shawe-Taylor and Williamson (1997). These have recently seen a resurgence of interest through a variety of successful attempts to apply them to neural networks, beginning with Dziugaite and Roy (2017, 2018); Zhou et al. (2019). They provide a generic form of generalisation bound for randomised predictors; but the specific way in which we use them differs from chapter to chapter. Some recurring themes are: the use of bounds with specially-chosen randomised predictors as a stepping stone to prove new bounds for non-random predictors, either through ensemble-style learning or margins (a measure of confidence); the use of specific bounds as objectives to motivate new learning algorithms; and attempts to provide the sharpest empirical bounds, *i.e.* those with the smallest gap between the bound on performance and true performance as measured by a test set.

Research Contribution and Structure

During my PhD, I have primarily worked to formulate theoretical explanations of why generalisation happens in real-world settings, and to use PAC-Bayesian tools to work towards this goal. I began by addressing the question of optimising towards the tightest possible PAC-Bayesian bounds in the simple setting of binary classification, giving new methods for training the stochastic neural networks primarily used there that improved on existing methods. My next three works addressed the question of applying PAC-Bayesian tools to non-randomised classification methods, either through margins or majority votes. The focus here was on using randomised predictors that somehow approximate the behaviour of deterministic ones. First, I developed a generic framework for using PAC-Bayes as a stepping stone in proving margin bounds, proving several new ones. Next, I used the concept of

majority votes to obtain the first non-vacuous bounds for deterministic single-hidden-layer networks. Finally, I obtained new margin bounds for voting-style predictors, giving far tighter bounds than had previously existed in that setting. I completed this work at a more general level, proving a new generic PAC-Bayes bound which can make specific bounds sharper.

This thesis comprises a background chapter and five core chapters of research contributions, each based on a single publication. It is designed to be read by someone with some background knowledge of machine learning methods, particularly those for classification, including basic neural networks, gradient descent-based training methods, and various forms of linear prediction. Some knowledge of probability and statistics is also assumed, though most more technical details, such as concerns of measurability, are elided. An outline of the chapters is given below.

Chapter 2 Background.

We introduce the ideas of statistical learning theory with a particular focus on classification problems. We overview different types of bounds, from VC and Rademacher to margin-based and PAC-Bayesian. We finish with a brief overview of other approaches to generalisation bounds and work applying them to neural networks.

Chapter 3 Introduction to PAC-Bayes and Concentration.

Here we provide a much more technical introduction and overview of PAC-Bayes ideas. Particular emphasis is placed here on the techniques used for concentration of measure and to prove generic PAC-Bayes bounds. This culminates in a proof of Catoni and Maurer's PAC-Bayes bounds (Catoni, 2004; Maurer, 2004), which we motivate through the preceding treatment. These are the bounds primarily used in the later chapters, but by introducing the proof techniques in detail we are able to overview the topic of sub-Gaussian variables, used heavily in our margin-based bounds (Chapters 5 and 7) and lay the groundwork for the new generic bound we prove in Chapter 8.

Chapter 4 Differentiable PAC-Bayes Objectives with Partially Aggregated Neural Networks.

This chapter is based on the paper Biggs and Guedj (2021), which is published in Entropy. It examines the training of randomised neural networks for binary classification, particularly when they are being trained to optimise a PAC-Bayes bound. A new training method is introduced which leads to lower variances of gradients by considering the whole randomisation process; when applied to PAC-Bayes bounds for these randomised networks, it leads to tighter empirical guarantees than previous work. The primary ideas have been developed further by Clerico et al. (2022) to obtain tight guarantees in multi-class classification.

Chapter 5 On Margins and Derandomisation in PAC-Bayes.

This chapter is adapted from the publication Biggs and Guedj (2022a) which appeared at AISTATS 2022. It is the first of three chapters examining methods to obtain generalisation bounds for non-randomised predictors by using PAC-Bayesian methods applied to specially constructed randomised predictors as a stepping stone. Specifically, it introduces a method for proving margin bounds by constructing randomised predictors that concentrate around a non-random predictor. These are used to prove new bounds for two types of linear prediction, deep ReLU (Rectified Linear Unit) networks, and "SHEL" networks, which we introduce. These SHEL networks consist of a single hidden (Gaussian) error function layer, and arise as the average of a specially-constructed function.

Chapter 6 Non-Vacuous Generalisation Bounds for Shallow Networks.

This chapter is adapted from Biggs and Guedj (2022b) which featured at ICML 2022. It looks at providing generalisation bounds for deterministic single-hidden-layer neural networks by de-randomising PAC-Bayesian bounds. This is done through the framework of "majority votes" instead of using margins as above. Generalisation bounds for the previously-introduced SHEL networks and for GeLU-activated (Gaussian Error Linear Unit) single-hiden-layer networks are provided. In empirical evaluation these are the first non-vacuous bounds for deterministic networks trained using standard methods on real-world datasets.

Chapter 7 On Margins and Generalisation for Voting Classifiers.

This chapter is based on Biggs, Zantedeschi, and Guedj (2022) which appeared at NeurIPS 2022. It revisits the idea of margins for majority voting of finite classifier ensembles, proving new bounds by constructing a new, Dirichlet distribution–based randomised predictor. These provide state-of-the-art guarantees on a number of classification tasks, and are competitive with a test set on several of the task evaluations. By providing such tight bounds we add perspective to the debate on the "margins theory" proposed by Schapire et al. (1998) for the generalisation of ensemble classifiers.

Chapter 8 Tighter PAC-Bayes Generalisation Bounds by Leveraging Example Difficulty.

This chapter is based on Biggs and Guedj (2023) which was published at AISTATS 2023. Rather than focusing on a specific hypothesis class or even on classification, this work takes a more general approach of looking to tighten the generic PAC-Bayes bounds of *e.g.* Catoni (2007); Maurer (2004). It does this by introducing a modified version of the excess risk which leverages information about the relative hardness of data examples to reduce the variance of its empirical counterpart, tightening the bound.

We empirically show that this new bound can improve numerical tightness on a number of real-world datasets.

Additional work undertaken during the PhD. In addition to the above which forms a coherent and standalone research contribution, I additionally completed work which is not included in this thesis, since it does not relate in any way to understanding and optimising generalisation. In the arXiv note Biggs (2022) I give a generalisation of the inverse Chernoff bound often used for evaluation of PAC-Bayes bounds; this can lead to more computationally-efficient numerical evaluation. In the work Biggs, Schrab, and Gretton (2023) which appeared at NeurIPS 2023, we proposed a new two-sample statistical permutation test using maximum mean discrepancy; theoretical guarantees are given and favourable empirical power compared with similar tests is shown.

Notation. I have made a great effort to ensure consistency of notation throughout the thesis as a pedagogical aid. Despite this, there are some small differences. The most notable is that random vectors are denoted by capital letters in Chapters 2, 3 and 8, where linear algebra does not play a large role, while in the other chapters they are reserved for linear operators or matrices. Most of this notation is introduced in Chapter 2, and is recalled where necessary in each chapter; there is a reference table on page 15.

Chapter 2

Background

Here we introduce the main ideas of statistical learning theory as applied to classification. We introduce classes of linear, majority voting and neural network–based classifiers. We overview uniform convergence, PAC-Bayes and margin bounds, and their applications to the above classes.

We begin by outlining a fairly general statistical learning theory setting alongside its specialisation to classification. Our data lies in some space $\mathcal{Z}$, on which is defined some **data-generating distribution** $\mathcal{D} \in \mathscr{P}(\mathcal{Z})$ (denoting by $\mathscr{P}(\mathcal{A})$ an appropriate space[1] of probability measures defined on set $\mathcal{A}$). We are given access a (training) **sample** $S = (Z_1, \ldots, Z_m)$ of $m \in \mathbb{N}$ (the sample size) **examples**, $Z_i \in \mathcal{Z}$, assumed drawn independently and identically distributed (i.i.d.) from $\mathcal{D}$.

We wish to learn some parameter $w \in \mathcal{W}$ using S. Specifically, it should minimise the (distribution-dependent) **risk** of w, which is defined in terms of a (bounded) loss function, $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$, as

$$\mathcal{L}(w) := \mathop{\mathbb{E}}_{Z \sim \mathcal{D}}[\ell(w, Z)].$$

In other words, we wish to pick a parameter minimising the population (expected) loss. We do not consider the more general case of unbounded loss functions, and note that the bounding in $[0,1]$ is without loss of generality (w.l.o.g.) since we can always re-scale.

Since we have an i.i.d. set of samples, a natural place to begin is with the data-dependent **empirical risk**,

$$\widehat{\mathcal{L}}(w) := \frac{1}{m} \sum_{i=1}^{m} \ell(w, Z_i).$$

This is clearly an unbiased estimate of $\mathcal{L}(w)$ when w is independent of the sample (as when we are using a *test* set), but when w is chosen based on the sample, it is biased, as we discuss

[1] An unstated assumption beyond this point will be that this set, its corresponding sigma field, and any functions are constrained to make these measurability concerns immaterial, since such discussions add little pedagogical value in our problems of interest.

further below.

Test bounds. For now, we ask how well $\widehat{\mathcal{L}}(w)$ estimates $\mathcal{L}(w)$ in the unbiased case, where w is fixed independent of the sample. This is a so-called *test set bound*. Using Hoeffding's inequality (Hoeffding, 1956), for a fixed $w, \delta \in (0,1)$ and m,

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left(|\mathcal{L}(w) - \widehat{\mathcal{L}}(w)| \leq \sqrt{\frac{\log\frac{2}{\delta}}{2m}}\right) \geq 1-\delta. \tag{2.1}$$

In other words, we can upper bound the risk of w with high probability over the sample using $\widehat{\mathcal{L}}(w)$: this is a Probably Approximately Correct (PAC; Valiant, 1984) result. The high-probability part is vital, as it is always possible (if unlikely for large datasets) that we received a particularly "unlucky" and unrepresentative sample. This bound is reasonably tight: if we have 10000 samples and we are working at 99.9% confidence, then the square root term is less than 0.02. Somewhat tighter test set bounds also exist, as we discuss in Chapter 3.

2.1 Supervised Classification

We make the above concrete by considering a more specific setting, that of supervised classification, which is the setting of Chapters 4 to 7 (but not Chapter 8, which instead considers more generic bounded losses). Here we are given both an **input space**, $\mathcal{X}$ (where for many problems, $\mathcal{X} \subset \mathbb{R}^{d_{\text{in}}}$ for some input dimension d_{in}), and a **label space** $\mathcal{Y} = [d_{\text{out}}] = \{1, \ldots, d_{\text{out}}\}$, where d_{out} is the number of classes. A **binary** classification problem has two classes, $d_{\text{out}} = 2$; we will usually denote these $\{+1, -1\}$ instead of $\{1, 2\}$, as this makes some formulas simpler. Our overall data space is $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$; we will also assume that the data generating distribution factorises as $\mathcal{D} = \mathcal{D}_{\mathcal{X}} \otimes \mathcal{D}_{\mathcal{Y}|\mathcal{X}}$. We note the *noiseless* case where $\mathcal{D}_{\mathcal{Y}|\mathcal{X}}$ is deterministic, *i.e.* any x is unambiguously associated with a single label.

Our goal is to learn a **classifier**, $\text{class}_w : \mathcal{X} \to \mathcal{Y}$ indexed by w (with the overall space notated $\mathcal{H}$), that predicts the most likely class given an $x \in \mathcal{X}$ (and hence approximates the argmaximum of the "regression function" $\mathcal{D}_{\mathcal{Y}|\mathcal{X}}$). Examples of real world problems in this setting include image classification of *e.g.* digits (as per LeCun et al., 2010) or items of clothing (Xiao et al., 2017); here each input pixel is mapped and scaled into a component of $\mathcal{X} = [0,1]^{d_{\text{in}}}$, and the labels correspond to different digits 0–9 or previously-defined clothing types. A real world binary classification problem might be to take as input strings from SMS messages, and classify whether each is spam or not (Dua and Graff, 2017).

This is formalised through by specialising to the **misclassification loss** function, $\ell_0(\text{class}_w, (x,y)) = \mathbf{1}\{\text{class}_w(x) \neq y\}$ (denoting by $\mathbf{1}\{\mathcal{A}\}$ or $\mathbf{1}_{\mathcal{A}}$ the indicator function on set $\mathcal{A}$). In this case the risk becomes the probability of misclassifying a new (data-distribution

drawn) example,

$$\mathcal{L}_0(\text{class}_w) = \mathop{\mathbb{P}}_{(X,Y)\sim\mathcal{D}}(\text{class}_w(X) \neq Y).$$

Note that this means that incorrect predictions on regions of $\mathcal{X}$ with low probability under $\mathcal{D}_\mathcal{X}$ do not necessarily increase the risk by much. The empirical risk is the proportion of incorrect examples on the dataset,

$$\widehat{\mathcal{L}}_0(\text{class}_w) = \frac{|\{(x,y) \in S : \text{class}_w(x) \neq y\}|}{m}.$$

Thus this setting relates to the above more general setting by putting $\ell(w,(x,y)) = \ell_0(\text{class}_w,(x,y))$.

2.1.1 Basic Classifiers

Here we introduce the basic classifiers we primarily consider in this thesis, without mentioning the algorithms used to learn them.

All of the classification methods we directly consider in this thesis are *score-based*. This means for a given $x \in \mathcal{X}$, they output an intermediate set of values in $\widehat{\mathcal{Y}}$. For multi-class classification, $\widehat{\mathcal{Y}} = \mathbb{R}^{d_{\text{out}}}$, and for binary classification $\widehat{\mathcal{Y}} = \mathbb{R}$; the classification prediction is the argmaximum index for the multi-class case, or the sign in the binary case. Specifically, in the multi-class case our prediction functions can be defined by some (w-indexed) function $f_w : \mathcal{X} \to \mathbb{R}^{d_{\text{out}}}$, and the overall prediction is given by $\text{class}_w(x) = \text{argmax}_{y\in\mathcal{Y}} f_w(x)[y]$ (with the $[y]$ notation denoting the yth component); in the binary case $f_w : \mathcal{X} \to \mathbb{R}$ and $\text{class}_w(x) = \text{sign}(f_w(x))$.

We denote a space of such functions by $\mathcal{F}$, with its exact definition (and that of $\widehat{\mathcal{Y}}$) being contextual. Note that, just as the same weight $w \in \mathcal{W}$ might lead to identical score functions $f_w \in \mathcal{F}$, so can different f_w lead to identical classification outputs and thus $\text{class}_w \in \mathcal{H}$. Hence $\mathcal{H}$ is a quotient space of $\mathcal{F}$ and $\mathcal{F}$ of $\mathcal{W}$, so we will often be somewhat loose in distinguishing these spaces, writing for example $\mathcal{L}_0(f_w) = \mathcal{L}_0(\text{class}_w)$.

Binary linear prediction. One of the most fundamental forms of classifiers is linear prediction, where $\mathcal{X}$ is in a vector space, and our score output is $f_{\boldsymbol{w}}(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ for $\boldsymbol{w} \in \mathcal{W}$ also in a (dual) vector space. We often assume the restriction of $\mathcal{X}$ and $\mathcal{W}$ to some origin-centred balls. Since the classification output is the sign of $f_{\boldsymbol{w}}$, it effectively divides $\mathcal{X}$ into two half-spaces with opposite labels. Note this classifier can be considerably extended by first mapping all data through some fixed feature mapping, $\phi : \mathcal{X} \to \mathcal{X}'$, which can potentially be seen as a pre-processing step. In this case, $\mathcal{X}$ might not be a vector space, while $\mathcal{X}'$ could potentially be an *infinite* dimensional space (where the infinite spaces are taken care of implicitly, *e.g.* through kernel methods).

Weighted majority votes. Here we have a set of "base" classifiers $\mathcal{H}_{\text{base}}$, where each $h \in \mathcal{H}_{\text{base}}$ is a function $h : \mathcal{X} \to \mathcal{Y}$. A majority vote is defined by a distribution $Q \in \mathscr{P}(\mathcal{H}_{\text{base}})$. In the case where $|\mathcal{H}| = d_{\text{vot}} < \infty$, this can be characterised directly by a weight vector $\boldsymbol{w} \in \Delta_{d_{\text{vot}}-1}$, where $\Delta_{d-1} = \{\boldsymbol{a} \in [0,1]^d : \sum_i a_i \leq 1\}$ is the d-dimensional simplex. The (scored) majority vote (MV) then outputs

$$f_{\boldsymbol{w}}(x)[k] = \mathop{\mathbb{E}}_{h\sim Q}[\mathbf{1}\{h(x) = k\}] = \sum_{h\in\mathcal{H}_{\text{base}}} w_i \mathbf{1}\{h(x) = k\}$$

for each $k \in \mathcal{Y}$. The implied classifier is $\text{MV}_{\boldsymbol{w}}(x) := \text{argmax}_{k\in\mathcal{Y}} f_{\boldsymbol{w}}(x)[y]$. When the base classifier space is infinite, the sum in $f_{\boldsymbol{w}}$ is replaced by an integral, so the definition is $f_Q(x)[k] = \mathbb{P}_{h\sim Q}(h(x) = k)$.

Feed-forward neural networks. Deep neural networks combine linear transformations by learned parameters with various fixed non-linearities. When used for classification, they are usually score-based, with output in $\mathbb{R}^{d_{\text{out}}}$ (or potentially $\mathbb{R}$ when considering binary classification). A feed-forward network takes the form

$$f_{W_1,\ldots,W_L}(\boldsymbol{x}) = W_L\phi_{L-1}(W_{L-1}\phi_{L-2}(\ldots\phi_1(W_1\boldsymbol{x})\ldots)),$$

for fixed-size matrices $W_1, \ldots, W_L$ and *fixed* activation or "pooling" functions ϕ_i. For example, in a fully-connected ReLU (Rectified Linear Unit) network, the matrix entries are unconstrained and the activation functions are element-wise applied ReLU functions, $\text{ReLU}(t) = t\mathbf{1}\{t \geq 0\}$. Another feed-forward variant is convolutional neural networks (Fukushima, 1980), which effectively constrain the weight matrices in a special way to ensure translation-invariance. Recurrent (Rumelhart et al., 1985) or attention-based (Vaswani et al., 2017) networks take other forms which can be considerably more complex.

2.2 Worst Case Bounds

Suppose that the space $\mathcal{W}$ has been chosen for us by a domain expert, and we have no prior knowledge about the data distribution $\mathcal{D}$. We wish to investigate how well we can choose w based on the sample to optimise $\mathcal{L}(w)$. An **algorithm**, $\mathbb{A} : \mathcal{Z}^m \to \mathcal{W}$, takes the sample and outputs a parameter. Without prior knowledge, we would like to choose $\mathbb{A}$ to work as well as possible regardless of the data distribution, $\mathcal{D}$. We are therefore in some sense studying the **worst-case** behaviour, since we are concerned about every $\mathcal{D}$ equally, even those that have been chosen by an adversary.

Empirical Risk Minimisation. A natural choice of algorithm is *empirical risk minimisation*, which chooses a predictor $w_{\text{ERM}} \in \text{argmin}_{w\in\mathcal{W}} \widehat{\mathcal{L}}(w)$, called an empirical risk minimiser or *ERM* (note that it may not be unique). Studying $\mathcal{L}(w_{\text{ERM}})$ turns out to be considerably

more complex than just looking at $\widehat{\mathcal{L}}(w_{\text{ERM}})$, due to the aforementioned bias of this term. The result in Eq. (2.1) is not applicable, because w_{ERM} depends on the sample, and $\widehat{\mathcal{L}}(w_{\text{ERM}})$ is a biased estimate of $\mathcal{L}(w_{\text{ERM}})$. This bias creeps in because if $|\mathcal{W}|$ is large, then it is likely that at least one $w \in \mathcal{W}$ leads to a large statistical fluctuation in $\widehat{\mathcal{L}}(w)$, and thus underestimate $\mathcal{L}(w)$; the ERM is likely to pick such a value and hence maximise the bias.

The problem of generalisation. This problem can be truly pathological, as we show through the following example. Consider ERM in a binary classification problem on $\mathcal{X} = [0, 1]$, with $w \in \mathcal{W}$ indexing the entire set of binary classification functions on $\mathcal{X}$. Suppose the data distribution is uniform on $\mathcal{X}$ and fixes $y = 1$ everywhere. Since the problem is noiseless, an ERM returns a predictor matching any training data observed. However, since $\mathcal{W}$ indexes all possible deterministic mappings $\mathcal{X} \to \mathcal{Y}$, we must choose how it behaves elsewhere. A priori we have no reason to favour or disfavour any over another, we could choose it to predict $y = -1$ at every x not observed in the sample. This ERM has a risk of 1, since the sample has measure zero under $\mathcal{D}_{\mathcal{X}}$.

In the above, not only is w_{ERM} a poor parameter choice, but $\widehat{\mathcal{L}}(w_{\text{ERM}}) \ll \mathcal{L}(w_{\text{ERM}})$ is a poor indicator of performance; we call this *over-fitting*. We also say that ERM (the algorithm) on a rich parameter space does not **generalise**. This latter term is slightly overloaded, as we could say an algorithm generalises either when $\mathcal{L}(w)$ is low (but still potentially much higher than $\widehat{\mathcal{L}}(w)$), or when it does not over-fit; we lean towards the former definition but may be somewhat loose.[2]

No-free-lunch. A natural question is if we can do better than the ERM. Unfortunately we cannot in our worst-case scenario: there is no universal learner, in the sense that no algorithm succeeds on every task. This fact is proved by several no-free-lunch theorems. For example, consider a binary classification task on $\mathcal{X}$ with sample size $m < |\mathcal{X}|/2$. Shalev-Shwartz and Ben-David (2014) show that for any algorithm, $\mathbb{A}$, there exists a data distribution, $\mathcal{D}$, such that both $\mathcal{L}(h^*) = 0$ for some h^*, and

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left(\mathcal{L}(\mathbb{A}(S)) \geq \frac{1}{8} \right) \geq \frac{1}{7}.$$

Thus, there is a classifier that succeeds perfectly on the task (*i.e.*, h^*), while our algorithm has a constant probability of failing to get near the minimum risk. When the space $\mathcal{X}$ is infinite, increasing the sample size will not even fix this problem. Therefore, on an unrestricted space $\mathcal{H}$, *any* algorithm that we choose will fail on some problem that another learner (*e.g.*, one that simply outputs h^*) will easily solve.

[2] A related concept is that of *under*-fitting; here $\mathcal{W}$ is relatively un-expressive or poorly matched to the data distribution, so that even choosing the optimal $w \in \mathcal{W}$ results in a high risk $\mathcal{L}(w)$, while other spaces $\mathcal{W}$ might give a far lower minimum risk.

2.2.1 ERM can work!

However, the above examples could be seen as somewhat contrived, since they require a large and expressive space $\mathcal{W}$, that includes every possible classification function $\text{class}_w : \mathcal{X} \to \mathcal{Y}$. None of our classifiers in Section 2.1.1 have these properties. The no-free-lunch theorem shows that an ERM will succeed perfectly on the classifier space including only $\text{class}_w = h^*$. If our $\mathcal{W}$ is more restricted as in Section 2.1.1, but still contains a w achieving low risk, will ERM work?

Uniform convergence. A sufficient condition for ERM's consistency (in that it converges to the minimum risk solution as $m \to \infty$) is *uniform convergence*[3] (Vapnik, 1991). This requires $\widehat{\mathcal{L}}(w) \to \mathcal{L}(w)$ simultaneously for all $w \in \mathcal{W}$ (in probability as $m \to \infty$). More formally,

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left(\exists w \in \mathcal{W} : |\mathcal{L}(w) - \widehat{\mathcal{L}}(w)| > \epsilon\right) \to 0 \quad \text{as} \quad m \to \infty$$

at every $\epsilon > 0$. We next examine some cases in which we have uniform convergence.

A training set union bound. When the class $\mathcal{W}$ is finite, we can directly apply Eq. (2.1) to each $w \in \mathcal{W}$ with $\delta' = \delta/|\mathcal{W}|$, where $|\mathcal{A}|$ is the cardinality of set $\mathcal{A}$. By the union bound, $\mathbb{P}\left(\cup_{w\in\mathcal{W}} : |\mathcal{L}(w) - \widehat{\mathcal{L}}(w)| > \sqrt{\log(2|\mathcal{W}|/\delta)/2m}\right) < \sum_{w\in\mathcal{W}} \delta/|\mathcal{W}| = \delta$, and therefore

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left(\forall w \in \mathcal{W} \ : \ |\mathcal{L}(w) - \widehat{\mathcal{L}}(w)| \le \sqrt{\frac{\log(\frac{2|\mathcal{W}|}{\delta})}{2m}}\right) \ge 1 - \delta. \tag{2.2}$$

Therefore a finite hypothesis class has the uniform convergence property at sample size *rate* $\mathcal{O}(1/\sqrt{m})$, and hence consistency of ERM on $\mathcal{W}$. This is our first PAC bound for a training set, and we see that it can be used both to estimate the error $\mathcal{L}(w)$ without a test set, and to understand when a particular algorithmic approach will work. It tells us how large the error can be based on the size of the sample, and that $\widehat{\mathcal{L}}(w_{\text{ERM}}) \approx \mathcal{L}(w_{\text{ERM}})$ with high probability (w.h.p.) whenever $\log|\mathcal{W}| \ll m$. The bound itself tells us that ERM with a small finite $\mathcal{W}$ will get close to the minimum risk solution.

However, the need for $\log|\mathcal{W}| < m$ can be quite restrictive. The classifiers we are interested in have real-valued parameters, and thus an infinite $\mathcal{W}$. In reality, floating point operations on a computer are done at finite precision, but the number of bits needed to represent our classifiers may be much larger than m (and even a single parameter represented by n bits implies $|\mathcal{W}| = 2^n$). Further, we likely do not want to use algorithms which technically rely on the details (and specific implementation) of floating point arithmetic, so we should be able to analyse learning methods without relying on this limited precision. Note however

[3] Shalev-Shwartz et al. (2010) show that this is not actually a necessary condition for certain non-trivial learning problems as specified by triples $(\ell, \mathcal{W}, \mathcal{Z})$, though it is in the context of binary supervised classification.

that many algorithms work well even in deliberately low-precision arithmetic, which is a first indication of the deep links between *compressibility* and generalisation.

VC Dimension. More sophisticated uniform bounds for binary classification use the Vapnik-Chervonenkis (VC) dimension (Vapnik and Chervonenkis, 1971). In fact, in this simple setting, it turns out that a finite VC dimension is necessary and sufficient for both uniform convergence and the consistency of ERM. Sufficiency follows from the simple bound (in presentation, not proof, since this is a chaining-based bound; Lugosi, 2002):

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left(\forall h\in\mathcal{H}\,:\,|\mathcal{L}_0(h)-\widehat{\mathcal{L}}_0(h)|\leq C\sqrt{\frac{\mathrm{VC}(\mathcal{H})+\log\frac{2}{\delta}}{m}}\right)\geq 1-\delta, \tag{2.3}$$

where $C>0$ is some constant. $\mathrm{VC}(\mathcal{H})$ is the VC dimension, that describes the "capacity" of the class $\mathcal{H}$ to fit adversary-chosen labels.

Equation (2.3) matches a lower bound arising for some adversarial data distribution (Anthony and Shawe-Taylor, 1993; Kontorovich and Pinelis, 2016; Vapnik, 1999). It has also been extended to multi-class classification (Natarajan, 1989), and numerical improvements to the base VC bound have been made by Leboeuf et al. (2021). In particular, the square root term in the VC bound can be improved to $\widetilde{\mathcal{O}}(\mathrm{VC}/m)$ for w such that $\widehat{\mathcal{L}}_0(w)=0$ (the "realisable case"; note also the "relative deviation" bound of Anthony and Shawe-Taylor, 1993; Vapnik and Chervonenkis, 1971; Vapnik, 1999), and we can get "intermediate" rates of $\widetilde{\mathcal{O}}((\mathrm{VC}/m)^\alpha)$ with $\alpha\in(\frac{1}{2},1)$ under additional assumptions (*e.g.* on noise Boucheron et al., 2005; Mammen and Tsybakov, 1999). Most of these bounds can be proved and refined further by considering Rademacher (and Gaussian) complexities (Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2000) or covering arguments (see Anthony et al., 1999).

Learning our classifiers. For binary linear classification with feature space $\mathbb{R}^{d_{\mathrm{hid}}}$, the VC dimension is $d_{\mathrm{hid}}+1$; for (binary) majority votes on $\mathcal{H}_{\mathrm{base}}$ it is upper bounded by $\widetilde{\mathcal{O}}(T\times\mathrm{VC}(\mathcal{H}_{\mathrm{base}}))$ where T is the number of non-zero weighted voters; for sign-function activated feed-forward neural networks it is $\widetilde{\Theta}(\text{number of parameters})$ (Shalev-Shwartz and Ben-David, 2014). Each of these are therefore learnable by an ERM with enough data (*i.e.* $m\gg\mathrm{VC}$).

Implementing ERM. A small but important concern is implementation: learning an ERM can be computationally complex, and the ERM is rarely *unique*. ERM for linear classification is known to be computationally hard (Ben-David and Simon, 2000) in general, though in the noiseless and separable case (where there exists at least one $\boldsymbol{w}\in\mathcal{W}$ achieving $\mathcal{L}_0(f_{\boldsymbol{w}})=0$) it can be solved through either linear programming or the Perceptron algorithm (Rosenblatt, 1958). For this reason, many popular algorithms for this class only output approximately error

minimising solutions, generally by optimising some convex objective function; for example in the support vector machine (Cortes and Vapnik, 1995). For majority votes, boosting also minimises some convex objective (Freund and Schapire, 1997; Schapire, 1990). The error of these nearly-ERM solutions can sometimes still be bounded by uniform convergence–style arguments (Boucheron et al., 2005; Zhang, 2004).

Implementation is particularly challenging in deep neural networks, which are highly non-convex, so finding the ERM may not be feasible, and most training methods for this case employ some variant of gradient descent on a smooth surrogate loss function. Even when interpolating the data (*i.e.*, achieving zero train error) is possible, there may be very many different possible parameters for this, and hence different ERMs, with potentially different performance. However, this non-uniqueness of the ERM only matters in the low-data regime[4], and is not a problem for the case $\mathrm{VC} \ll m$ since Eq. (2.3) applies to every h simultaneously.

No better algorithm. The bound Eq. (2.3) is two-sided and holds for any w (and hence any algorithmic output), so in the regime $\mathrm{VC} \ll m$ we really have $\mathcal{L}(w) \approx \widehat{\mathcal{L}}(w)$ with high probability. The improved VC bound variants only increase the rate of this convergence, making the empirical risk an even better proxy for the true risk. Therefore no other algorithm can improve on ERM in this regime, at least by more than a small correction $\mathcal{O}(m^{-\frac{1}{2}})$ term; the improved bounds only show that this is also true sometimes for even less data. The only thing that ultimately differs between $\mathcal{D}$ in these bounds is the risk of the best $w \in \mathcal{W}$, and any consistent algorithm must be (asymptotically) an ERM.

Various lower bounds (*e.g.* Shalev-Shwartz and Ben-David, 2014, Theorem 6.7) show that $m > \Omega(\mathrm{VC}(\mathcal{W}))$ is a necessary condition for general learning of $\mathcal{W}$, in the sense that with high probability $\mathcal{L}(\mathbb{A}(S)) \to \inf_{w \in \mathcal{W}} \mathcal{L}(w)$ for *any* $\mathcal{D}$. In this case, they also show that ERM is a successful algorithm. We also have the no-free-lunch theorems showing that learning on a class with no "complexity control" is not possible in general.

Therefore, in the worst-case situations where we must cover every $\mathcal{D}$ with equal importance, there are two possibilities: either we have enough data to solve the learning problem (requiring $m \gg \mathrm{VC}$), and the ERM is successful; or we do not, and learning is not possible in general as shown by the no-free-lunch theorems.

2.3 Beyond Worst Case

The two-sided bound Eq. (2.3) shows that when $\mathrm{VC} \ll m$ and asymptotically we cannot improve on the ERM; no-free-lunch theorems suggest that we cannot in-general do anything outside in this case. Have we therefore solved the generalisation puzzle? No, for a variety of

[4]Although technically, in multi-class classification where d_{out} is allowed to grow with m, there are settings where one ERM may be provably better than another regardless of data distribution (Daniely et al., 2011).

reasons. The point linking all of them is that the above bounds are all **worst-case**, which is not the usual setting in which machine learning is actually applied.

Machine learning practitioners often successfully apply algorithms without $\mathrm{VC} \ll m$: for example, an SVM with infinite VC dimension successfully learns many real-world data distributions, and neural networks with many more parameters than data often generalise well. In many boosting methods, we reach an h obtaining zero empirical risk, then modify it so that the zero empirical risk is maintained but the VC dimension is *increased*; we repeat this process iteratively, but despite the increasing capacity, the true risk of each new h tends to decrease. Therefore, VC bounds do not "explain" the generalisation which is commonly seen on real world datasets.

This happens because real-world data distributions tend to have properties which make them easier to learn than adversary-chosen ones (as required for no-free-lunch theorems), and we often have some domain knowledge about what kind of predictors are likely to work well. The free-lunch theorems (and test bound) show that if we choose our algorithm to be well-matched to our problem, we can do very well. Fortunately, it turns out that some algorithms have biases which are actually useful for a very wide variety of real-world tasks.

Approximation vs estimation trade off. Suppose we are given a class $\mathcal{W}$ that is too large for our dataset: *i.e.* $\mathrm{VC}(\mathcal{W}) > m$. A classical approach is to restrict $\mathcal{W}$ to some less complex set $\mathcal{W}'$. We can control the **estimation error** of $\mathcal{L}(w) - \widehat{\mathcal{L}}(w)$ over the smaller class $\mathcal{W}'$, since if this smaller class has *e.g.* $\mathrm{VC}(\mathcal{W}') \ll m$ we can simply find an ERM within it. The price we pay for this method is the **approximation error** $\mathcal{L}(w) - \inf_{w^* \in \mathcal{W}} \mathcal{L}(w^*)$, which will be large if we choose $\mathcal{W}'$ poorly for our problem, or too small.

We can then let $\mathcal{W}'$ grow depending on m or some sample-dependent quantities (which could also potentially enable consistency even if the optimal w is not in one of the smaller $\mathcal{W}'$). For example, depending on the "difficulty" of fitting the sample, which leads to ideas of structural risk minimisation (Vapnik, 1999; Vapnik and Chervonenkis, 1974) and minimum description length (Rissanen, 1978, 1983), which relates to compression, as discussed above. This is also in some sense the dual of the regularisation approach (as first popularised by Tikhonov et al., 1943), which considers instead minimising some functional $\widehat{\mathcal{L}}(w) + \mathrm{pen}(w)$, with $\mathrm{pen}(w)$ a weight-dependent penalty, and subsumes a wide variety of methods. Finally, some algorithms might be implicitly (*i.e.* without being the solution of an explicitly regularised optimisation problem) biased towards regularised or short-description-length solutions, automatically trading off between estimation and approximation errors; this is suggested to be behind the performance of boosting, stable algorithms (as discussed as far back as Rogers and Wagner, 1978, see Devroye et al., 2013 for a more generic reference), and sometimes SGD on neural networks.

It is important to note that the above approach necessarily favours some $w \in \mathcal{W}$ over others, and therefore by improving performance on some data distributions we worsen it on others. We need to use some a priori knowledge of likely $\mathcal{D}$ to ensure it works. Fortunately, it turns out that many real world datasets have properties in common, *e.g.* certain types of smoothness, which makes it possible for fairly generic algorithms to perform well across a variety of real world tasks.

Not all ERMs are created equal. On neural networks, we often find a set of parameters giving zero empirical loss, so the solution is an ERM. Other ERMs usually also exist which have far worse generalisation, as we can see through the following: Zhang et al. (2021, in Figure 1(c)) fit a large neural network to zero training error on CIFAR-10 (Krizhevsky, 2009) with 50% randomly corrupted labels, so $S = S^{\text{corrupt}} \cup S^{\text{uncorr}}$, which is clearly an ERM on $S^{\text{uncorr}} \overset{\text{iid}}{\sim} \mathcal{D}$. This network obtains a test error on $\mathcal{D}$ of ≈ 0.7. The same network is also trained to perfectly fit the uncorrupted sample without using S^{corrupt}, and obtains a test error of ≈ 0.2. Both weight solutions are clearly ERMs since both achieve $\widehat{\mathcal{L}}(w) = 0$ on $S^{\text{uncorr}} \overset{\text{iid}}{\sim} \mathcal{D}$, but they achieve completely different risks, a fact which clearly cannot be explained by VC-based bounds.

The above can potentially be explained in terms of an approximation/estimation trade off: when fitting S with zero empirical risk, we end up minimising over a larger space of functions than we do when fitting only S^{uncorr}, and so we incur a much greater estimation cost for the former. However, there are similar cases in which it can be shown the space is too large to control the estimation error by uniform convergence. Nagarajan and Kolter (2019) show that for linear classifiers and neural networks there are settings where *any* bounds derived via uniform convergence fail (*i.e.* are near-**vacuous**, the bound on $\mathcal{L}(w)$ being greater than 1), even when the stochastic gradient descent commonly used in these settings provably generalises. Technically, these show that non-vacuous bounds cannot be constructed by bounding the estimation part of the error through a uniform convergence bound, even when we are aware of the data distribution and take account of implicit algorithmic bias to the maximum extent possible.

Non-uniform bounds. We have seen that uniform convergence bounds essentially fully characterise the worst-case behaviour of learners and motivate when generic ERM algorithms should be used. In this thesis we work towards proving new and different bounds that involve more information about the learning problem, in order to understand what qualities lead to good generalisation and performance for different algorithms and data distributions outside of the worst case. A general learning theory approach is to prove finite-sample PAC bounds

on $\mathcal{L}(\mathbb{A}(S))$ for different algorithms or weights, specifically results taking the general form

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}(\mathcal{L}(\mathbb{A}(S)) \leq \text{bound}) \geq 1-\delta \tag{2.4}$$

for some *bound*. This bound term may depend on $\widehat{\mathcal{L}}(\mathbb{A}(S))$ as well as potentially other quantities related to S and the algorithm.

We will primarily consider two types of bound: PAC-Bayesian, and margin bounds, both of which we discuss further below. In fact, the margin bounds we prove (in Chapters 5 and 7) actually use PAC-Bayes as an intermediate step. They will hold for any (i.i.d.) data and algorithm, but will be **non-uniform** so that the tightness is affected by the choice of w (unlike in the uniform convergence bounds which hold "uniformly" over all $w \in \mathcal{W}$, or algorithmic bounds that hold only for a w output by a specific algorithm).

Alternative approaches we do not closely consider include compression-scheme based bounds (Littlestone and Warmuth, 1986), stability or regularisation–based approaches (see Devroye et al., 2013; Shalev-Shwartz and Ben-David, 2014, and references therein), and (conditional) mutual information based bounds (which generally hold only in-expectation; when not, they relate very closely to PAC-Bayes) as developed by, *e.g.* Grünwald et al. (2021); Steinke and Zakynthinou (2020); Xu and Raginsky (2017).

2.3.1 Margin Bounds

Margin-based bounds are some of the most classical examples of *non-uniform* bounds, with proofs utilising them as far back as Novikoff (1962) for the Perceptron algorithm. Mostly, they are proved by considering Rademacher-complexity based arguments, the fat-shattering dimension (which relates most closely to VC dimension), or by using PAC-Bayes bounds as an intermediate step.

Linearly seperable data. Suppose our data is linearly separable, so for some $\boldsymbol{w}$, $\mathcal{L}(\text{class}_{\boldsymbol{w}}) = 0$, using a binary linear classifier, $\text{class}_{\boldsymbol{w}}(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w}\rangle$; but the dimension $d_{\text{in}} \gg m$, so the bound Eq. (2.3) or the faster-rate realisable version is vacuous. It turns out we can still get high probability bounds if the sample is separable by some margin γ. Specifically, our $\boldsymbol{w}$ defines some hyperplane that divides the space $\mathcal{X}$ into half-spaces of labels $+1$ and -1: if all the data a distance of at least $\gamma > 0$ from this plane, we say it can be separated at margin γ. This is a property of both the sample and the chosen $\boldsymbol{w}$. Note that any $\boldsymbol{w}$ for which this is possible is also an ERM, since the data is also classified exactly.

Assume that $\|\boldsymbol{x}\|_2$ and $\|\boldsymbol{w}\|_2$ are bounded for all $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{w} \in \mathcal{W}$. We show in Chapter 5 (Biggs and Guedj, 2022a, originally) that

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left(\forall \boldsymbol{w} \in \mathcal{W} \,:\, \mathcal{L}_0(\boldsymbol{w}) \leq C\frac{\gamma_*(\boldsymbol{w},S)^{-2}\log m + \log\frac{1}{\delta}}{m}\right) \geq 1-\delta,$$

where

$$\gamma_*(\boldsymbol{w}, S) = \sup\left\{\gamma > 0 : \min_{(\boldsymbol{x},y)\in S} \langle y\boldsymbol{x}, \boldsymbol{w}\rangle > \gamma\right\}$$

depends on both $\boldsymbol{w}$ and S.

Therefore, if we choose our algorithm to output a $\boldsymbol{w}$ maximising the margin γ_* as much as possible, we will get a faster reduction of the population risk. This applies even for high or infinite dimensional $\mathcal{X}$, justifying the use of the SVM (Cortes and Vapnik, 1995), which outputs an approximately margin-maximising solution.

Of course, the bound depends on the data distribution having certain properties, and adversarial distributions will not be separable. It gives improved rates of convergence for favourable distributions, and shows that in the "small" data regime, not all ERMs are equal. This would be a moot point, as the free-lunch theorem already demonstrates such, except that it turns out this property is common on a fairly wide variety of real-world problems. The reason machine learning works at all is because it turns out that many real-world problems appear to have properties which can be exploited by fairly generic learners, so we usually do not need to over-specialise our algorithms or rely on extremely large datasets.

General margins. Building on the above leads to a more general idea of margins, which quantify how strongly we prefer one prediction has been made over another, in some ways a measure of confidence. On an example (x, y), the margin is defined as

$$M(f_w, x, y) = f_w(x)[y] - \max_{y' \neq y} f_w(x)[y'],$$

or in the binary case, $M(f_w, x, y) = y f_w(x)$ (with $y \in \{+1, -1\}$; note these differ by a factor of 2 for consistency with the literature). The **margin loss**, $\ell_\gamma(w, (x, y)) = \mathbf{1}\{M(f_w, x, y) \leq \gamma\}$ for $\gamma \geq 0$, quantifies how often predictions are wrong or correct but with only low margin (as a proxy for low confidence). We can use this to extend the above results to cases where not all of the data is actually separable at margin γ. We define analogously with $\mathcal{L}_0$ and $\widehat{\mathcal{L}}_0$ the margin risks $\mathcal{L}_\gamma$ and $\widehat{\mathcal{L}}_\gamma$; note that the misclassification loss is the margin loss at $\gamma = 0$, which motivates its notation of ℓ_0.

For linear binary classification and majority votes. To consider the margin without arbitrary scaling, we constrain the sets $\mathcal{X}$ and $\mathcal{W}$: for example, an L_2 constraint, where we set $\|\boldsymbol{x}\|_2 \leq 1, \|\boldsymbol{w}\|_2 \leq 1$; or an L_1/L_∞ constraint, where we set $\|\boldsymbol{x}\|_\infty \leq 1, \|\boldsymbol{w}\|_1 \leq 1$. This extends to the case where we are using a fixed feature map first. We already discussed the SVM, which uses the L_2 constraint; while L_1/L_∞-constrained algorithms that approximately maximise some margin include varieties of boosting (Freund and Schapire, 1997; Schapire, 1990) and bagging (Breiman, 2001, *e.g.* random forest), with such bounds also being applied to majority votes for binary classification.

In Chapter 5 we work to obtain tighter margin bounds for this setting, both under L_2 and L_1/L_∞ norms. In Chapter 7 we prove a new margin bound for majority votes that is seen to be empirically extremely tight, effectively explaining the generalisation we see on the datasets to which we apply it.

For neural networks. Finally, similar bounds have been proved for deep neural networks (see *e.g.* Bartlett et al., 2017; Neyshabur et al., 2018). Here, the bound complexity term usually depends on various norms of the weights divided by the margin. Unfortunately, these bounds are typically vacuous on real-world datasets. It may be that any such bounds proved via uniform convergence–based arguments (which includes fat-shattering, Rademacher, and some PAC-Bayes de-randomisation techniques) must fail in some settings where SGD provably generalises (Nagarajan and Kolter, 2019). However, the idea that neural networks generalise because they achieve a large margin on their datasets still has some traction, and we prove two different margin bounds for neural networks in Chapter 5.

2.3.2 PAC-Bayes Bounds

PAC-Bayesian bounds (originated by Catoni, 2003, 2004, 2007; McAllester, 1998, 1999; Shawe-Taylor and Williamson, 1997, we refer to recent surveys in Alquier, 2021; Guedj, 2019; Hellström et al., 2023) have obtained a great deal of attention in recent years, since they are the only framework within which non-vacuous bounds for (randomised or compressed variants of) neural networks have been proved. Unlike the bounds we considered above, they generally hold for *randomised* predictors or weights. Specifically, they tend to bound the expectation of the risk with respect to a w sampled from some "posterior" distribution $Q \in \mathscr{P}(\mathcal{W})$. The bounds also usually involve a "complexity measure" based on the Kullback-Liebler (KL) divergence, $\mathrm{KL}(Q,P) = \mathbb{E}_{W\sim Q}\left[\log \frac{dQ}{dP}(W)\right]$, from some fixed "prior" distribution $P \in \mathscr{P}(\mathcal{W})$. This prior can usually be chosen in any data-independent way, lending much flexibility to the generic PAC-Bayesian bounds, as it is chosen to optimise the tightness for the problem types or classification classes we are considering.

The following PAC-Bayes bound, (derived from Kakade et al. (2008), Corollary 8) is very slightly different from those usually presented, but is more similar to Eq. (2.3) above. It states that for a fixed "prior" $P \in \mathscr{P}(\mathcal{W})$ and some numerical constant $C > 0$,

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left(\forall Q \in \mathscr{P}(\mathcal{W})\,,\ \mathop{\mathbb{E}}_{W\sim Q}\mathcal{L}(W) \le \mathop{\mathbb{E}}_{W\sim Q}\widehat{\mathcal{L}}(W) + C\sqrt{\frac{\mathrm{KL}(Q,P)+\log\frac{1}{\delta}}{m}}\right) \ge 1-\delta. \quad (2.5)$$

For majority votes. One of the oldest uses of PAC-Bayesian methods (in Seeger, 2002, applied to Gaussian process classification) is in obtaining bounds for the risk of the majority vote of Q, *i.e.* $\mathcal{L}(\mathrm{MV}_Q)$. A wide variety of "oracle" bounds (Lacasse et al., 2010; Masegosa

et al., 2020; Wu et al., 2021)[5] have been proved that relate the risk of the majority vote to the average risk of W sampled from Q, or similar quantities which can be bounded by PAC-Bayesian methods.

The simplest and often tightest of these bounds is the first-order (FO) oracle bound

$$\mathcal{L}(\mathrm{MV}_Q) \leq 2 \mathop{\mathbb{E}}_{W\sim Q} \mathcal{L}(W),$$

called the "folk" theorem by Langford and Shawe-Taylor (2003). This can be trivially combined with Eq. (2.5) to obtain bounds for the majority vote risk in terms of the weighted empirical risks of its components.

Majority votes over a finite set are the topic of Chapter 7, while in Chapter 6 we use a neural network that can be expressed as a specially-constructed majority vote to prove a generalisation result using the FO bound; technically some of the predictors in Chapter 4 or Chapter 5 can also be seen as majority votes, though we do not use this fact directly.

For neural networks. The resurgence of interest in PAC-Bayesian methods has largely come due to their successful application in obtaining non-vacuous generalisation bounds for neural networks. Specifically, they have been used to obtain such bounds for neural networks with Normally-distributed weights (Dziugaite and Roy, 2017, 2018; Langford and Caruana, 2002), and compressed networks (Zhou et al., 2019). Later results looked at optimising PAC-Bayesian bounds for neural networks (Pérez-Ortiz et al., 2021b,c) as is our focus in Chapter 4, or improving tightness by choosing the prior more optimally (Dziugaite et al., 2020; Pérez-Ortiz et al., 2021a). Neyshabur et al. (2018) used a PAC-Bayes bound for these Gaussian weight networks as a stepping stone to prove a *margin* bound for a deterministic network, an approach we will also explore in Chapter 5.

[5] We discuss these extensively in Chapter 7.

Chapter 3

From Test Bounds to PAC-Bayes

This chapter provides an accessible introduction to the technical aspect and proofs of PAC-Bayesian generalisation bounds, as well as giving intuition for why the bounds look the way they do. A secondary aim is to introduce sub-Gaussian random variables and their concentration, which are used in both Chapters 5 and 7 for the *de-randomsiation* of PAC-Bayesian bounds via *margins*.

3.1 Introduction

This chapter can be read both as a new, stand-alone introduction to PAC-Bayes and concentration; or the concentration part can be read as technical background for the margin-based work in this thesis. We draw on the introductions of Langford (2005); van Erven (2014) to PAC-Bayes, as well as on Boucheron et al. (2013) for concentration of measure. Although the majority of this thesis builds off on a single PAC-Bayesian bound (or variations thereof), Chapter 8 will build off of these ideas to prove a new PAC-Bayesian bound.

The most basic concentration inequality, which effectively forms the basis for all others we discuss here, is Markov's inequality. Let $Z \geq 0$ be a random variable; then for any $\epsilon > 0$

$$\mathbb{P}(Z \geq \epsilon) \leq \epsilon^{-1}\,\mathbb{E}[Z].$$

This can also be stated in a PAC-like form as

$$\mathbb{P}\big(Z \leq \delta^{-1}\,\mathbb{E}[Z]\big) \geq 1-\delta.$$

This bound is extremely useful and general, but can be quite loose. We note also that taking empirical averages of i.i.d. does not improve the bound, so the central limit theorem suggests we could do better.

Chebyshev's inequality can be obtained by applying Markov's inequality to the non-negative variable $(Z - \mathbb{E}\,Z)^2$ (where Z itself no longer needs to be non-negative):

$$\mathbb{P}\left(|Z - \mathbb{E}\,Z| \leq \sqrt{\delta^{-1}\mathbb{V}[Z]}\right) \geq 1-\delta.$$

It bounds the deviation of a variable from its *mean*, rather than a general upper tail as in Markov's inequality.

This bound has the advantage that it effectively becomes tighter with increasing sample size m, since the variance of a sum of i.i.d. variables is simply the sum of variances. Application to $\widehat{\mathcal{L}}$ gives the bound

$$\mathbb{P}\left(|\mathcal{L}(w) - \widehat{\mathcal{L}}(w)| \leq \sqrt{\frac{\mathbb{V}[\ell(Z_1, w)]}{m\delta}}\right) \geq 1 - \delta. \tag{3.1}$$

Thus provided the variance of the loss is bounded (a condition necessary for almost any practical learning situation), we obtain $\mathcal{O}(1/\sqrt{m})$ convergence of the empirical risk to its population average. Suppose our losses are bounded in $[0,1]$; then $\mathbb{V}[\ell(Z_1, w)] \leq \frac{1}{4}$ by Popoviciu's inequality. This gives our first "PAC bound".

So are we finished with the topic of statistical learning theory? No, for a few reasons, which we will address in the following order:

- The dependence on δ is sub-optimal; under conditions such as a bounded loss function the dependence $\mathcal{O}(1/\delta)$ can be greatly improved to $\mathcal{O}(\log(1/\delta))$. Indeed, when "high-probability" bounds are discussed, such a logarithmic dependence on δ is often assumed. In order to prove such bounds, we introduce the Cramer-Chernoff method, which is at the center of more general PAC-Bayesian proofs, and sub-Gaussian random variables, which are utilised heavily in some of the following chapters.

- The dependence on m can sometimes be improved. Although the central limit theorem would appear to suggest a rate of $\mathcal{O}(1/\sqrt{m})$ cannot be improved, in certain cases considerably tighter bounds on $\mathcal{L}(w)$ can still be obtained; for example when $\widehat{\mathcal{L}}(w) \approx 0$. As a short example, consider when $\ell \in \{0,1\}$, so that $\mathbb{V}[\ell(Z_1, w)] = \mathcal{L}(w)(1 - \mathcal{L}(w)) \leq \mathcal{L}(w)$, and $\widehat{\mathcal{L}}(w) = 0$ is observed; solving Eq. (3.1) for $\mathcal{L}(w)$ gives $\mathcal{L}(w) \leq \frac{1}{m\delta} \in \mathcal{O}(1/m)$.

- The above assumes that w is chosen *independently* of the sample. This corresponds to the sample being from a test set. However, we may want to make predictions about $\mathcal{L}(w)$ using the training sample. Such predictions can be useful in understanding why and when certain learning algorithms work, as well as in obtaining guarantees. In order to do so we introduce PAC-Bayesian bounds. Indeed, this last point will form the majority of this thesis.

3.2 Test Bounds that are Logarithmic in δ

The Cramer-Chernoff method can lead to considerably tighter bounds.

Theorem 3.1. Chernoff Bound, PAC Form*: for any* $\delta \in (0,1)$ *and random variable* Z*,*

$$\mathbb{P}\left(Z \leq \inf_{\lambda > 0} \frac{1}{\lambda} \log \frac{M_Z(\lambda)}{\delta}\right) \geq 1 - \delta$$

where $M_Z(\lambda) := \mathbb{E}\, e^{\lambda Z}$ is the moment-generating function of Z.

Proof. For any $\lambda > 0, \epsilon > 0$,

$$\mathbb{P}(Z \geq \epsilon) = \mathbb{P}\left(e^{\lambda Z} \geq e^{\epsilon\lambda}\right) \leq e^{-\epsilon\lambda}\,\mathbb{E}\, e^{\lambda Z} =: \delta$$

with the inequality following from Markov's inequality since $e^{\lambda Z}$ is non-negative. We invert for ϵ and note that λ can be optimised independently of the realisation of Z to obtain the final result. □

The above is a slightly different form from the usual statements, and is chosen because of its similarity to PAC-Bayesian inequalities later discussed. From this, many bounds of a similar form to those used in PAC-Bayes can be obtained. The problem of obtaining concentration has been reduced to that of obtaining tight upper bounds on the moment-generating-function.

Sub-Gaussian Random Variables. The class of sub-Gaussian random variables are those for which Gaussian-like concentration be obtained. They will be of great interest in proving the *margin* bounds we later discuss, as well as in developing an understanding of PAC-Bayesian proof techniques.

Definition (Sub-Gaussian Variable)**.** *We say random variable Z is σ^2-**sub-Gaussian** (sub-Gaussian with parameter σ^2), if $\forall t \in \mathbb{R}$*

$$\log \mathbb{E}\left[e^{t(Z - \mathbb{E}\, Z)}\right] \leq \frac{1}{2}\sigma^2 t^2. \tag{3.2}$$

Now what variables are sub-Gaussian? Firstly, we note that the inequality in Eq. (3.2) is obtained by a Gaussian variable with variance σ^2. Secondly, by Hoeffding's lemma, a random variable bounded in $[a, b]$ is sub-Gaussian with parameter $\sigma^2 \leq \frac{1}{4}(b-a)^2$. Finally, we note that sums of independent sub-Gaussian variables are sub-Gaussian with parameter $\sigma^2 = \sigma_1^2 + \sigma_2^2$ (which folllows since by independence $\mathbb{E}\, e^{Z_1 + Z_2} = \mathbb{E}\, e^{Z_1} \times \mathbb{E}\, e^{Z_2}$).

If a variable is sub-Gaussian, then

$$\inf_{\lambda > 0} \frac{1}{\lambda} \log \frac{M_{Z - \mathbb{E}\, Z}(\lambda)}{\delta} \leq \inf_{\lambda > 0} \frac{1}{2}\sigma^2\lambda + \frac{\log(1/\delta)}{\lambda} = \sqrt{2\sigma^2 \log(1/\delta)}. \tag{3.3}$$

Further, if we take the empirical average of m i.i.d variables Z_i, by Theorem 3.1

$$\mathbb{P}\left(\frac{1}{m}\sum_{i=1}^{m} Z_i - \mathbb{E}[Z] \leq \sqrt{\frac{2\sigma^2 \log(1/\delta)}{m}}\right) \geq 1 - \delta.$$

This gives our first improvement of Eq. (3.1), by applying the above to the (negative) empirical risk we obtain Eq. (2.1) and the theorem below.

Theorem 3.2. *For fixed w and $\ell \in [0, 1]$*

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m}\left(\mathcal{L}(w) \leq \widehat{\mathcal{L}}(w) + \sqrt{\frac{\log(1/\delta)}{2m}}\right) \geq 1 - \delta$$

3.3 Faster-Rate Test Bounds

Now we have improved the rate in δ, what about the rate in m? We already showed how it can be improved in Eq. (3.1), but what about with logarithmic δ? As a motivating example, consider again loss values in $\{0,1\}$, so that $m\widehat{\mathcal{L}}$ is Binomial distributed with parameters $\mathcal{L}(w)$ and m. Suppose we observe $\widehat{\mathcal{L}}=0$: the probability of obtaining such a realisation of losses is $\mathbb{P}_{Z\sim\text{Binomial}(\mathcal{L}(w),m)}(Z=0)=(1-\mathcal{L}(w))^m\leq e^{-m\mathcal{L}(w)}$. Therefore with high probability, observing $\widehat{\mathcal{L}}=0$ implies that $\mathcal{L}(w)\leq\frac{\log(1/\delta)}{m}$, an $\mathcal{O}(1/m)$ rate.

Some of the sharpest results for bounded losses will be given through the following method. First we introduce the following function, related to the moment-generating-function of a Bernoulli random variable,

$$\Phi_C(p)=-\frac{1}{C}\log(1-p+pe^{-C}).$$

Now note that by convexity, for any $t\in\mathbb{R}$ and $Z\in[0,1]$, $\exp(tZ)\leq(1-Z)e^0+Ze^t$. Therefore

$$M_{\Phi_C(\mathbb{E}Z)-Z}(C)=\mathbb{E}\exp(C(\Phi_C(\mathbb{E}Z)-Z))=\frac{\mathbb{E}e^{-CZ}}{1-p+pe^{-C}}\leq\frac{\mathbb{E}\left[1-Z+Ze^{-C}\right]}{1-p+pe^{-C}}=1. \quad (3.4)$$

By combination with Theorem 3.1 (noting that the result holds for any $C>0$), for i.i.d. $Z_i\in[0,1]$,

$$\mathbb{P}\left(\forall C>0:\Phi_C(\mathbb{E}Z)\leq\frac{1}{m}\sum_{i=1}^m Z_i+\frac{1}{Cm}\log(1/\delta)\right)\geq 1-\delta.$$

Furthermore, Φ_C is invertible. Through this we get the following result, which is strongly reminiscent of Catoni's (Catoni, 2007) PAC-Bayes theorem, with the exception of the infimum over C.

Theorem 3.3. Catoni-like Bound*: For fixed w and $\ell\in[0,1]$,*

$$\mathbb{P}\left(\mathcal{L}(w)\leq\inf_{C>0}\Phi_C^{-1}\left(\widehat{\mathcal{L}}(w)+\frac{1}{mC}\log\frac{1}{\delta}\right)\right)\geq 1-\delta$$

where

$$\Phi_C^{-1}(x):=\frac{1-e^{-Cx}}{1-e^{-C}}.$$

Before discussing the behaviour of the above as m and $\widehat{\mathcal{L}}(w)$ vary, we introduce an alternative formulation of the above.

3.3.1 Bernoulli Small-kl Based Bounds

This result can be stated in an equivalent way using the Bernoulli KL divergence function (or *small-kl*), which we discuss next, giving more intuition for the behaviour of this bound. The variant takes the same form as our most commonly-used PAC-Bayes bound, just as Theorem 3.3 is similar to the Catoni bound. Although they are equivalent here, once we proceed to PAC-Bayesian analysis there are subtle differences between both results.

For parameters $p \in [0,1], q \in [0,1]$, we define the small-kl as

$$\mathrm{kl}(q,p) := q \log \frac{q}{p} + (1-q) \log \frac{1-q}{1-p}.$$

This function is convex in both arguments with a minimum at $p = q$.

Small-kl PAC-Bayes bounds will take the approximate form $\mathrm{kl}\left(\widehat{\mathcal{L}}, \mathcal{L}\right) \leq B$ for some upper bound $B > 0$. We wish to invert these to get a bound on $\mathcal{L}$, so we define the *inverse* small-kl, which is a generalised inverse, as

$$\mathrm{kl}^{-1}(q,B) := \sup \{p \in (0,1) : \mathrm{kl}(q,p) \leq B\}.$$

The following useful lemma shows Theorem 3.3 can also be written using kl^{-1}.

Theorem 3.4 (Small-kl inversion and Inverse CGF)**.** *For $p \in (0,1)$, $q \in [0,1)$, $B > 0$*

$$\mathrm{kl}^{-1}(q,B) = \inf_{C>0} \Phi_C^{-1}(q + B/C).$$

Proof. This useful result is closely related to Germain et al. (2009, Proposition 2.1), who show that for any $0 \leq q \leq p < 1$

$$\sup_{C>0} [C\Phi_C(p) - Cq] = \mathrm{kl}(q,p).$$

Note that this can be seen as an instance of the Donsker-Varadhan equality (Donsker and Varadhan, 1975). We slightly generalise the above by noting that when $p < q$,

$$\sup_{C>0} [C\Phi_C(p) - Cq] = 0.$$

Therefore for any $q \in [0,1]$, $p \in (0,1)$,

$$\sup_{C>0} [C\Phi_C(p) - Cq] = \mathrm{kl}_+(q,p),$$

where we define the *upper tail* kl (as also used in *e.g.* Biggs, 2022; Langford, 2002; Wu and Seldin, 2022) as

$$\mathrm{kl}_+(q,p) := \begin{cases} \mathrm{kl}(q,p) & \text{for } q \leq p, \\ 0 & \text{else.} \end{cases}$$

Note that

$$\mathrm{kl}^{-1}(q,B) = \sup\{p \in [0,1] : \mathrm{kl}(q,p) \leq B\} = \sup\{p \in [0,1] : \mathrm{kl}_+(q,p) \leq B\},$$

as the supremum will always be in the right hand part of the function where $p \geq q$. Combining these results we find that

$$p \leq \mathrm{kl}^{-1}(q,B) \iff \mathrm{kl}_+(q,p) \leq B \iff \sup_{C>0} [C\Phi_C(p) - Cq] \leq B$$
$$\iff p \leq \inf_{C>0} \Phi_C^{-1}(q + B/C).$$

from which the conclusion follows. □

Theorem 3.5. Relative-Entropy Test Set Bound*: for fixed w,*

$$\mathbb{P}\left(\mathcal{L}(w) \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}(w), \frac{1}{m}\log\frac{1}{\delta}\right)\right) \geq 1-\delta.$$

The technique we used to prove this bound is somewhat unorthodox (the usual method proceeding via Chernoff's relative entropy bound) but highlights the link with Theorem 3.3.

We discuss some relaxations of this bound in order to serve as intuition. By Pinsker's inequality, $\mathrm{kl}^{-1}(q,B) \leq q+\sqrt{B/2}$, which gives a $\mathcal{O}(1/\sqrt{m})$ rate. From this we directly recover Theorem 3.2, so this formulation is strictly tighter. Further, when $q=0$, $\mathrm{kl}^{-1}(0,B) \leq 1-e^{-B} \leq B$, since $\mathrm{kl}(0,p) = -\log(1-p)$. This recovers the rate $\mathcal{O}(1/m)$ as seen with the Binomial and $\mathcal{L}(w)=0$ seen above.

What happens between these two regimes (often known in learning theory as the *agnostic* and *realisable* cases)? An alternative relaxation is given by a relative entropy relaxation (proved in *e.g.* McAllester, 2003 and discussed further in Chapter 8) as $\mathrm{kl}^{-1}(q,B) \leq q+\sqrt{2Bq}+2B$. This second variation shows the upper bound on $\mathcal{L}(w)-\widehat{\mathcal{L}}(w)$ interpolates between $\mathcal{O}(1/\sqrt{m})$ when $\widehat{\mathcal{L}}(w)$ is large, and $\mathcal{O}(1/m)$ when it approaches zero. The inequality $\Phi_C^{-1}(x) \leq \frac{1}{1-e^{-C}}x$ also implies that for any $c>1$, $\mathcal{L} \leq c\widehat{\mathcal{L}} + \mathcal{O}(1/m)$, where the constants in the asymptotic notation depend on c.

Finally, we also note that the inverse kl function can be calculated straightforwardly by a couple of iterations of Newton's method, as can its gradients with respect to its arguments (Clerico et al., 2022).

3.3.2 Alternative Fast-Rate Bounds

We mention also a slightly different approach using variances. We already discussed how these can lead to faster rates without being logarithmic in δ. Logarithmic bounds in δ with fast rates using variances can be obtained using so-called "Bernstein" inequalities. This has been one of the main approaches in non PAC-Bayesian learning theory (and more recently, in PAC-Bayes also) to obtaining such fast rates. We discuss this much further in Chapter 8, where we are concerned with proving new fundamental PAC-Bayes bounds; in most other places we use a bound similar to Theorem 3.5, which we discuss next.

3.4 PAC-Bayes Bounds

In order to move from the above "test set" bounds to PAC-Bayesian bounds, we introduce the KL divergence and Donsker-Varadhan inequality.

Definition (Kullback-Liebler Divergence)**.** *For $P,Q \in \mathscr{P}(\mathcal{H})$, the KL divergence*

$$\mathrm{KL}(Q,P) := \int_{\mathrm{supp}(P)} \log\left(\frac{dQ}{dP}\right) dQ$$

if $Q \ll P$ (Q is absolutely continuous w.r.t. P) and $+\infty$ otherwise.

The KL is non-negative by Gibbs' inequality (essentially an application of Jensen's inequality to the convex function $x \log x$) and jointly convex in its arguments. If P is taken uniform over a finite set, $\mathrm{KL}(Q, \mathrm{Unif}(\mathcal{W})) = \log|\mathcal{W}| - \mathbb{H}[Q] \leq \log|\mathcal{W}|$ where $\mathbb{H}[Q]$ is the entropy of Q (this sometimes reduces later PAC-Bayesian bounds to covering or finite hypothesis space bounds). A less elementary property is the Donsker-Varadhan dual formulation of the KL divergence, based on a following generalisation of the moment generating function.

Theorem 3.6 (Csiszár (1975); Donsker and Varadhan (1975))**.** *Let $P \in \mathscr{P}(\Omega)$ be a probability measure on Ω, and $\mathcal{F}$ be the set of measurable bounded real-valued functions on Ω. The function $\mathrm{KL}(\cdot, P)$ has a convex dual, so that for any $Q \in \mathscr{P}(\Omega)$,*

$$\mathrm{KL}(Q, P) = \sup_{\phi \in \mathcal{F}} \left[\int \phi \, dQ - \log \int \exp \circ \phi \, dP \right],$$

and for any $\phi \in \mathcal{F}$,

$$\log \int \exp \circ \phi \, dP = \sup_{Q \in \mathscr{P}(\Omega)} \left[\int \phi \, dQ - \mathrm{KL}(Q, P) \right].$$

3.4.1 Generic PAC-Bayesian Result

This result can be used almost immediately to prove a result similar to the Cramer-Chernoff bound Theorem 3.1.

Theorem 3.7 (Generic PAC-Bayes)**.** *For any fixed $f : \mathcal{Z}^m \times \mathcal{W}$ and $P \in \mathscr{P}(\mathcal{W})$,*

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left(\forall Q \in \mathscr{P}(\mathcal{W}) : \mathop{\mathbb{E}}_{W \sim Q}[f(S, W)] \leq \mathrm{KL}(Q, P) + \log \frac{M_f}{\delta} \right) \geq 1 - \delta,$$

where

$$M_f := \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^m} \mathop{\mathbb{E}}_{h \sim P} \left[e^{f(S', W)} \right].$$

Proof. The proof of this statement is very similar to the Cramer-Chernoff method, but uses Theorem 3.6. By Theorem 3.6, exponentiation, Markov's inequality, and the definition of M_f we find that

$$\begin{aligned}
&\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left(\sup_{Q \in \mathscr{P}(\mathcal{W})} \mathop{\mathbb{E}}_{W \sim Q}[f(S, W)] - \mathrm{KL}(Q, P) > \log \frac{M_f}{\delta} \right) \\
&= \mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left(\log \mathop{\mathbb{E}}_{W \sim P} \left[e^{f(S', W)} \right] > \log \frac{M_f}{\delta} \right) \\
&= \mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left(\mathop{\mathbb{E}}_{W \sim P} \left[e^{f(S', W)} \right] > \frac{M_f}{\delta} \right) \\
&\leq \mathop{\mathbb{E}}_{S \sim \mathcal{D}^m} \mathop{\mathbb{E}}_{W \sim P} \left[e^{f(S', W)} \right] \times \frac{\delta}{M_f} \\
&= \delta.
\end{aligned}$$

The result follows by taking the complement of both sides. □

Similarly to in Theorem 3.1, the problem is reduced to that of bounding M_f. We will study a few simple cases, firstly relating back to sub-Gaussian assumptions, then to Theorem 3.3 and Theorem 3.5.

3.4.2 Sub-Gaussian PAC-Bayes

An easy case assumes that $\mathcal{L}(w) - \widehat{\mathcal{L}}(w)$ is sub-Gaussian under $\mathcal{D}^m$ for any fixed w, and that P is independent of S. This arises *e.g.* for bounded losses. The result obtained is like an intermediate step in Theorem 3.2, and involves an additional scaling factor λ which cannot be optimised over, unlike in Theorem 3.2.

Theorem 3.8 (Sub-Gaussian PAC-Bayes, *e.g.* Alquier et al., 2016[1])**.** *Let*

$$\log \mathop{\mathbb{E}}_{Z\sim\mathcal{D}} \exp\left(t\left(\ell(Z,w) - \mathop{\mathbb{E}}_{Z'\sim\mathcal{D}} \ell(Z',w)\right)\right) \leq \frac{1}{2}\sigma^2 t^2$$

for any $t \in \mathbb{R}$, $w \in \mathcal{W}$. Then for any fixed $P \in \mathscr{P}(\mathcal{W}), \delta \in (0,1)$ and $\lambda > 0$,

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left(\forall Q \in \mathscr{P}(\mathcal{W}) : \mathop{\mathbb{E}}_{W\sim Q}[\mathcal{L}(W) - \widehat{\mathcal{L}}(W)] \leq \frac{1}{\lambda}\left(\mathrm{KL}\,(Q,P) + \log\frac{1}{\delta} + \frac{\sigma^2\lambda^2}{2m}\right)\right) \geq 1-\delta.$$

Proof. This result comes from Theorem 3.7 by substitution of $f(S,W) = \lambda(\mathcal{L}(W) - \widehat{\mathcal{L}}(W))$. We find that

$$M_f = \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m} \mathop{\mathbb{E}}_{h\sim P}\left[e^{\lambda(\mathcal{L}(W)-\widehat{\mathcal{L}}(W))}\right] = \mathop{\mathbb{E}}_{h\sim P} \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m}\left[e^{\lambda(\mathcal{L}(W)-\widehat{\mathcal{L}}(W))}\right]$$

since P is independent of S. This factorises by the independence of the Z_i and the individual terms are bounded by the sub-Gaussian assumption. □

Note that in particular, we can set $\lambda = \sqrt{m}$ to get the high probability result over all Q that

$$\mathop{\mathbb{E}}_{W\sim Q}[\mathcal{L}(W) - \widehat{\mathcal{L}}(W)] \leq \frac{\mathrm{KL}\,(Q,P) + \log\frac{1}{\delta} + \frac{1}{2}\sigma^2}{\sqrt{m}},$$

with $\mathcal{O}(1/\sqrt{m})$ convergence. However, unlike in Theorem 3.2, we cannot choose λ optimally to place a square root over all the terms, because $\mathrm{KL}\,(Q,P)$ is different for different values of Q and therefore not known beforehand. This could be compared to the result in Eq. (3.3) before taking the infimum over λ. Note also the crucial step where the order of expectations under P and $\mathcal{D}$ is exchanged, through the independence of the prior from the data.

3.4.3 Catoni's PAC-Bayes Bound

We next discuss a PAC-Bayes bound based on Theorem 3.3, where a similar issue of not being able to optimise over C arises.

Theorem 3.9. Catoni's PAC-Bayes Bound*: Given $\ell \in [0,1]$, $P \in \mathscr{P}(\mathcal{W})$ and $C > 0$,*

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left(\mathop{\mathbb{E}}_{W\sim Q}\mathcal{L}(W) \leq \Phi_C^{-1}\left(\mathop{\mathbb{E}}_{W\sim Q}\widehat{\mathcal{L}}(W) + \frac{\mathrm{KL}\,(Q,P) + \log\frac{1}{\delta}}{mC}\right)\right) \geq 1-\delta.$$

[1] Note also Alquier and Biau (2013b); Guedj and Alquier (2013b) which consider PAC-Bayesian regression under sub-Gaussian noise, rather than sub-Gaussian losses.

Proof. For fixed $C > 0$, consider $f(S,W) = C(\Phi_C(\mathcal{L}(w)) - \widehat{\mathcal{L}}(W))$ in Theorem 3.7. Again using the independence of P from S, and the Z_i from each other

$$M_f = \mathop{\mathbb{E}}_{h\sim P} \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m} \left[e^{C(\Phi_C(\mathcal{L}(w)) - \widehat{\mathcal{L}}(W))} \right] = \prod_i e^{C(\Phi_C(\mathcal{L}(w)) - \ell(Z_i,W))} \leq 1,$$

with the last step following from Eq. (3.4). Therefore

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m} \left(\forall Q \in \mathscr{P}(\mathcal{W}) : \mathop{\mathbb{E}}_{W\sim Q} [\Phi_C(\mathcal{L}(W))] \leq \mathop{\mathbb{E}}_{W\sim Q} \widehat{\mathcal{L}}(W) + \frac{\mathrm{KL}\,(Q,P) + \log\frac{1}{\delta}}{Cm} \right) \geq 1 - \delta.$$

Now Φ_C is convex so by Jensen's inequalty we can replace $\mathbb{E}_Q\, \Phi_C(\mathcal{L})$ by $\Phi_C(\mathbb{E}_Q\, \mathcal{L})$ in the above. Inversion of Φ_C then gives the result. □

The above bound is extremely tight when C is carefully chosen; in fact, when C is optimal, the bound is the tightest possible for $\{0,1\}$-valued losses that can be proved using Theorem 3.7 (Foong et al., 2021). However, as already noted, C cannot be chosen using the data. One solution to this is to define a covering of C, *i.e.* a set $\{C_i : i \in \mathcal{N}\}$, and combine the bounds holding for each C_i in the cover using a union bound. This is the approach used in the following bound (whose full proof we neglect, but which begins by constructing a cover with $C_i = \alpha^i$ for $\alpha > 1$ and $i \in \mathcal{N}$). In the bound, $\alpha > 1$ is a parameter controlling the spacing of the cover, which we set near to 1 (*e.g.* $1 + 10^{-5}$) when empirically evaluating.

Theorem 3.10 (Catoni, 2007, Theorem 1.2.6)**.** *Given $P \in \mathscr{P}(\mathcal{W})$, $\alpha > 1$ and $\delta \in (0,1)$*

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m} \left(\forall Q \in \mathscr{P}(\mathcal{W}) : \mathop{\mathbb{E}}_{W\sim Q} [(\mathcal{L}(W))] \leq \inf_{C > 1/m} \Phi_C^{-1} \left(\mathop{\mathbb{E}}_{W\sim Q} \widehat{\mathcal{L}}(W) + \frac{K\alpha}{Cm} \right) \right) \geq 1 - \delta.$$

where

$$K := \mathrm{KL}\,(Q,P) + \log\frac{1}{\delta} + 2\log\left(\frac{\log(\alpha^2 Cm)}{\log\alpha} \right).$$

This bound has the advantage that it gives a linear objective function in Q of

$$\mathop{\mathbb{E}}_{W\sim Q} \widehat{\mathcal{L}}(W) + \frac{\mathrm{KL}\,(Q,P)}{Cm},$$

where C is either fixed, or optimised in Theorem 3.10 alternatively with this linear objective.

Can we however optimise to get a bound in the form of Theorem 3.5? It turns out that we can get such a bound through a different method, which is of similar overall tightness to the above.

3.4.4 Maurer's PAC-Bayes Bound

An alternative, more direct approach to obtain a result like Theorem 3.5 relies on the following lemma.

Theorem 3.11 (Maurer, 2004, Eq. 1)**.** *Let $Z_i \in [0,1], i \in [m]$ be i.i.d. random variables with mean μ; then for any $m \geq 1$*[2]*,*

$$\mathbb{E} \exp\left(\mathrm{kl}\left(\frac{1}{m} \sum_{i=1}^{m} Z_i, \mu \right) \right) \leq 2\sqrt{m}.$$

From this we give the following bound, a slightly weaker version of which was originally proved in Seeger et al. (2001).

Theorem 3.12. Maurer's PAC-Bayes Bound *(Maurer, 2004; Seeger et al., 2001): Given $P \in \mathscr{P}(\mathcal{W})$, and $\ell \in [0,1]$,*

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left(\forall Q \in \mathscr{P}(\mathcal{W}) : \mathrm{kl}\left(\mathop{\mathbb{E}}_{W \sim Q} \widehat{\mathcal{L}}(W), \mathop{\mathbb{E}}_{W \sim Q} \mathcal{L}(W) \right) \leq \frac{\mathrm{KL}\left(Q, P\right) + \log \frac{2\sqrt{m}}{\delta}}{m} \right) \geq 1 - \delta.$$

Proof. The proof is very similar to that of Theorem 3.9. With $f(S,W) = m\,\mathrm{kl}\Big(\widehat{\mathcal{L}}(W), \mathcal{L}(W)\Big)$, Theorem 3.11 gives $M_f \leq 2\sqrt{m}$ in Theorem 3.7 (since we can exchange the order of expectations over P and S by the independence of P from S). Since kl is jointly convex in its arguments, $\mathrm{kl}\Big(\mathbb{E}_{W\sim Q}\,\widehat{\mathcal{L}}(W), \mathbb{E}_{W\sim Q}\,\mathcal{L}(W)\Big) \leq \mathbb{E}_{W\sim Q}\,\mathrm{kl}\Big(\widehat{\mathcal{L}}(W), \mathcal{L}(W)\Big) = \mathbb{E}_{W\sim Q}\, f(S,W)/m$, so the result follows from Theorem 3.7. □

An immediate corollary of this bound is a result similar to the test bound Theorem 3.5, with an extra $\log(2\sqrt{m})$ term and the change-of-measure KL-divergence term, which generally makes the largest contribution to the bound.

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m} \left(\forall Q \in \mathscr{P}(\mathcal{W}) : \mathop{\mathbb{E}}_{W \sim Q} \mathcal{L}(W) \leq \mathrm{kl}^{-1}\left(\mathop{\mathbb{E}}_{W \sim Q} \widehat{\mathcal{L}}(W), \frac{\mathrm{KL}\left(Q, P\right) + \log \frac{2\sqrt{m}}{\delta}}{m} \right) \right) \geq 1 - \delta.$$

[2]Technically Maurer (2004) proved the above for $m \geq 8$, while the extension to $1 \leq m \leq 7$ was performed via a numerical verification in Germain et al. (2015), Lemma 15.

Chapter 4

Differentiable PAC-Bayes Objectives with Partially Aggregated Neural Networks

We make two related contributions motivated by the challenge of training stochastic neural networks, particularly in a PAC-Bayesian setting: (1) we show how averaging over an ensemble of stochastic neural networks enables a new class of partially-aggregated estimators, proving that these lead to unbiased lower-variance output and gradient estimators; (2) we reformulate a PAC-Bayesian bound for signed-*output* networks to derive in combination with the above a directly optimisable, differentiable objective and a generalisation guarantee, without using a surrogate loss or loosening the bound. We show empirically that this leads to competitive generalisation guarantees and compares favourably to other methods for training such networks. Finally, we note that the above leads to a simpler PAC-Bayesian training scheme for sign-*activation* networks than existing work by Letarte et al. (2019).

4.1 Introduction

The use of stochastic neural networks has become widespread in the PAC-Bayesian and Bayesian deep learning (Blundell et al., 2015) literature as a way to quantify predictive uncertainty and obtain generalisation bounds. PAC-Bayesian theorems generally bound the expected loss of *randomised* estimators, so it has proven easier to obtain non-vacuous numerical guarantees on generalisation in such networks.

However, when training these networks in the PAC-Bayesian setting, the objective used is generally somewhat divorced from the bound on misclassification loss itself, because the

non-differentiability of bounds and stochasticity of predictors lead to difficulties with direct optimisation. For example, Langford and Caruana (2002), Zhou et al. (2019), and Dziugaite and Roy (2017) all initially train non-stochastic networks under a surrogate loss. These are used as the mode of a distribution over predictors with variance chosen, respectively, through a computationally-expensive sensitivity analysis, as a proportion of weight norms, or by optimising an objective with both a surrogate loss function and a different dependence on the Kullback–Leibler (KL) divergence from their bound.

In exploring methods to circumvent this gap, we note that PAC-Bayesian bounds can often be straightforwardly adapted to aggregates or averages of estimators, leading directly to analytic and differentiable objective functions (for example, Germain et al., 2009). Unfortunately, averages over deep stochastic networks are usually intractable or, if possible, very costly (as found by Letarte et al., 2019).

Motivated by these observations, our main contribution is to obtain a compromise by defining new and general "*partially-aggregated*" Monte Carlo estimators for the average output and gradients of deep stochastic networks, with the direct optimisation of PAC-Bayesian bounds in mind. Although our main focus here is on the use of this estimator in a PAC-Bayesian application, we emphasise that the technique applies generally to stochastic networks and thus has links to other variance-reduction techniques for training them, such as the path-wise estimator used in the context of neural networks by (Kingma and Welling, 2013) amongst many others or Flipout (Wen et al., 2018); indeed, it can be used in combination with these techniques. We prove that this application leads to lower variances than a Monte Carlo forward pass and lower variance final-layer gradients than REINFORCE (Williams, 1992).

Our first application of this general estimator is to non-differentiable "signed-output" networks for binary classification (with a final output $\in \{-1, +1\}$ and arbitrarily complex other structure, see Section 4.3). This allows direct application of partial-aggregation to neural networks under the misclassification loss. As well as reducing variances as stated above, a small amount of additional structure in combination with partial-aggregation enables us to extend the path-wise estimator to the other layers, which usually requires a fully differentiable network and eases training by reducing the variance of gradient estimates.

We adapt a binary classification bound from Catoni (2007) to these networks, yielding straightforward and directly differentiable objectives when used in combination with aggregation, closing this gap between objectives and bounds. Since most of the existing PAC-Bayes bounds for neural networks have a heavy dependency on the distance from initialisation of the obtained solution, we would intuitively expect these lower variances to lead to faster convergence and tighter bounds (from finding low-error solutions nearer to the initialisation). We indeed observe this experimentally, showing that training PAC-Bayesian objectives

in combination with partial aggregation leads to competitive experimental generalisation guarantees, and improves upon naive Monte Carlo with REINFORCE.

A further contribution is an specialisation of our method to sign-*activation* neural networks, where we define new and computationally efficient gradient estimators. This application leads us to a similar PAC-Bayesian training method to Letarte et al. (2019) for these networks, but is arguably much simpler. That work successfully aggregated networks with all sign activation functions $\in \{+1, -1\}$ and a non-standard tree structure, but incurred an exponential KL divergence penalty and a heavy computational cost. Further, the lower variance of our obtained estimator predictions enables us to use the Gibbs estimator directly (where we draw a single sample function for every new example), leading to a modified bound on the misclassification loss which is a factor of two tighter without a significant performance penalty. We show empirically that our method lets us train deeper such networks and obtains tighter bounds.

A final contribution is an outline for generalising our partial-aggregation estimators to more general settings.

Overview. In Section 4.2 we refresh notation and discuss gradient-based optimisation of the misclassification loss in a PAC-Bayesian setting as well as aggregated predictors. In Section 4.2.1 we introduce our "partial-aggregation" approach to the above problem in a binary classification setting, showing how lower-variance gradient estimates can be obtained, before specialising the above in Section 4.3 to the use of neural networks with binary-only activation functions (as in Letarte et al., 2019), giving a different formulation of the above and a more computationally efficient gradient estimator. We then go in the other direction in Section 4.5 and show how partial-aggregation can be generalised beyond the binary classification setting. We give empirical results in Section 4.6 for the optimisation of PAC-Bayes bounds using partial-aggregation for binary classification, before discussing our results in Section 4.7.

4.2 Background

We begin by refreshing notation and the requisite background. We will primarily consider the problem of binary classification, or learning (a distribution over) parameterised functions of the form $\{f_\theta : \mathcal{X} \to \{+1, -1\} \,|\, \theta \in \Theta \subset \mathbb{R}^{d_\theta}\}$ with $\mathcal{X} \subset \mathbb{R}^{d_0}$. These functions should minimise the out-of-sample misclassification risk, $\mathcal{L}_0(f_\theta) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \mathbf{1}\{f_\theta(\boldsymbol{x}) \neq y\}$, and be chosen based on an i.i.d. sample $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m \overset{\text{iid}}{\sim} \mathcal{D}$ from the data distribution $\mathcal{D}$, using the surrogate of in-sample empirical risk, $\widehat{\mathcal{L}}_0(f)$.

We take a PAC-Bayesian approach, introducing a distribution Q over θ; Q itself will be parameterised by $\phi \in \Phi \subset \mathbb{R}^{d_\phi}$, and we often assume it has a density, q_ϕ. PAC-Bayesian

theorems then provide bounds on the expected generalisation risk of our randomised classifiers, where every prediction is made using a newly sampled function from our posterior, $f_\theta, \theta \sim Q$, *i.e.* they bound $\mathbb{E}_{\theta\sim Q}\,\mathcal{L}_0(f_\theta)$. Note that the term $\widehat{\mathcal{L}}_0(f)$ is typically non-differentiable, since ℓ_0 is non-continuous. This is a problem when trying to optimise bounds, and is usually addressed by introducing a surrogate loss function. However, we note that while $\ell_0(f_\theta,(\boldsymbol{x},y))$ is non-differentiable, $\mathbb{E}_{\theta\sim Q}\,\ell_0(f_\theta,(\boldsymbol{x},y))$ might be with respect to ϕ, the parameter of Q.

To explore this possibility, we consider averaging to obtain Q-aggregated prediction functions,

$$F_Q(\boldsymbol{x}) := \mathop{\mathbb{E}}_{\theta\sim Q} f_\theta(\boldsymbol{x}) \tag{4.1}$$

as often also seen in PAC-Bayes. Because $f_\theta(\boldsymbol{x}) \in \{+1,-1\}$ is a signed-output function, there is an exact equivalence between the *linear* and misclassification losses; *i.e.* $\ell_0(f_\theta,(\boldsymbol{x},y)) = \ell_{\text{lin}}(f_\theta,(\boldsymbol{x},y))$, where the linear loss is defined for any score-output function as $\ell_{\text{lin}}(f,(\boldsymbol{x},y)) = \frac{1}{2}(1-yf(\boldsymbol{x}))$. By linearity we then have

$$\mathop{\mathbb{E}}_{\theta\sim Q} \ell_0(f_\theta,(\boldsymbol{x},y)) = \mathop{\mathbb{E}}_{\theta\sim Q} \ell_{\text{lin}}(f_\theta,(\boldsymbol{x},y)) = \ell_{\text{lin}}(F_Q,(\boldsymbol{x},y)). \tag{4.2}$$

If F_Q is differentiable with respect to ϕ, we have removed the difficult stochastic derivative and replaced it by a simple non-stochastic one when optimising with respect to ϕ.

Combining this with Theorem 3.10 (and recalling that $\alpha > 1$ is a hyperparameter set very close to 1), we then obtain a directly optimisable, high-probability bound on the misclassification loss:

$$\mathop{\mathbb{E}}_{\theta\sim Q} \mathcal{L}_0(f_\theta) \leq \Phi^{-1}_{\lambda/m}\left[\widehat{\mathcal{L}}_{\text{lin}}(F_Q) + \frac{\alpha}{\lambda}\left(\text{KL}\,(Q,P) - \log\delta + 2\log\left(\frac{\log\alpha^2\lambda}{\log\alpha}\right)\right)\right] \tag{4.3}$$

simultaneously holding for any $\lambda > 1$, with $\Phi^{-1}_\gamma(t) := \frac{1-\exp(-\gamma t)}{1-\exp(-\gamma)}$. This slightly opaque bound was used previously by Zhou et al., 2019, and gives almost identical results to the better-known "small-kl" PAC-Bayes bounds originated by Langford and Seeger (2001); Seeger et al. (2001). It is chosen because it leads to objectives that are *linear* in the empirical loss and KL divergence, like

$$\widehat{\mathcal{L}}_{\text{lin}}(F_Q) + \frac{\text{KL}\,(Q,P)}{\lambda}. \tag{4.4}$$

This objective is minimised by a Gibbs distribution and is closely related to the evidence lower bound (ELBO) usually optimised by Bayesian Neural Networks (Blundell et al., 2015). Such a connection has been noted throughout the PAC-Bayesian literature; we refer the reader to Germain et al. (2016) or Knoblauch et al. (2019) for a formalised treatment.

4.2.1 Analytic Q-Aggregates for Signed Linear Functions

Q-aggregate predictors can be stated in closed analytic form for functions like $f_{\boldsymbol{w}}(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}\cdot\boldsymbol{x})$ under a normal distribution, $Q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu},\mathbb{I})$.[1] This case was specifically considered

[1] We define the sign function and "signed" functions have outputs $\in\{+1,-1\}$, as the terminology "binary", used sometimes in the literature, suggests to us too strongly an output $\in\{0,1\}$).

in a PAC-Bayesian context for binary classification by Germain et al. (2009), obtaining a differentiable objective similar to the SVM. They show that (as also in Langford, 2005)

$$\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu},\mathbb{I})} f_{\boldsymbol{w}}(\boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu},\mathbb{I})} \operatorname{sign}(\boldsymbol{w}\cdot\boldsymbol{x}) = \operatorname{erf}\left(\frac{\boldsymbol{\mu}\cdot\boldsymbol{x}}{\sqrt{2}\|\boldsymbol{x}\|}\right), \tag{4.5}$$

using the Gaussian error function $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$, which leads to the closed form

$$\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu},\mathbb{I})} \ell_0(f_{\boldsymbol{w}},(\boldsymbol{x},y)) = \frac{1}{2}\left(1 - y\operatorname{erf}\left(\frac{\boldsymbol{\mu}\cdot\boldsymbol{x}}{\sqrt{2}\|\boldsymbol{x}\|}\right)\right).$$

This is smooth and differentiable with respect to $\boldsymbol{\mu}$, a parameter of Q, and thus can give an easy gradient-optimisable objective function.

An approach which builds on the above was Letarte et al. (2019), which considered extending Q-aggregates to signed-output feed-forward neural networks with sign activation functions, but they did not obtain a closed form for the aggregate.

4.2.2 Monte Carlo and Gradients for More Complex Q-Aggregates

In general, the framework of Q-aggregates can be extended to less tractable cases (for example, with f_θ a randomised or a "Bayesian" neural network, see, *e.g.*, (Blundell et al., 2015)) through a simple and unbiased Monte Carlo approximation using $\{\theta^t\}_{t=1}^T \overset{\text{iid}}{\sim} Q$:

$$F_Q(\boldsymbol{x}) = \mathop{\mathbb{E}}_{\theta\sim Q} f_\theta(\boldsymbol{x}) \approx \frac{1}{T}\sum_{t=1}^{T} f_{\theta^t}(\boldsymbol{x}) =: \widehat{F}_Q(\boldsymbol{x}). \tag{4.6}$$

Suppose Q has a density q_ϕ and we wish to obtain gradients w.r.t. ϕ without a closed form for $F_Q(\boldsymbol{x}) = \mathbb{E}_{\theta\sim q_\phi} f_\theta(\boldsymbol{x})$. There are two main possibilities. One is REINFORCE (Williams, 1992), which makes a Monte Carlo approximation to the right hand side of the identity $\nabla_\phi \mathbb{E}_{\theta\sim q_\phi} f_\theta(\boldsymbol{x}) = \mathbb{E}_{\theta\sim q_\phi}[f_\theta(\boldsymbol{x})\nabla_\phi \log q_\phi(\theta)]$[2].

The other is the path-wise estimator, also known as the "re-parameterisation trick". This additionally requires that $f_\theta(\boldsymbol{x})$ be differentiable with respect to θ, and that the probability distribution of q_ϕ has a "standardisation" function, S_ϕ, which removes the ϕ dependence. This turns a parameterised q_ϕ into a non-parameterised distribution p: for example, $S_{\mu,\sigma}(X) = (X-\mu)/\sigma$ to transform a general normal distribution into a standard normal. If this exists, the right hand side of $\nabla_\phi \mathbb{E}_{\theta\sim q_\phi} f_\theta(\boldsymbol{x}) = \mathbb{E}_{\epsilon\sim p}\nabla_\phi f_{S_\phi^{-1}(\epsilon)}(\boldsymbol{x})$ generally yields lower-variance estimates than REINFORCE (see Mohamed et al., 2019, for a modern survey). In fact, the variance introduced by REINFORCE can make it difficult to train neural networks when the path-wise estimator is not available.

Unfortunately, we have defined f_θ to be non-continuous, so the pathwise estimator is not directly applicable. However, in this work find a compromise between the analytically closed form of (4.5) and the basic monte carlo estimators that enable us to make differentiable

[2]The proof of this identity exchanges derivative with integral and uses the log-derivative trick $\nabla_\phi q_\phi(\theta) = q_\phi(\theta)\nabla_\phi \log q_\phi(\theta)$

certain classes of network and extend the path-wise estimator where otherwise it could not be used. Through this we are able to stably train a new class of network.

4.3 Aggregating Signed-Output Networks

Suppose we have a neural network for binary-classification with a final linear projection and the rest of the network taking an arbitrary stochastic form (for example convolutions, residual layers or a non-feedforward structure). We divide the parameter set $\theta = \text{vec}(\boldsymbol{w}, \theta^g) \in \Theta \subset \mathbb{R}^{d_\theta}$, into $\boldsymbol{w} \in \mathbb{R}^{d_{\text{final}}}$ for the final layer, and $\theta^g \in \Theta^g \subset \mathbb{R}^{d_\theta - d_{\text{final}}}$ the parameter set excluding $\boldsymbol{w}$, for the non-final layers of the network. The predictions of the network are binary in $\{+1, -1\}$ and take the form

$$f_\theta(\boldsymbol{x}) := \text{sign}(\boldsymbol{w} \cdot \boldsymbol{g}(\boldsymbol{x}, \theta^g)) \tag{4.7}$$

with non-final layers included in $\boldsymbol{g} : \mathcal{X} \times \Theta^g \to \mathcal{A}^{d_{\text{final}}} \subseteq \mathbb{R}^{d_{\text{final}}}$. We call such a binary classification network (Eq. (4.7)) a **signed-output** network.

We introduce a distribution over parameters, $Q(\theta)$, which factorises like $Q(\theta) = Q^{\boldsymbol{w}}(\boldsymbol{w})Q^g(\theta^g)$, which is consistent with the literature (where Q factorises over neural network layers). The final layer weights are drawn from a unit variance normal distribution, $Q^{\boldsymbol{w}}(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu}, \mathbb{I})$, while the distribution Q^g could be any tractable form.

Single Hidden Layer. As a clarifying example we will briefly consider the case of a neural network a hidden ReLU layer and Gaussian weights. This sets $f_\theta(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}_2 \cdot \text{ReLU}(W_1 \boldsymbol{x}))$, where $\text{ReLU}(t) = t\,\mathbf{1}\{t > 0\}$ is applied elementwise. $\boldsymbol{g}(\boldsymbol{x}, W_1) = \boldsymbol{A}_1(W_1 \boldsymbol{x})$ for elementwise applied activation $\boldsymbol{A}_1$, and the randomised parameters are $\theta = \text{vec}(\boldsymbol{w}_2, W_1)$, $W_1 \in \mathbb{R}^{d_1 \times d_0}$, $\boldsymbol{w}_2 \in \mathbb{R}^{d_1}$. The distribution $Q(\theta) = Q_2(\boldsymbol{w}_2)Q_1(W_1)$ factorises over the layers as above, and $Q_1(W_1)$ is Gaussian-distributed.

Signed-Outputs for PAC-Bayes. We now move to obtain stochastic binary classifiers with guarantees for the expected misclassification error, $\mathbb{E}_{\theta \sim Q} \mathcal{L}_0(f_\theta)$, which we do by optimising PAC-Bayesian bounds. These PAC-Bayes bounds (as in Theorem 3.10) will usually involve the non-differentiable and non-convex misclassification loss ℓ_0. However, to train a neural network we usually need to replace this by a differentiable surrogate or use REINFORCE, as discussed in Section 4.1. When using our signed output networks, we can instead use the trick in Eq. (4.2) to reduce it to the problem of studying the aggregates F_Q, with the trick also applicable to the gradients, since ℓ_{lin} is linear.

Partial-Aggregation. We recover a similar functional form to that considered in Section 4.2.1 by rewriting the function as $\text{sign}(\boldsymbol{w} \cdot \boldsymbol{a})$ with $\boldsymbol{a} \in \mathcal{A}^{d_{\text{final}}}$ the randomised hidden-layer

activations. This arises by combining Eq. (4.5) with the factorisation of the layers:

$$F_Q(\boldsymbol{x}) = \mathop{\mathbb{E}}_{Q(\theta)}[f_\theta(\boldsymbol{x})] = \mathop{\mathbb{E}}_{Q^g(\theta^g)} \mathop{\mathbb{E}}_{Q^{\boldsymbol{w}}(\boldsymbol{w})} [\operatorname{sign}(\boldsymbol{w} \cdot \boldsymbol{g}(\boldsymbol{x}, \theta^g)] \tag{4.8}$$

$$= \mathop{\mathbb{E}}_{Q^g(\theta^g)} \left[\operatorname{erf}\left(\frac{\boldsymbol{\mu} \cdot \boldsymbol{g}(\boldsymbol{x}, \theta^g)}{\sqrt{2}\|\boldsymbol{g}(\boldsymbol{x}, \theta^g)\|} \right) \right]. \tag{4.9}$$

In most cases the outer expectation cannot be calculated exactly or involves a large summation, so we resort to a Monte Carlo estimate. Specifically, using Eq. (4.6) and Eq. (4.9) with independent samples $\{(\boldsymbol{w}^t, \theta^{g,(t)})\}_{t=1}^T \overset{\text{iid}}{\sim} Q$ and $\boldsymbol{a}^t := \boldsymbol{g}(\boldsymbol{x}, \theta^{g,(t)})$ leads to two different unbiased estimators for the output of $F_Q(\boldsymbol{x})$:

$$\widehat{F}_Q(\boldsymbol{x}) := \frac{1}{T} \sum_{t=1}^{T} \operatorname{sign}(\boldsymbol{w}^t \cdot \boldsymbol{a}^t) \tag{4.10}$$

$$\widehat{F}^*_Q(\boldsymbol{x}) := \frac{1}{T} \sum_{t=1}^{T} \operatorname{erf}\left(\frac{\boldsymbol{\mu} \cdot \boldsymbol{a}^t}{\sqrt{2}\|\boldsymbol{a}^t\|} \right). \tag{4.11}$$

The second we call the "partially-aggregated" estimator. We can use the above further to obtain gradient estimates with respect to $\boldsymbol{\mu}$ using either REINFORCE (for Eq. (4.10)) or direct differentiation of Eq. (4.11) coupled with a Monte Carlo evaluation of $\boldsymbol{a}$. This gives the estimators $\widehat{\nabla_{\boldsymbol{\mu}} F_Q}^* \approx \widehat{\nabla_{\boldsymbol{\mu}} F_Q} \approx \frac{\partial F_Q(\boldsymbol{x})}{\partial \boldsymbol{\mu}}$ defined by

$$\widehat{\nabla_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x}) := \frac{1}{T} \sum_{t=1}^{T} \operatorname{sign}(\boldsymbol{w}^t \cdot \boldsymbol{a}^t)(\boldsymbol{\mu} - \boldsymbol{w}^t)$$

$$\widehat{\nabla_{\boldsymbol{\mu}} F_Q}^*(\boldsymbol{x}) := \frac{1}{T} \sum_{t=1}^{T} \frac{\boldsymbol{a}^t}{\|\boldsymbol{a}^t\|} \sqrt{\frac{2}{\pi}} \exp\left[-\frac{1}{2} \left(\frac{\boldsymbol{\mu} \cdot \boldsymbol{a}^t}{\|\boldsymbol{a}^t\|} \right)^2 \right].$$

Re-parameterisation for other layers. Since the partially-aggregated final layer is now differentiable, we may also be able to use of path-wise gradients for the other layers, which would not be possible otherwise due to the non-continuity of the sign function. For example, this is possible in the single-hidden-layer ReLu case discussed above, since then g is differentiable in W_1, and $Q_1(W_1)$ has a renormalisable density. Computationally, we might implement the above by analytically finding the distribution on the "pre-activations" $W_1\boldsymbol{x}$ (trivial for the normal distribution) before sampling this and passing through the activation. With the path-wise estimator this is known as the "local reparameterization trick" (Kingma et al., 2015). It can lead to considerable computational savings on parallel minibatches compared to direct hierarchical sampling, via $\boldsymbol{a} = \boldsymbol{A}_1(W_1\boldsymbol{x})$ with $W_1 \sim Q_1$. We will utilise this in all our reparameterizable dense networks, and a variation on it to save computation when using REINFORCE in Sections 4.4 and 4.6.

This partial aggregation estimator thus leads to lower-variance gradient estimates and better-behaved training objectives across a range of other possible network structure.

4.3.1 Lower Variance Estimates of Aggregated Sign-Output Networks

Here we prove some small results to show when lower variance estimates result from using the aggregated estimator. In particular, we find that the reduction in variance from using the partial-aggregation estimator is controlled by the norm $\|\boldsymbol{\mu}\|$, so that for small $\|\boldsymbol{\mu}\|$ (as could be expected early in training) the difference can be large, while as $\|\boldsymbol{\mu}\|$ grows, the difference in variance is controlled and we could reasonably revert to the Monte Carlo (or Gibbs) estimator. Note also that as $F_Q(\boldsymbol{x}) \to \pm 1$ (as expected after training), both variances disappear.

We also show that the non-aggregated gradients are noisier in all cases than the aggregate, and demonstrate that these lower variances also lead to lower variance loss estimates.

Proposition 4.1. *With the definitions given by Eqs. (4.10) and (4.11), for all $\boldsymbol{x} \in \mathbb{R}^{d_0}$, $T \in \mathbb{N}$, and Q with normally-distributed final layer,*

$$0 \leq \mathbb{V}_Q[\widehat{F}_Q(\boldsymbol{x})] - \mathbb{V}_Q[\widehat{F}^*_Q(\boldsymbol{x})] \leq \frac{1}{T}\left(1 - \left|\operatorname{erf}\left(\frac{\|\boldsymbol{\mu}\|}{\sqrt{2}}\right)\right|^2\right).$$

Proposition 4.2. *Under the conditions of Proposition 4.1 and $y \in \{+1, -1\}$,*

$$\mathop{\mathbb{E}}_{Q}[\ell_{lin}(\widehat{F}^*_Q, (\boldsymbol{x}, y))] = \mathop{\mathbb{E}}_{Q}[\ell_{lin}(\widehat{F}_Q, (\boldsymbol{x}, y))] = \mathop{\mathbb{E}}_{Q}[\ell_0(f_\theta, (\boldsymbol{x}, y))]$$

while

$$\mathbb{V}_Q[\ell_{lin}(\widehat{F}^*_Q, (\boldsymbol{x}, y))] \leq \mathbb{V}_Q[\ell_{lin}(\widehat{F}_Q, (\boldsymbol{x}, y))] = \frac{1}{T}\mathbb{V}_Q[\ell_0(f_\theta, (\boldsymbol{x}, y))] = \frac{1}{4T}(1 - |F_Q(\boldsymbol{x})|^2).$$

From the above we see that we can use the partially-aggregated $\widehat{F}^*_Q$ in the linear loss as a lower-variance way to estimate the misclassification loss under $\theta \sim Q$ compared with a standard MC estimate from T samples. Below, we show that the final-layer gradient estimates are *strictly* improved by the partially-aggregated estimator.

Proposition 4.3. *Under the conditions of Proposition 4.1,*

$$\operatorname{Cov}[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^*(\boldsymbol{x})] + \frac{1 - 2/\pi}{T}\mathbb{I} \preceq \operatorname{Cov}[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x})],$$

where $A \preceq B \iff B - A$ is positive semi-definite. Thus, for all $\boldsymbol{u}$ with $\|\boldsymbol{u}\| = 1$,

$$\mathbb{V}[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^*(\boldsymbol{x}) \cdot \boldsymbol{u}] + \frac{1 - 2/\pi}{T} \leq \mathbb{V}[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x}) \cdot \boldsymbol{u}].$$

Note $1 - 2/\pi \approx 0.36 > 0$.

We note that the linearity of the linear loss (which is equivalent to the misclassification loss for our signed-output functions) means that Proposition 4.3 also applies to the loss gradient estimates up to scaling.

Proof of Proposition 4.1. The left identity follows directly since conditioning cannot increase the variance. Since $\mathbb{E}_Q \widehat{F}_Q(\boldsymbol{x}) = \mathbb{E}_Q \widehat{F}^\star_Q$, we also have

$$\begin{aligned}\mathbb{V}_Q[\widehat{F}_Q(\boldsymbol{x})] - \mathbb{V}_Q[\widehat{F}^\star_Q(\boldsymbol{x})] &= \frac{1}{T}\left(1 - \mathop{\mathbb{E}}_{\boldsymbol{a}}\left|\operatorname{erf}\left(\frac{\boldsymbol{\mu}\cdot\boldsymbol{a}}{\sqrt{2}\|\boldsymbol{a}\|}\right)\right|^2\right) \\ &\leq \frac{1}{T}\left(1 - \left|\operatorname{erf}\left(\frac{\|\boldsymbol{\mu}\|}{\sqrt{2}}\right)\right|^2\right).\end{aligned}$$

□

Proof of Proposition 4.2. The first equation follows from the linearity of the loss and the equivalence Eq. (4.2). For the second we note

$$\mathbb{V}_Q[\ell_{\text{lin}}(\widehat{F}_Q, (\boldsymbol{x}, y))] = \mathop{\mathbb{E}}_Q\left|\frac{1}{2}(yF_Q(\boldsymbol{x}) - y\widehat{F}_Q(\boldsymbol{x}))\right|^2 = \frac{1}{4}\mathbb{V}_Q[\widehat{F}_Q(\boldsymbol{x})]$$

and a similar result for $\widehat{F}^*_Q$, so the inequality follows from Proposition 4.1. Since $\widehat{F}_Q = \sum_{t=1}^T f_{\theta^t}$ and $f_{\theta^t} \in \{+1, -1\}$, we have $\ell_{\text{lin}}(F_Q, (\boldsymbol{x}, y)) = \frac{1}{T}\sum_{t=1}^T \ell_0(f_{\theta^t}, (\boldsymbol{x}, y))$. The first equality then follows by the independence of the θ^t and the second by substitution. □

Proof of Proposition 4.3. Basic algebra shows (using $\mathbb{E}_Q \widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q} = \mathbb{E}_Q \widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^*$) that

$$\operatorname{Cov}[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x})] - \operatorname{Cov}[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^*(\boldsymbol{x})] = \frac{1}{T}\left(\mathbb{I} - \mathbb{E}\left[\frac{\boldsymbol{a}\boldsymbol{a}^T}{\|\boldsymbol{a}\|^2}\frac{2}{\pi}e^{-\left(\frac{\boldsymbol{\mu}\cdot\boldsymbol{a}}{\|\boldsymbol{a}\|}\right)^2}\right]\right).$$

Thus for $\boldsymbol{u} \neq \boldsymbol{0}$,

$$T\boldsymbol{u}^T\left(\operatorname{Cov}[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x})] - \operatorname{Cov}[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^*(\boldsymbol{x})]\right)\boldsymbol{u} = \|\boldsymbol{u}\|^2 - \frac{2}{\pi}\mathbb{E}\left[\frac{|\boldsymbol{u}\cdot\boldsymbol{a}|^2}{\|\boldsymbol{a}\|^2}e^{-\left(\frac{\boldsymbol{\mu}\cdot\boldsymbol{a}}{\|\boldsymbol{a}\|}\right)^2}\right] \geq \|\boldsymbol{u}\|^2\left(1 - \frac{2}{\pi}\right).$$

Above we took $\boldsymbol{u}$ inside the expectation term, which is then bounded using $|\boldsymbol{u}\cdot\boldsymbol{a}|/\|\boldsymbol{a}\| \leq \|u\|$, and $e^{-|t|} \leq 1$ for all $t \in \mathbb{R}$. □

4.4 All Sign Activation Networks

Here we examine an important special case previously examined by Letarte et al. (2019): a feed-forward network with all sign activations and normal weights. For this setting combine our partial aggregation idea with another idea of "conditional sampling" is inspired by the local reinforce trick. This lets us define and use a marginalised REINFORCE-style gradient estimator for *conditional* distributions; this does not necessarily have better statistical properties but in combination with the above is much more computationally efficient.

These networks take the form

$$f^{\text{sign}}_\theta(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}_L \cdot \operatorname{sign}(W_{L-1} \ldots \operatorname{sign}(W_1\boldsymbol{x})\ldots))$$

with $\theta := \operatorname{vec}(\boldsymbol{w}_L, \ldots, W_1)$ and $W_l := [\boldsymbol{w}_{l,1} \ldots \boldsymbol{w}_{l,d_l}]^T$; $l \in \{1, ..., L\}$ are the layer indices. For the weight distribution, we choose unit-variance normal distributions on the weights, which

factorise into $Q_l(W_l) = \prod_{i=1}^{d_l} q_{l,i}(\boldsymbol{w}_{l,i})$ with $q_{l,i} = \mathcal{N}(\boldsymbol{\mu}_{l,i}, \mathbb{I}_{d_{l-1}})$. $\boldsymbol{g}(\boldsymbol{x}, \mathrm{vec}(W_1, \ldots, W_{L-1})) = \mathrm{sign}(W_{L-1} \ldots \mathrm{sign}(W_1 \boldsymbol{x}) \ldots)$ is the final layer activation, which could easily be sampled by mapping $\boldsymbol{x}$ through the first $L-1$ layers with draws from the weight distribution. Therefore we could simply apply our partial aggregation to the final layer and be finished.

Instead, we go on to make an iterative replacement of the weight distributions on each layer by conditionals on the layer activations. If a_l is the activations of the hidden layer l (with $\boldsymbol{a}_0 := \boldsymbol{x}$ the input), then the conditional distribution $\tilde{Q}_l(\boldsymbol{a}_l | \boldsymbol{a}_{l-1})$ is that defined by sampling $W_l \sim Q_l$ and setting $\boldsymbol{a}_l = \mathrm{sign}(W_l \boldsymbol{a}_{l-1})$. These conditionals can be found in closed form: we can factorise individual hidden units $\tilde{Q}_l(\boldsymbol{a}_l | \boldsymbol{a}_{l-1}) := \prod_{i=1}^{d_l} \tilde{q}_{l,i}(a_{l,i} | \boldsymbol{a}_{l-1})$, and find their activation distributions (with z a dummy integration variable in Normal distribution $\mathcal{N}(z; \mu, \sigma^2)$):

$$\tilde{q}_{l,i}(a_{l,i} = \pm 1 \,|\boldsymbol{a}_{l-1}) = \int_0^\infty \mathcal{N}(z; \pm \boldsymbol{\mu}_{l,i} \cdot \boldsymbol{a}_{l-1}, \|\boldsymbol{a}_{l-1}\|^2) \,\mathrm{d}z = \frac{1}{2}\left[1 \pm \mathrm{erf}\left(\frac{\boldsymbol{\mu}_{l,i} \cdot \boldsymbol{a}_{l-1}}{\sqrt{2}\|\boldsymbol{a}_{l-1}\|}\right)\right].$$

We can then write

$$F_Q^{\mathrm{sign}}(\boldsymbol{x}) = \sum_{\boldsymbol{a}_1 \in \{+1,-1\}^{d_1}} \tilde{Q}_1(\boldsymbol{a}_1 | \boldsymbol{x}) \quad \ldots \sum_{\boldsymbol{a}_{L-1} \in \{+1,-1\}^{d_{L-1}}} \tilde{Q}_{L-1}(\boldsymbol{a}_{L-1} | \boldsymbol{a}_{L-2}) \,\mathrm{erf}\left(\frac{\boldsymbol{\mu}_L \cdot \boldsymbol{a}_{L-1}}{\sqrt{2}\|\boldsymbol{a}_{L-1}\|}\right). \tag{4.12}$$

The number of terms is exponential in the depth so we instead hierarchically sample the $\boldsymbol{a}_l$. This leads to a "local REINFORCE trick" gradient estimator inspired by local reparameterisation, which leads to a considerable computational saving over sampling a separate weight matrix for every input. Using samples $\{(\boldsymbol{a}_1^t \ldots \boldsymbol{a}_{L-1}^t)\}_{t=1}^T \sim \tilde{Q}$, this is defined by

$$\frac{\partial F_Q^{\mathrm{sign}}(\boldsymbol{x})}{\partial \boldsymbol{\mu}_{l,i}} \approx \frac{1}{T} \sum_{t=1}^{T} \mathrm{erf}\left(\frac{\boldsymbol{\mu}_L \cdot \boldsymbol{a}_{L-1}^t}{\sqrt{2}\|\boldsymbol{a}_{L-1}^t\|}\right) \frac{\partial}{\partial \boldsymbol{\mu}_{l,i}} \log \tilde{q}_{l,i}(a_{l,i}^t | \boldsymbol{a}_{l-1}^t). \tag{4.13}$$

Comparison to Letarte et al. (2019). Our formulation in Eq. (4.13) and Eq. (4.12) somewhat resembles the PBGNet model of Letarte et al. (2019), but derived in a very different way. Both are equivalent in the single-hidden-layer case, but with more layers PBGNet uses an unusual tree-structured network to make the individual activations independent and avoid an exponential computational dependency on the depth in Eq. (4.12). This makes the above summation exactly calculable in less time, but in practice the speed up is not enough, so they resort further to a Monte Carlo approximation. Informally, this draws new samples for every layer l based on an average of those from the previous layer, $\boldsymbol{a}_l | \{\boldsymbol{a}_{l-1}^{(t)}\}_{t=1}^T \sim \frac{1}{T} \sum_{t=1}^T \tilde{Q}(\boldsymbol{a}_l | \boldsymbol{a}_{l-1}^{(t)})$. This approach is justified within their tree-structured framework but leads to an exponential KL penalty which—as hinted by Letarte et al. (2019) and shown empirically in Section 4.6—makes PAC-Bayes bound optimisation strongly favour shallower such networks. Our formulation avoids this, is more general—applying to alternative network structures—and we believe it is significantly easier to understand and use in practice.

4.5 The General Partial Aggregation Estimator

Q-aggregation can instead be *generalised* to more complex neural networks. This leads to different lower-variance Monte Carlo estimators for their outputs and gradients. Specifically, we can generalise to the form

$$r_\theta(\boldsymbol{x}) = \psi(\boldsymbol{g}(\boldsymbol{x}, \theta^g), \boldsymbol{w}) \tag{4.14}$$

with $\theta = \text{vec}(\boldsymbol{w}, \theta^g) \in \Theta \subset \mathbb{R}^{d_\theta}$, $\boldsymbol{w} \in \mathbb{R}^{d_{\text{final}}}$, and $\theta^g \in \Theta^g \subset \mathbb{R}^{d_\theta - d_{\text{final}}}$ the parameter set excluding $\boldsymbol{w}$, for the non-final layers of the network. These non-final layers are included in $\boldsymbol{g} : \mathcal{X} \times \Theta^g \to \mathcal{A}^{d_{\text{final}}} \subseteq \mathbb{R}^{d_{\text{final}}}$ and the final output passes through $\psi : \mathbb{R}^{d_{\text{final}}} \times \mathbb{R}^{d_{\text{final}}} \to \mathbb{R}$. For simplicity we have used a one-dimensional output but we note that the formulation and below derivations could extend to a vector-valued function. We require the distribution over parameters to factorise like $Q(\theta) = Q^{\boldsymbol{w}}(\boldsymbol{w})Q^g(\theta^g)$.

We then recover a similar functional form to that considered in Section 4.2.1 by rewriting the function as $\psi(\boldsymbol{a}, \boldsymbol{w})$ with $\boldsymbol{a} = \boldsymbol{g}(\boldsymbol{x}, \theta^g)$ the hidden-layer activations. We define the "aggregated" final layer function, $I(\boldsymbol{a}) := \int \psi(\boldsymbol{a}, \boldsymbol{w}) \, \mathrm{d}Q^{\boldsymbol{w}}(\boldsymbol{w})$, which may be analytically tractable. For example, with $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbb{I})$ and $\psi(\boldsymbol{a}, \boldsymbol{w}) = \text{sign}(\boldsymbol{a} \cdot \boldsymbol{w})$ as in Section 4.3.1, we recall Eq. (4.5) where $I(\boldsymbol{a}) = \text{erf}\left(\frac{\boldsymbol{\mu} \cdot \boldsymbol{a}}{\sqrt{2}\|\boldsymbol{a}\|}\right)$.

When this is analytically tractable, we can define a general form of the partial aggregation estimator using

$$R_Q(\boldsymbol{x}) = \mathop{\mathbb{E}}_{Q^g(\theta^g)} \mathop{\mathbb{E}}_{Q^{\boldsymbol{w}}(\boldsymbol{w})} [\psi(\boldsymbol{g}(\boldsymbol{x}, \theta^g), \boldsymbol{w})] = \mathop{\mathbb{E}}_{Q^g(\theta^g)} [I(\boldsymbol{g}(\boldsymbol{x}, \theta^g))].$$

We therefore obtain a general partially-aggregated estimator, $\widehat{R}^*_Q(\boldsymbol{x})$ alongside the standard Monte Carlo estimate $\widehat{R}_Q(\boldsymbol{x})$ in

$$\widehat{R}_Q(\boldsymbol{x}) := \frac{1}{T} \sum_{t=1}^{T} \psi(\boldsymbol{a}^t, \boldsymbol{w}^t) \qquad \widehat{R}^*_Q(\boldsymbol{x}) := \frac{1}{T} \sum_{t=1}^{T} I(\boldsymbol{a}^t) \tag{4.15}$$

where $\boldsymbol{a}^t = \boldsymbol{g}(\boldsymbol{x}, \theta^{g,(t)})$ for $\{\boldsymbol{w}^t, \theta^{g,(t)}\}_{t=1}^T \overset{\text{iid}}{\sim} Q$. Assuming the existence of densities q_ϕ, we can use these to also define the REINFORCE and partial-aggregation gradient estimators as

$$\widehat{\nabla_{\boldsymbol{\mu}} R_Q}(\boldsymbol{x}) := \frac{1}{T} \sum_{t=1}^{T} \psi(\boldsymbol{w}^t \cdot \boldsymbol{a}^t) \nabla_\phi \log q_\phi(\boldsymbol{w}^t) \qquad \widehat{\nabla_{\boldsymbol{\mu}} R_Q}^*(\boldsymbol{x}) := \frac{1}{T} \sum_{t=1}^{T} \nabla_\phi I_\phi(\boldsymbol{a}^t). \tag{4.16}$$

We show below partial-aggregation can lead to lower variances. We note also that under additional conditions, it might enable the use of the path-wise estimator for the gradients of non-final layers by making the overall output continous in cases where ψ is not.

Proposition 4.4. *For a function defined by Eq. (4.14) with the unbiased Q-aggregation estimators defined by Eqs. (4.15) and (4.16),*

$$\mathbb{V}_Q[\widehat{R}^*_Q(\boldsymbol{x})] \leq \mathbb{V}_Q[\widehat{R}_Q(\boldsymbol{x})].$$

and

$$\mathrm{Cov}_Q[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^{\,*}(\boldsymbol{x})] \preceq \mathrm{Cov}_Q[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x})]$$

where $A \preceq B \iff B - A$ *is positive semi-definite.*

Proof. The first part follows since conditioning cannot increase the variance. For the second part, writing $\boldsymbol{v} := \nabla_\phi \log q_\phi(\boldsymbol{w})$ and using the unbiasedness of the estimators,

$$\begin{aligned}
&\mathrm{Cov}_Q[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x})] - \mathrm{Cov}_Q[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^{\,\star}(\boldsymbol{x})] \\
&= \mathop{\mathbb{E}}_Q[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x})\,\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}(\boldsymbol{x})^T] - \mathop{\mathbb{E}}_Q[\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^{\,\star}(\boldsymbol{x})\,\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^{\,\star}(\boldsymbol{x})^T] \\
&= \frac{1}{T}\mathop{\mathbb{E}}_{\boldsymbol{a}}\left[\mathop{\mathbb{E}}_{\boldsymbol{w}}[\psi(\boldsymbol{w}\cdot\boldsymbol{a})^2\,\boldsymbol{v}\boldsymbol{v}^T] - \nabla_\phi I_\phi(\boldsymbol{a})\,\nabla_\phi I_\phi(\boldsymbol{a})^T\right] \\
&= \frac{1}{T}\mathop{\mathbb{E}}_{\boldsymbol{a}}\left[\mathrm{Cov}_{\boldsymbol{w}}[\psi(\boldsymbol{w}\cdot\boldsymbol{a})\nabla_\phi \log q_\phi(\boldsymbol{w})]\right] \succeq 0
\end{aligned}$$

where in the final lines we have used $\nabla_\phi I_{q_\phi}(\boldsymbol{a}^t) = \nabla_\phi \mathbb{E}_{\boldsymbol{w}}[\psi(\boldsymbol{w}\cdot\boldsymbol{a})] = \mathbb{E}_{\boldsymbol{w}}[\psi(\boldsymbol{w}\cdot\boldsymbol{a})\boldsymbol{v}]$ and the positive semi-definiteness of the covariance. □

4.6 Empirical Evaluation

We now move to obtain binary classifiers with guarantees for the expected misclassification error on a real dataset by optimising PAC-Bayesian bounds. Such bounds (as in Theorem 3.10) will usually involve the non-differentiable and non-convex misclassification loss ℓ_0. However, to train a neural network we usually need to replace this by a differentiable surrogate, as discussed in the introduction. Instead we use the gradient descent objective function

$$\widehat{\mathcal{L}}_{\mathrm{lin}}(\widehat{F}_Q^*) + \frac{\mathrm{KL}(Q,P)}{\lambda}. \tag{4.17}$$

based on Eq. (4.4) with the partial-aggregation estimator and its gradient estimates $\widehat{\boldsymbol{\nabla}_{\boldsymbol{\mu}} F_Q}^{\,*}$. λ is either held fixed ("**fix-λ**") or simultaneously optimised on alternative minibatches using Eq. (4.3) (with $\alpha = 1 + 10^{-5}$) throughout training for automatic regularisation tuning ("**optim-λ**").

Our experiments (Table 4.1) run on "binary"-MNIST, dividing MNIST into two classes, of digits 0–4 and 5–9. The considered neural networks had three hidden layers with 100 units per layer and **sign**, sigmoid (**sgmd**) or **relu** activations, before a single-unit final layer with sign activation. Q is chosen as an isotropic, unit-variance normal distribution with initial means drawn from a truncated normal distribution of variance 0.05. The data-free prior P is fixed equal to the initial Q, as motivated by Dziugaite and Roy (2017, Section 5 and Appendix B).

The objectives **fix-λ** and **optim-λ** were used for batch-size 256 gradient descent with Adam (Kingma and Ba, 2014) for 200 epochs. Every five epochs, the bound (for a minimising λ) was evaluated using the entire training set; the learning rate was then halved if the bound

Table 4.1: Average (from ten runs) binary-MNIST losses and bounds ($\delta = 0.05$) for the best epoch and optimal hyperparameter settings of various algorithms. Hyperparameters and epochs were chosen by bound if available and non-vacuous, otherwise by training linear loss.

	mlp	**pbg**	**Reinforce**		**Fix-λ**			**Optim-λ**		
			sign	**relu**	**sign**	**sgmd**	**relu**	**sign**	**sgmd**	**relu**
Train Linear	**0.78**	8.72	26.0	18.6	8.77	7.60	6.35	6.71	6.47	5.41
error, 1σ	*0.08*	*0.08*	*0.8*	*1.4*	*0.04*	*0.19*	*0.10*	*0.11*	*0.18*	*0.16*
Test 0–1	**1.82**	5.26	25.4	17.9	8.73	7.88	6.51	6.85	6.84	5.61
error, 1σ	*0.16*	*0.18*	*1.0*	*1.5*	*0.23*	*0.30*	*0.19*	*0.27*	*0.21*	*0.20*
Bound 0–1	-	40.8	100	100	21.7	18.8	**15.5**	22.6	19.3	16.0
error, 1σ	-	*0.2*	*0.0*	*0.0*	*0.04*	*0.17*	*0.04*	*0.03*	*0.31*	*0.05*

was unimproved from the previous two evaluations. The best hyperparameters were selected using the best bound achieved in these evaluations through a grid search of initial learning rates $\in \{0.1, 0.01, 0.001\}$, sample sizes $T \in \{1, 10, 50, 100\}$. Once these were selected training was repeated 10 times to obtain the values in Table 4.1.

λ in **optim-λ** was optimised through Eq. (4.3) on alternate mini-batches with SGD and a fixed learning rate of 10^{-4} (whilst still using the objective (4.17) to avoid effectively scaling the learning rate with respect to empirical loss by the varying λ). After preliminary experiments in **fix-λ**, we set $\lambda = m = 60{,}000$, the training set size, as is common in Bayesian deep learning.

Baselines. We also report the values of three baselines: **reinforce**, which uses the fix-λ objective without partial-aggregation, forcing the use of REINFORCE gradients everywhere; **mlp**, an unregularised non-stochastic relu neural network with tanh output activation; and the PBGNet model (**pbg**) from Letarte et al. (2019). For the latter, a misclassification error bound obtained through $\ell_0 \leq 2\ell_{\text{lin}}$ must be used as their test predictions were made by averaging multiple weight draws, giving a prediction function $\in [-1, +1]$, not $\in \{+1, -1\}$. Further, despite significant additional hyperparameter exploration, we were unable to train a three layer network through the PBGNet algorithm directly comparable to our method, likely because of the exponential KL penalty (in their Equation 17) within that framework; to enable comparison, we therefore allowed the number of hidden layers in this scenario to vary $\in \{1, 2, 3\}$. Other baseline tuning and setup was similar to the above, see Section 4.A for more details.

4.7 Discussion

The experiments demonstrate that partial-aggregation enables training of multi-layer non-differentiable neural networks in a PAC-Bayesian context, which is not possible with REINFORCE gradients and a multiple-hidden-layer PBGNet (Letarte et al., 2019). These obtained only vacuous bounds, and our misclassification bounds also improve those of a single-hidden-layer PBGNet.

Our experiments raise a couple of questions: firstly, why is it that lower variance estimates empirically lead to tighter bounds? We speculate that the faster convergence of SGD in this case takes us to a more "local" minimum of the objective, closer to our initialisation. Since most existing PAC-Bayes bounds for neural networks have a very strong dependence on this distance from initialisation through the KL term, this leads to tighter bounds. This distance could also be reduced through other methods we consider out-of-scope, such as the data-dependent bounds employed by Dziugaite and Roy (2018) and Letarte et al. (2019).

A second and harder question is asking why the non-stochastic mlp model obtains a lower overall error. The bound optimisation is empirically quite conservative, but does not necessarily lead to better generalisation; understanding this gap is a key question in the theory of deep learning.

Survey of later work. Two later publications by different authors examined very similar topics to this work, with ideas coming from this original publication. Clerico et al. (2022) combined our general partial-aggregation idea with a gradient estimator for the misclassification loss under *multi-class* linear classification with Gaussian-distributed weights from Clerico et al. (2023). Through this they were able to directly gradient-optimise PAC-Bayes bounds for deep neural networks with Gaussian weights, obtaining state-of-the-art bounds for multi-class classification.

Fortier-Dubois et al. (2023) further explored the aggregation of signed-activation networks with Gaussian weights based on our Eq. (4.12) with conditional distributions over the activations. They note that this sum, which we approximate by sub-sampling, can be directly calculated in much faster time by a dynamic program, at the expense of higher memory usage.

4.A Further Experimental Details

4.A.1 Aggregating Biases with the Sign Function

We used a bias term in our network layers, leading to a simple extension of the above formulation, omitted in the main text for conciseness:

$$\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu},\Sigma),b\sim\mathcal{N}(\beta,\sigma^2)} \operatorname{sign}(\boldsymbol{w}\cdot\boldsymbol{x}+b) = \operatorname{erf}\left(\frac{\boldsymbol{\mu}\cdot\boldsymbol{x}+\beta}{\sqrt{2(\boldsymbol{x}^T\Sigma\boldsymbol{x}+\sigma^2)}}\right)$$

since $\boldsymbol{w}\cdot\boldsymbol{x}+b\sim\mathcal{N}(\boldsymbol{\mu}\cdot\boldsymbol{x}+\beta,\boldsymbol{x}^T\Sigma\boldsymbol{x}+\sigma^2)$ and

$$\begin{aligned}\mathop{\mathbb{E}}_{z\sim\mathcal{N}(\alpha,\beta^2)} \operatorname{sign} z &= P(z\geq 0)-P(z<0)\\ &= [1-\Phi(-\alpha/\beta)]-\Phi(-\alpha/\beta)\\ &= 2\Phi(\alpha/\beta)-1 = \operatorname{erf}(\alpha/\sqrt{2}\beta).\end{aligned}$$

The bias and weight co-variances were chosen to be diagonal with a scale of 1, which leads to some simplification in the above.

4.A.2 Scaling REINFORCE

During evaluation **reinforce** draws a new set of weights for every test example, equivalent to the evaluation of the other models; but doing so during training, with multiple parallel samples, is prohibitively expensive. Two different approaches to straightforward, not partially-aggregated, gradient estimation for this case suggest themselves, arising from different approximations to the Q-expected loss of the minibatch, $B\subseteq S$ (with data indices $\mathcal{B}$). From the identities

$$\begin{aligned}\nabla_\phi \mathop{\mathbb{E}}_{\theta\sim q_\phi}\widehat{\mathcal{L}}^B(f_\theta) &= \mathop{\mathbb{E}}_{\theta\sim q_\phi}\frac{1}{|B|}\sum_{i\in\mathcal{B}}\ell(f_\theta,(\boldsymbol{x}_i,y_i))\nabla_\phi\log q_\phi(\theta)\\ &= \frac{1}{|B|}\sum_{i\in\mathcal{B}}\mathop{\mathbb{E}}_{\theta\sim q_\phi}\ell(f_\theta,(\boldsymbol{x}_i,y_i))\nabla_\phi\log q_\phi(\theta)\end{aligned}$$

we obtain two slightly different estimators for $\nabla_\phi\mathbb{E}_{\theta\sim q_\phi}\widehat{\mathcal{L}}^B(f_\theta)$:

$$\frac{1}{T|B|}\sum_{t=1}^{T}\sum_{i\in\mathcal{B}}\ell(f_{\theta^{(t,i)}},(\boldsymbol{x}_i,y_i))\nabla_\phi\log q_\phi(\theta^{(t,i)})$$

$$\frac{1}{T|B|}\sum_{i\in\mathcal{B}}\sum_{t=1}^{T}\ell(f_{\theta^t},(\boldsymbol{x}_i,y_i))\nabla_\phi\log q_\phi(\theta^t).$$

The first draws many more samples and has lower variance but is much slower computationally; even aside from the $\mathcal{O}(|B|)$ increase in computation, there is a slowdown as the optimised BLAS matrix routines cannot be used, and the very large matrices involved may not fit in memory (see Kingma et al., 2015, for more information).

Therefore, as is standard in the Bayesian Neural Network literature with the path-wise estimator, we use the latter formulation, which has a similar computational complexity to

local-reparameterisation and our marginalised REINFORCE estimator (4.13). We should note though that in preliminary experiments, the alternate estimator did not appear to lead to improved results. This clarifies the advantages of marginalised sampling, which can lead to lower variance with a similar computational cost.

4.A.3 Dataset Details

We used the MNIST dataset version 3.0.1, available online at http://yann.lecun.com/exdb/mnist/, which contains 60,000 training examples and 10,000 test examples, which were used without any further split, and rescaled to lie in the range $[0,1]$. For the "binary"-MINST task, the labels $+1$ and -1 were assigned to digits in $\{5,6,7,8,9\}$ and $\{0,1,2,3,4\}$, respectively, and images were scaled into the interval $[0,1]$.

4.A.4 Hyperparameter Search for Baselines

The baseline comparison values offered with our experiments were optimized similarly to the above, for completeness we report everything here.

The MLP model had three hidden ReLu layers of size 100 each trained with Adam, a learning rate $\in \{0.1, 0.01, 0.001\}$ and a batch size of 256 for 100 epochs. Complete test and train evaluation was performed after every epoch, and in the absence of a bound, the model and epoch with lowest train linear loss was selected.

For PBGNet we choose the values of hyperparameters from within these values giving the least bound value. Note that, unlike in Letarte et al. (2019), we do not allow the hidden size to vary $\{\in 10, 50, 100\}$, and we use the entire MNIST training set as we do not need a validation set. While attempting to train a three hidden layer network, we also searched through the hyperparameter settings with a batch size of 64 as in the original, but after this failed, we returned to the original batch size of 256 with Adam. All experiments were performed using the code from the original paper, available at https://github.com/gletarte/dichotomize-and-generalize.

Since we were unable to train a multiple-hidden-layer network through the PBGNet algorithm, for this model only we explored different numbers of hidden layers $\in \{1,2,3\}$.

4.A.5 Final Hyperparameter Settings

In Table 4.2 we report the hyperparameter settings used for the experiments in Table 4.1 after exploration. To save computation, hyperparameter settings that were not learning (defined as having a whole-train-set linear loss of > 0.45 after ten epochs) were terminated early. This was also done on the later evaluation runs, where in a few instances the fix-λ sigmoid network failed to train after ten epochs; to handle this we reset the network to obtain the main experimental results.

For clarity we repeat here the hyperparameter settings and search space:

Table 4.2: Chosen Hyperparameter settings and additional details for results in Table 4.1. Best hyperparameters were chosen by bound if available and non-vacuous, otherwise by best training linear loss through a grid search as described in Section 4.6 and Section 4.A.4. Run times are rounded to nearest 5 min.

	mlp	**pbg**	**Reinforce**		**Fix-λ**			**Optim-λ**		
			sign	**relu**	**sign**	**relu**	**sgmd**	**sign**	**relu**	**sgmd**
Init. LR	0.001	0.01	0.1	0.1	0.01	0.1	0.1	0.01	0.1	0.1
Samples, T	-	100	100	100	100	50	10	100	100	10
Hid. Layers	3	1	3	3	3	3	3	3	3	3
Hid. Size	100	100	100	100	100	100	100	100	100	100
Mean KL	-	2658	15,020	13,613	2363	3571	3011	5561	3204	4000
Runtime/min	10	5	40	40	35	30	25	35	30	25

- Initial Learning Rate $\in \{0.1, 0.01, 0.001\}$.
- Training Samples $\in \{1, 10, 50, 100\}$.
- Hidden Size $= 100$.
- Batch Size $= 256$.
- Fix-λ, $\lambda = m = 60{,}000$.
- Number of Hidden Layers $= 3$ for all models, except PBGNet $\in \{1, 2, 3\}$.

4.A.6 Implementation and Runtime

Experiments were implemented using Python and the TensorFlow library (Abadi et al., 2015). Reported approximate runtimes are for execution on a NVIDIA GeForce RTX 2080 Ti GPU.

Chapter 5

On Margins and Derandomisation in PAC-Bayes

We give a general recipe for derandomising PAC-Bayesian bounds using margins, with the critical ingredient being that our randomised predictions concentrate around some value. The tools we develop straightforwardly lead to margin bounds for various classifiers, including linear prediction—a class that includes boosting and the support vector machine—single-hidden-layer neural networks with an unusual erf activation function, and deep ReLU networks. Further, we extend to partially-derandomised predictors where only some of the randomness is removed, letting us extend bounds to cases where the concentration properties of our predictors are otherwise poor.

5.1 Introduction

PAC-Bayesian generalisation bounds have recently seen a resurgence of interest after the comparative successes of a series of papers applying them to deep neural networks, beginning with Dziugaite and Roy (2017, 2018); Neyshabur et al. (2018), and Letarte et al. (2019); Zhou et al. (2019). One can use these to understand where to apply techniques and motivate new learning algorithms (as for example Foret et al., 2021), as well as provide certification for a given predictor and address the more ambitious goal of understanding generalisation.

One particularly useful aspect of PAC-Bayesian results compared to standard PAC/VC results is that they are non-uniform: the tightness of the guarantee on the generalisation error depends on the specific predictor chosen, not merely on its performance on the training set. This is necessary in cases where our broad class can easily overfit—as for example with many neural networks architectures, which were shown by Zhang et al. (2021) to be able to fit random training labels—since any guarantee must then selectively favour predictors

which are reasonable given real data. If our strategic choice of learner turns out to be a good match in practice to the data-generating distribution, the bound should reflect this.

But how to measure this match based only on the training data? A common approach is that we should not just take into account the train error of a given predictor but also its *confidence*. One way to formalise this is the concept of a margin, introduced to bound the error of the perceptron (Novikoff, 1962) and later used to motivate the support vector machine (Cortes and Vapnik, 1995). A confident predictor with a large margin on a given example will be locally robust to parameter (and data) perturbations. If this is true across the dataset our bounds should reflect this and be tighter. From the perspective of Occam's razor this robustness leads to a large set of valid perturbations giving near-equivalent (in terms of dataset outputs) predictors; some predictor in this set is thus likely to be close to a "simple" prediction rule of the kind that we should generally favour (Schapire et al., 1998).

Remarkably, this idea of parameter robustness—as measured by margins—can be formalised through the lens of PAC-Bayes, which more typically bounds the loss of randomised predictors (although work on derandomised PAC-Bayesian bounds is as old as the field, particularly when relating to the average prediction; bounds holding with high probability over a sampled predictor directly drawn from the PAC-Bayes posterior appear in *e.g.* Alquier and Biau, 2013a; Catoni, 2007; Guedj and Alquier, 2013a). After picking a derandomised prediction rule, we can construct a (weighted) class of "proxy" predictors that approximate this rule, with the size and diversity of this class growing with the allowed margin. Since a larger such class is more likely to overlap strongly with our PAC-Bayesian prior, tighter bounds are obtained for larger margins. This idea has been used informally by Langford and Seeger (2001) and Langford and Shawe-Taylor (2003): here we formalise and extend it considerably.

A critical ingredient in this process is the construction of randomised classifiers that have favourable concentration properties—simply, that the parameters are robust to perturbations—so that their deviations from a central derandomised prediction rule are bounded with high probability. The insight that these deviations need only be controlled with high probability rather than certainty is crucial in obtaining better rates and simplifying proofs, and opens the door to the application of powerful concentration of measure results.

We use this to prove bounds for a variety of useful situations. Firstly L_2 and L_1 normed linear prediction; in the L_2 "hard-margin" case this improves on the bound of Bartlett and Shawe-Taylor (1998) and matches the lower bound of Grønlund et al. (2020); the other bounds relax to the state of the art with simpler proofs. Next for one-hidden-layer neural networks with erf activations, involving an interesting new randomised predictor taking the form of a mixture distribution. Finally we prove a bound for deep ReLU networks which is a slight improvement on that of and has proof ideas drawing from Neyshabur et al. (2018).

We go further and introduce the idea of partially-derandomised predictors, which remove only some of the randomness from the proxy predictors. This enables us to obtain bounds in further cases which would otherwise be difficult to address. We hope that these tools can be further developed to address situations where classical bounding techniques have not worked well (such as in deep neural networks), and that our derived corollaries—such as that for linear prediction—can be used in practice for the provision of self-certifying predictors and model selection.

Outline. In Section 5.2 we give an overview of our general method and of the bounds proved with it. We introduce our technique for proving margin bounds through PAC-Bayes in Section 5.3 before giving such bounds for a variety of different prediction function types and discussing previous bounds in Section 5.4. In Section 5.5 we show how our method can be generalised to "partially-derandomised' predictors before summarising in Section 5.6.

Notation. We will consider classification of i.i.d. examples from a distribution, $\mathcal{D}$, on some product space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, by vector-valued predictors in a function space $\mathcal{H} \subset \widehat{\mathcal{Y}}^{\mathcal{X}}$. For binary classification $\mathcal{Y} = \{+1, -1\}$, $\widehat{\mathcal{Y}} = \mathbb{R}$ and we take the sign of the output as our prediction, while for multi-class prediction, $\mathcal{Y} = [d_{\text{out}}] := \{1, \ldots, d_{\text{out}}\}$, $\widehat{\mathcal{Y}} = \mathbb{R}^{d_{\text{out}}}$ and the maximum argument is the prediction. In our specific function space bounds $\mathcal{X}$ will always be a subset of a real vector space; when the dimension is finite we denote it by d_{in}.

The multi-class margin, $M : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ is the mapping

$$M(f, x, y) := f(\boldsymbol{x})[y] - \max_{y' \neq y} f(\boldsymbol{x})[y']$$

where by $f(x)[y]$ we indicate the yth component of $f(x)$. In a slight abuse of notation we also define the binary margin $M(f, x, y) := yf(x)$. The margin loss is $\ell_\gamma(f, z) = \mathbf{1}\{M(f, z) \leq \gamma\}$; note it has the rescaling property $\ell_\gamma(f, z) = \ell_{\beta\gamma}(\beta f, z)$ for any $\beta > 0$. [1]

The margin error is $\mathcal{L}_\gamma(f) := \mathbb{P}_{z\sim\mathcal{D}}\{M(f, z) \leq \gamma\}$, so $\mathcal{L}(f) := \mathcal{L}_0(f)$ is the misclassification loss or probability of error, and $\widehat{\mathcal{L}}_\gamma(f) := m^{-1}|\{(x, y) \in S : yf(x) \leq \gamma\}|$ for the empirical margin error (defined for some sample $S \sim \mathcal{D}^m$ and margin $\gamma \geq 0$). We will also use the abbreviation $\mathbb{P}_{\mathcal{D}}(A) = \mathbb{P}_{z\sim\mathcal{D}}(A)$ and similar for the expectation.

5.2 Overview of Results

For each function f in the class $\mathcal{H}$ we define a corresponding proxy distribution Q such that

$$\forall z \in \mathcal{Z} \ : \ \mathbb{P}_{g\sim Q}(M(g, z) - M(f, z) > \gamma/2) \leq \epsilon.$$

[1] In a slight abuse of notation we will freely interchange $M(f, x, y)$ as per Chapter 2 and the less cluttered $M(f, z)$ where $z = (x, y)$.

In other words, the proxy distribution margins concentrate around f. This ϵ high probability term is a crucial difference from the covering number approach, and considerably simplifies proofs, since only a concentration bound is required. For example, if Q is a sub-Gaussian distribution with mean f for any fixed $\boldsymbol{x}$, the above is possible with exponentially small ϵ (*i.e.* $\epsilon \in \mathcal{O}(\exp(-\gamma^2))$). For example, we might choose the class of linear predictors $f_{\boldsymbol{w}}$ with parameters $\boldsymbol{w}$; a possible sub-Gaussian Q adds zero-mean Gaussian noise to these parameters.

When this happens, we can replace the loss of f with a loss of Q, at the price of a larger margin requirement to avoid an error (and a small additive term). Specifically, we show that the above implies

$$\mathcal{L}_0(f) - \mathop{\mathbb{E}}_{g\sim Q} \mathcal{L}_{\gamma/2}(g) \leq \epsilon \qquad \text{and} \qquad \mathop{\mathbb{E}}_{g\sim Q} \widehat{\mathcal{L}}_{\gamma/2}(g) - \widehat{\mathcal{L}}_{\gamma}(f) \leq \epsilon.$$

The purpose of constructing Q is that a PAC-Bayesian bound can then be freely applied to it using the loss $\ell_{\gamma/2}$. Such a PAC-Bayesian bound then involves a complexity term $\mathrm{KL}(Q,P)$ for prior P, which can sometimes be made quite small for clever choices of P. We optimise Q to trade off the additive ϵ terms (essentially by adjusting the tightness of the concentration) with the usual KL term (which increases as the proxy shrinks to a point). This gives a margin bound for f in which explicit dependence on the proxy has been eliminated.

Since in most cases we would like our bounds to be simultaneously valid for every $\gamma > 0$, a final step is often a union bound of some kind over different possible cases for this. We briefly overview the specific bounds proved in this work below.

L_2-normed linear predictors. This is the class of linear binary classifiers, $f_{\boldsymbol{w}}(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$, with $\|\boldsymbol{w}\|_2 \leq 1$, and $\|\boldsymbol{x}\| \leq R$. In this case, our class of proxy functions is constructed by adding standard Gaussian noise to the weight vector, and a zero-mean Gaussian prior on the weight vector. The proof also involves an additional scaling trick which simplifies the KL divergence and enables free choice of $\gamma > 0$, without a union bound argument.

A relaxation of our high probability margin bounds on this class (w.l.o.g. setting $R = 1$) is given by

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \widehat{\mathcal{L}}_{\gamma}(f_{\boldsymbol{w}}) + \mathcal{O}\left(\sqrt{\frac{\gamma^{-2}\log m + \log\frac{1}{\delta}}{m}}\right),$$

holding simultaneously for all $\gamma > 0$ and $\|\boldsymbol{w}\|_2 \leq 1$. Note however that our general form is significantly tighter than this weak relaxation.

L_1/L_∞-normed linear predictors. The above bound with L_2 norms for $\mathcal{X}$ and w, applies to situations such as the SVM. We also provide bound for linear classification under different norm constraints, where the L_1 norm of the weights and L_∞ norm of the features is restricted, as in boosting. The overall function takes the form $f_{\boldsymbol{w}} = \sum w_i x_i$ which can be approximated

in a sub-Gaussian way (since the features are bounded) by sampling index i with probability w_i. Since this does not give us control of the sub-Gaussian constant, we average multiple draws like $T^{-1}\sum_{t=1}^{T} x_{i(t)}$, with $i(t) \sim \text{Categ}(\boldsymbol{w})$ for each $i \in [T]$. The number of averaged features T, controls the size of ϵ (with $\epsilon \in \mathcal{O}(e^{-T})$).

Overall, this leads to bounds for such predictors that relax to

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}}) + \mathcal{O}\left(\sqrt{\frac{\gamma^{-2}\log m \log d_{\text{in}} + \log\frac{1}{\delta}}{m}}\right),$$

where d_{in} is the number of input features. Note again that the final form of our bound is considerably tighter than the above.

SHEL Networks Although the above contributions present significant improvements over previous margin bounds, their proof ideas still closely correlate with existing work. In the case of our bounds for SHEL networks, we introduce significant new ideas in the constriction of the proxy distribution, leading to a bound with new features, in particular a lack of dependence on the width. SHEL (single hidden erf layer) networks are defined straightforwardly for normalised data by $f_{U,V}(\boldsymbol{x}) = V\,\text{erf}(U\boldsymbol{x})$.

The proxy distribution has three components: the error function (erf) activation arises through the aggregation of a $\text{sign}(\boldsymbol{w}\cdot\boldsymbol{x})$ with normally-distributed $\boldsymbol{w}$; this is turned into a vector output by multiplying by a vector random variable supported on $\{+1,-1\}^{d_{\text{out}}}$; then multiple hidden units are differentiated by using a mixture distribution over the above, with each component of the mixture corresponding to a different hidden unit in $f_{U,V}$. To control the sub-Gaussian constant we use the averaging trick from the L_1 case; essentially this lets us trade off the size of ϵ with the KL divergence.

Overall, this leads to bounds for SHEL networks like

$$\mathcal{L}_0(F_{U,V}) \leq \widehat{\mathcal{L}}_\gamma(F_{U,V}) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{d_{\text{hid}}}}{\gamma\sqrt{m}}\left(V_\infty\|U-U^0\|_F + \|V\|_F\right)\right),$$

with $V_\infty := \max_{ij}|V_{ij}|$ and U^0 being a data-free matrix of "prior" hidden-unit features.

ReLU Neural Networks. Another class of classifiers we consider is that of fully-connected feedforward neural networks with weight matrices $\{W_i\}_{i=1}^h$. In this case we also construct a class of proxy functions by adding Gaussian noise to the weight matrices, refining a result from Neyshabur et al. (2018) to bound ϵ as a function of the noise scaling. The trade off between ϵ and the KL is controlled by adjusting the bandwidth of the randomisation, as is essentially also the case for the L_2 predictors. Overall, this leads to a ReLU Neural Network result like

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}}) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{h\prod_i\|W_i\|_2^2}{m\gamma^2}\sum_i\frac{\|W_i\|_F^2}{\|W_i\|_2^2}}\right),$$

where h is an upper bound on the number of units in any layer. This result is very similar to that from Neyshabur et al. (2018), but the full formulation removes some logarithmic terms and our proof is arguably much simpler.

5.3 General Method

We introduce a set of approximations or proxies for our chosen function f, defined by[2]

$$\mathcal{P}(f,\epsilon,\gamma) = \left\{ Q \in \mathscr{P}(\mathcal{H}) : \forall z \in \mathcal{Z}, \forall \zeta \in \{+1,-1\} : \mathop{\mathbb{P}}_{g\sim Q}(\zeta(M(f,z) - M(g,z)) > \gamma/2) \leq \epsilon \right\}.$$

This result simply defines $\mathcal{P}$ as a set of distributions over predictors, so that the margins of these distributions concentrate around f at everywhere; the parameters γ and ϵ just define the sharpness of this concentration. This implies that we can replace f by Q at cost of the additional margin and ϵ term. Our main PAC-Bayesian result is the following.

Theorem 5.1. *Fix data distribution $\mathcal{D}$, $\delta \in (0,1)$, margin $\gamma > 0$ and prior $P \in \mathscr{P}(\mathcal{H})$. With probability $\geq 1-\delta$ over sample $S \sim \mathcal{D}^m$, simultaneously for any $f \in \mathcal{H}$ and $\epsilon \geq 0$,*

$$\mathcal{L}_0(f) \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(f) + \epsilon, \frac{\inf_{Q \in \mathcal{P}(f,\epsilon,\gamma)} \mathrm{KL}\left(Q,P\right) + \log \frac{2\sqrt{m}}{\delta}}{m} \right) + \epsilon.$$

Proof. From the trivial identity $\mathbf{1}\{t \in A\} \leq \mathbf{1}\{t \in B\} + \mathbf{1}\{t \in A\}\mathbf{1}\{t \notin B\}$ on sets A, B,

$$\begin{aligned}
\ell_0(f,z) - \mathop{\mathbb{E}}_{g\sim Q} \ell_{\gamma/2}(g,z) &= \mathop{\mathbb{E}}_{g\sim Q}[\mathbf{1}\{M(f,z) \leq 0\} - \mathbf{1}\{M(g,z) \leq \gamma/2\}] \\
&\leq \mathop{\mathbb{E}}_{g\sim Q}[\mathbf{1}\{M(f,z) \leq 0\}\mathbf{1}\{M(g,z) > \gamma/2\}] \\
&\leq \mathop{\mathbb{E}}_{g\sim Q}[\mathbf{1}\{M(g,z) - M(f,z) > \gamma/2\}].
\end{aligned}$$

Similarly, $\mathbb{E}_{g\sim Q}\, \ell_{\gamma/2}(g,z) - \ell_\gamma(f,z) \leq \mathbb{E}_{g\sim Q}[\mathbf{1}\{M(f,z) - M(g,z) > \gamma/2\}]$. Thus for any $Q \in \mathcal{P}(f,\epsilon,\gamma)$ and at all $z \in \mathcal{Z}$

$$\ell_0(f,z) - \mathop{\mathbb{E}}_{g\sim Q} \ell_{\gamma/2}(g,z) \leq \epsilon \qquad \text{and} \qquad \mathop{\mathbb{E}}_{g\sim Q} \ell_{\gamma/2}(g,z) - \ell_\gamma(f,z) \leq \epsilon.$$

These hold for any $f \in \mathcal{H}$, and are true independently of the sample. Substitute these into Theorem 3.12 with the $\ell_{\gamma/2}$ margin loss, which is simultaneously valid for any Q, and take the infimum over ϵ and Q to obtain the result. □

The PAC-Bayesian part of our proofs is therefore extremely simple; the difficulty arises in constructing a sensible proxy distribution and prior. The following results are useful for that.

[2] Note that by using $\zeta \in \{+1,-1\}$ in $\zeta(M(f,z) - M(g,z))$ rather than $|M(f,z) - M(g,z)|$ we can avoid the need for two sided bounds, removing a later factor of 2 in the sub-Gaussian case.

5.3.1 Sub-Gaussian Concentration

The primary type of concentration we will use is the sub-Gaussian kind; the below gives a weaker sufficient condition of this type.

Proposition 5.1. *Let $Q \in \mathscr{P}(\mathcal{H})$ be such that for all $x \in \mathcal{X}, y \in \mathcal{Y}, t \in \mathbb{R}$, we have $\mathbb{E}_{g\sim Q}[g(\boldsymbol{x})] = f(\boldsymbol{x})$ and*

$$\mathop{\mathbb{E}}_{g\sim Q}[\exp(t(f(\boldsymbol{x})[y] - g(\boldsymbol{x})[y]))] \leq \exp\left(\frac{1}{2}\sigma^2 t^2\right).$$

Then $Q \in \mathcal{P}(f, \epsilon, \gamma)$ for any $\gamma > 0$ with

$$\epsilon \leq \begin{cases} \exp\left(-\gamma^2/8\sigma^2\right) & \text{for } \mathcal{Y} = \mathbb{R}, \\ \exp\left(-\gamma^2/16\sigma^2\right) & \mathcal{Y} = \mathbb{R}^{d_{\text{out}}}. \end{cases}$$

Proof. Considering the zero-mean random variable $X = f(x)[y] - g(x)[y]$ for $g \sim Q$ (which is σ^2-sub-Gaussian) and fixed $(x, y) \in \mathcal{Z}$, the Chernoff bound (see Boucheron et al., 2013, for example), immediately implies the one-sided tail bounds

$$\max(\mathbb{P}(X > t), \mathbb{P}(-X > t)) \leq e^{t^2/2\sigma^2}$$

for all $t > 0$. In the binary margin case, $M(f, z) = yf(x)$ which is either $f(x)$ or $-f(x)$; setting $t = \gamma/2$ in the above therefore gives the bound.

In the multi-class case we consider the upper bound obtained by letting y' achieve the maximum margin for g; then $M(f, z) \leq f(x)[y] - f(x)[y']$, so

$$\mathop{\mathbb{P}}_{g\sim Q}\left\{M(f, z) - M(g, z) > \frac{\gamma}{2}\right\} \leq \mathop{\mathbb{P}}_{g\sim Q}\left\{f(x)[y] - f(x)[y'] - g(x)[y] + g(x)[y'] > \frac{\gamma}{2}\right\}.$$

Since both $f(x)[y] - g(x)[y]$ and $f(x)[y'] - g(x)[y']$ are σ^2-sub-Gaussian, their sum is $2\sigma^2$-sub-Gaussian and the bound follows by repeating the process on with signs reversed. □

A simpler analysis could have used the bound $|M(f, z) - M(g, z)| \leq 2\max_{y\in\mathcal{Y}} |f(x)[y] - g(x)[y]|$ instead, leading to similar PAC-Bayes bounds; our more sophisticated variant improves constants in some derived bounds and removes a factor of d_{out}, the number of classes.

Sometimes, we may have a method for producing a sub-Gaussian proxy, but no easy way to control the tightness of its concentration; for example when considering bounded prediction functions. In this case, we can resort to averaging over multiple samples, with an additional KL cost, as shown by the following useful theorem.

Theorem 5.2. *Suppose there is a $\rho_f \in \mathscr{P}(\mathcal{H})$ for each $f \in \mathcal{H}$ which is σ^2-sub-Gaussian with mean f. Fix $T \in \mathbb{N}$, $\gamma > 0$ and $\pi \in \mathscr{P}(\mathcal{H})$; with probability at least $1 - \delta$ for every f*

$$\mathcal{L}_0(f) \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(f) + e^{-\frac{1}{8}T\gamma^2\sigma^{-2}}, \frac{T\,\mathrm{KL}\left(\rho_f, \pi\right) + \log\frac{2\sqrt{m}}{\delta}}{m}\right) + e^{-\frac{1}{8}T\gamma^2\sigma^{-2}}$$

in the binary classification case. In the multi-class case $\frac{1}{8}$ is replaced by $\frac{1}{16}$ in the exponentials.

Proof. We average over T i.i.d. samples from ρ_f, so that our overall proxy function looks like $g(\boldsymbol{x}) = T^{-1}\sum_{t=1}^{T} g^{(t)}(\boldsymbol{x})$ for $g^{(t)} \overset{\text{iid}}{\sim} \rho_f$; this resulting proxy function is σ^2/T-sub-Gaussian. Since T and $\pi \in \mathscr{P}(\mathcal{H})$ are data-independent, we can use Theorem 5.1 with this proxy, and a prior defined by the same averaging technique but with using π. Combing with Proposition 5.1 we obtain the result. □

Other concentration assumptions. We note that although we only discuss sub-Gaussian concentration here, it is possible to require other concentration properties, for example sub-exponential ones; our framework easily accommodates these. Sub-Gaussianity is only the simplest way to ensure such concentration, and we primarily consider it in our later results as it already leads to simple proofs of margin bounds in multiple settings.

5.3.2 Hard Margin Bounds

We can also give a slightly different form of bound which we use when considering hard margins, *i.e.* when we have a predictor which interpolates the data perfectly with some margin γ_* so that $\widehat{\mathcal{L}}_{\gamma_*}(f) = 0$. Note that unlike Theorem 5.1, we must fix $\epsilon \geq 0$ in advance (although this is what we commonly do in most of our later proofs anyway).

Theorem 5.3. *Fix data distribution $\mathcal{D}$, $\delta \in (0,1)$, $\gamma > 0$, $\epsilon \in [0, \frac{1}{2}]$ and prior $P \in \mathscr{P}(\mathcal{H})$. With probability $\geq 1-\delta$ over sample $S \sim \mathcal{D}^m$, simultaneously for any $f \in \mathcal{H}$, observing that $\widehat{\mathcal{L}}_\gamma(f) = 0$ implies*

$$\mathcal{L}_0(f) \leq \frac{\inf_{Q \in \mathcal{P}(f,\epsilon,\gamma)} \mathrm{KL}\left(Q, P\right) + \log \frac{1}{\delta}}{m} + 4\epsilon \log \frac{1}{\epsilon}.$$

Proof. We obtain from Catoni's bound Theorem 3.9 for $Q \in \mathcal{P}(f, \epsilon, \gamma)$ that

$$C\Phi_C(\mathcal{L}_0(f) - \epsilon) - C(\epsilon + \widehat{\mathcal{L}}_\gamma(f)) \leq \frac{\mathrm{KL}\left(Q, P\right) + \log \delta^{-1}}{m}$$

with probability at least $1-\delta$ simultaneously for all f. From this, with probability at least $1-\delta$ if we observe that $\widehat{\mathcal{L}}_\gamma(f) = 0$ we find[3]

$$C\Phi_C(\mathcal{L}_0(f) - \epsilon) - C\epsilon \leq \frac{\mathrm{KL}\left(Q, P\right) + \log \delta^{-1}}{m}$$

Since ϵ and $\mathcal{L}_0(f)$ are data independent, we can choose an optimal C, and the overall bound follows from Lemma 5.4. □

5.3.3 Relation to Covering

Here we discuss how our bounds can be used to derive a standard covering approach as a sub-case; we show that this leads to certain problems, which are circumvented by the concentration approach. A further consequence is that covering-based bounds usually lead to

[3] For propositions $p(f)$, $\forall f : p(f) \implies \forall f : q(f) \to p(f)$, where $\to$ is the material conditional.

"uniform" bounds which are subject to problems discussed in Nagarajan and Kolter (2019). All the bounds we provide in later sections are non-uniform and avoid these pitfalls.

By setting $\epsilon = 0$ with certain choices of prior we can obtain a fairly standard "covering" approach: call N_γ a γ-net of $\mathcal{H}$, if for any $f \in \mathcal{H}$, there exists $g \in N_\gamma$ such that $|M(f,z) - M(g,z)| \leq \gamma$ for all $z \in \mathcal{Z}$. If we choose a prior supported everywhere on a $\gamma/2$-net for $\mathcal{H}$, we can achieve $\epsilon = 0$ for a proxy $Q \ll P$, *i.e.* with a non-infinite KL divergence. The simplest approach chooses P_0 as uniform on these covering functions in $N_{\gamma/2}$ so that

$$\inf_{Q \in \mathcal{P}(f,\epsilon=0,\gamma)} \mathrm{KL}\left(Q, P\right) \leq \log |N_{\gamma/2}|$$

where $|N_{\gamma/2}|$ is the cardinality of the net. A more sophisticated choice of non-uniform prior enables structural risk minimisation-type covering number bounds.

However, such bounds will typically be dependent on the *dimension* of the parameter space, as demonstrated by the following result for linear classification. Our bounds in the following sections will avoid this dimension-dependence by permitting $\epsilon > 0$.

Theorem 5.4. *Consider binary classification with functions $f_{\boldsymbol{w}} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ and $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}, \|\boldsymbol{x}\|_2 \leq 1$. For any prior P on weights in $\mathbb{R}^{d_{\text{in}}}$, there exists a $\boldsymbol{w}$ such that $\|vecw\|_2 \leq 1$ and*

$$\inf_{Q \in \mathcal{P}(f,\epsilon=0,\gamma)} \mathrm{KL}\left(Q, P\right) \geq \Omega(d_{\text{in}}).$$

Proof of Theorem 5.4. Firstly we note that $\mathcal{H}$ is a space of linear functions, so we can represent Q by its corresponding distribution over $\mathbb{R}^{d_{\text{in}}}$. In the second step we substitute the margin definition and use this property.

$$\begin{aligned}
&\mathcal{P}(f_{\boldsymbol{w}}, \epsilon = 0, \gamma) \\
&= \left\{ Q \in \mathscr{P}(\mathcal{H}) : \forall z \in \mathcal{Z}, \forall \zeta \in \{+1, -1\} : \mathop{\mathbb{P}}_{g \sim Q}(\zeta(M(f,z) - M(g,z)) > \gamma/2) = 0 \right\} \\
&= \left\{ Q \in \mathscr{P}(\mathbb{R}^{d_{\text{in}}}) : \forall (\boldsymbol{x}, y) \in \mathcal{Z}, \forall \zeta \in \{+1, -1\} : \mathop{\mathbb{P}}_{\boldsymbol{u} \sim Q}(\zeta y(\langle \boldsymbol{x}, \boldsymbol{w} \rangle - \langle \boldsymbol{x}, \boldsymbol{u} \rangle) > \gamma/2) = 0 \right\} \\
&= \left\{ Q \in \mathscr{P}(\mathbb{R}^{d_{\text{in}}}) : \forall \boldsymbol{x} \in \mathcal{X} : \mathop{\mathbb{P}}_{\boldsymbol{u} \sim Q}(\langle \boldsymbol{x}, \boldsymbol{w} - \boldsymbol{u} \rangle > \gamma/2) = 0 \right\} \\
&= \left\{ Q \in \mathscr{P}(\mathbb{R}^{d_{\text{in}}}) : \forall \boldsymbol{x} \in \mathcal{X}, \forall \boldsymbol{u} \in \mathrm{Supp}(Q) : \langle \boldsymbol{x}, \boldsymbol{w} - \boldsymbol{u} \rangle \leq \gamma/2 \right\} \\
&= \left\{ Q \in \mathscr{P}(\mathbb{R}^{d_{\text{in}}}) : \forall \boldsymbol{u} \in \mathrm{Supp}(Q) : \|\boldsymbol{w} - \boldsymbol{u}\|_2 \leq \gamma/2 \right\} \\
&= \left\{ Q \in \mathscr{P}(\mathbb{R}^{d_{\text{in}}}) : \mathrm{Supp}(Q) \subset \mathrm{Ball}(\boldsymbol{w}, \gamma/2) \right\}.
\end{aligned}$$

Combining with the Lemma 5.5 we find that $\mathrm{KL}\left(Q, P\right) \geq -\log P[\mathrm{Ball}(\boldsymbol{w}, \gamma/2)]$.

Since $\boldsymbol{w}$ can be any such that $\|\boldsymbol{w}\|_2 \leq 1$ and thus chosen in an adversarial manner based on P, P must not over-weight any such ball. The P which minimises the KL over all such admissible choices of Q is thus uniform over the set of possible balls $\mathrm{Ball}(\boldsymbol{w}, \gamma/2)$, which is the set $\mathrm{Ball}(\boldsymbol{0}, 1 + \gamma/2)$ (since $\|\boldsymbol{w}\|_2 \leq 1$).

Basic calculation then shows that

$$\mathrm{KL}(Q,P) \geq -\log \frac{\mathrm{vol}[\mathrm{Ball}(\boldsymbol{w},\gamma/2)]}{\mathrm{vol}[\mathrm{Ball}(0,1+\gamma/2)]} = d\log\frac{2+\gamma}{\gamma} = \Omega(d).$$

□

5.4 Main Results

5.4.1 L_2 Linear Prediction

First we demonstrate our framework in action by deriving generalisation bounds for L_2-normed linear predictors. Specifically, we consider linear predictors with an L_2 norm on both the weights and the input vector. These bounds essentially follow from an initial Gaussian assumption combined with the sharp (sub-Gaussian) concentration of the predictor output around its mean. We hope they can be useful for self-certification in the low data regime, and for model (or kernel) selection without a validation set.

Theorem 5.5. *Consider the binary classification setting with $\mathcal{X}$ in Hilbert space with $\|x\|_2 \leq 1$ and $f_{\boldsymbol{w}}(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w}\rangle$. Fix $\delta \in (0,1)$. With probability $\geq 1-\delta$ over the sample simultaneously for all $\|\boldsymbol{w}\|_2 \leq 1$ (and thus $f_{\boldsymbol{w}}$) and all $\gamma > 0$,*

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}}) + \frac{1}{m}, \frac{4\gamma^{-2}\log m + \log\frac{2\sqrt{m}}{\delta}}{m}\right) + \frac{1}{m}.$$

Additionally, under the same conditions and probability, provided $\gamma_\star = \max\{\gamma > 0 : \widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}}) = 0\}$ exists,

$$\mathcal{L}_0(f_w) \leq \frac{8\gamma_\star^{-2}\log m + \log\frac{1}{\delta}}{m}.$$

We call the first and second results above "hard" and "soft" margin respectively, and note that the first bound can be relaxed to

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}}) + \sqrt{\frac{\widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}})\cdot C}{m}} + \frac{C+\sqrt{C}+2}{m}, \tag{5.1}$$

where we define $C := 2\log(2/\delta) + 9(R/\gamma)^2\log m$.

In the hard margin case Theorem 5.5 improves on Bartlett and Shawe-Taylor (1998, Theorem 4.17, which uses fat-shattering) by a factor of $\mathcal{O}(\log m)$, matching the lower bound of Grønlund et al. (2020, Theorem 4). In the soft margin case the relaxation Eq. (5.1) is of the same order as the state-of-the-art bound given by Grønlund et al. (2020) but with explicitly stated constants. The small-kl form given in Theorem 5.5 is considerably tighter still. We emphasise also the extreme simplicity of our proof compared to that given for the state-of-the-art in Grønlund et al. (2020, Theorem 2 with proof in Section 2, p.3-7).

We also note other weaker soft margin results, such as Bartlett and Mendelson (2002, Theorem 22, using Rademacher complexity), Bartlett and Shawe-Taylor (1998, Theorem

3.12), and McAllester (2003), which is itself an attempt to find a expression for the implicit PAC-Bayesian result of Langford and Shawe-Taylor (2003), and uses a similar proof method to ours. We also acknowledge the result of Hanneke and Kontorovich (2021, Theorem 1) which gives a *algorithm*-dependent hard-margin bound specifically for the SVM output, and eliminates the $\log m$ factor. This is provably optimal in the algorithm-dependent (as ours is in the general) case, which is shown in Grønlund et al. (2020, Theorem 5). We discuss these existing bounds at greater length in Section 5.B.

We note that the soft-margin formulation of the bound is true universally across $\gamma > 0$, allowing the bound can be optimised for γ in $\mathcal{O}(m)$ time. If the margin is large for most examples, we can choose γ so that $\widehat{\mathcal{L}}_\gamma$ is small and Eq. (5.1) shows that the bound is $\mathcal{O}(1/m)$ (which is of the same order as the hard-margin bound). Since the minimum margin can be sensitive to outliers, this bound will often be tighter than the hard-margin one.

We note that we can consider the slightly more general case $\|x\|_2 \leq R$ by simply scaling the margin to γ/R, so scaling the data does not affect the bound. However, we note that the bound can sometimes be decreased by *normalising* the data (as this maximises the margin for every data point), so we recommend this when using such predictors.

Proof of Theorem 5.5. We use the scaling property of the margin loss to obtain a result that holds for any $\gamma > 0$ simultaneously. Specifically, we note that for any $\beta > 0$ we can scale f by β^{-1} and give the same overall predictions, so $\mathcal{L}_0(f) = \mathcal{L}_0(\beta^{-1} f)$, while the empirical margin risk is scaled as $\widehat{\mathcal{L}}_\theta(\beta^{-1} f) = \widehat{\mathcal{L}}_{\beta\theta}(f)$.

Now consider a prior on the weight of $\mathcal{N}(0, I)$, and a proxy Q on $f_{\boldsymbol{u}}$ where $\boldsymbol{u} \sim \mathcal{N}(\beta^{-1}\boldsymbol{w}, I)$. The KL divergence from this prior is $\frac{1}{2}\beta^{-2}\|\boldsymbol{w}\|_2^2 \leq \frac{1}{2}\beta^{-2}$. Further, Q is sub-Gaussian with constant $\sigma = 1$ with mean $f_{\boldsymbol{w}/\beta}$, since $f_{\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{w}/\beta}(\boldsymbol{x})$ has distribution $\mathcal{N}(0, \|x\|_2^2)$ and $\|\boldsymbol{x}\| \leq 1$. Therefore, by Proposition 5.1, $Q \in \mathcal{P}(f_{\boldsymbol{w}/\beta}, e^{-\frac{1}{8}\theta^2}, \theta)$.

Applying Theorem 5.1 to $f_{\boldsymbol{w}/\beta}$ with margin θ, and using the scaling property of the margin loss with the linearity $f_{\boldsymbol{w}/\beta} = \beta^{-1} f_{\boldsymbol{w}}$ therefore gives

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \inf_{\beta > 0} \left[\mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}_{\beta\theta}(f_{\boldsymbol{w}}) + \exp\left(-\frac{1}{8}\theta^2\right), \frac{\frac{1}{2}\beta^{-2} + \log\frac{2\sqrt{m}}{\delta}}{m} \right) + \exp\left(-\frac{1}{8}\theta^2\right) \right].$$

We note $\beta > 0$ can be freely chosen based on the data in the above, so we set $\theta^2 = 8\log(m)$ (which is data-independent), and re-parameterise $\beta = \gamma/\theta$ to get the soft margin result.

If instead we use Theorem 5.3, and the re-parameterise $\beta = \gamma_*/\theta$,

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \inf_{\beta > 0 : \widehat{\mathcal{L}}_{\beta\theta}(f_{\boldsymbol{w}}) = 0} \left[\frac{\frac{1}{2}\beta^{-2} + \log\frac{1}{\delta}}{m} + 4\exp\left(-\frac{1}{8}\theta^2\right) \cdot \frac{1}{8}\theta^2 \right] = \frac{4(1+\gamma_*^{-2})\log m + \log\frac{1}{\delta}}{m}.$$

THe statement follows since $\gamma_* \leq 1$. □

5.4.2 L_1/L_∞ Linear Prediction

Theorem 5.5 is a bound under L_2 norms for $\mathcal{X}$ and w, applying to situations such as the SVM. Here we provide a bound for linear classification under different norm constraints, where the L_1 norm of the weights and L_∞ norm of the features is restricted, as in boosting. Specifically, this is for the case when $\|\boldsymbol{w}\|_1 \leq 1$ and $\|\boldsymbol{x}\|_\infty \leq 1$ (without loss of generality, since we can always re-scale the margin).

These results are similar but not identical to to the k-th margin bound of Gao and Zhou (2013), or the central result of Langford and Seeger (2001). The fundamental proof idea is to approximate our predictor by a randomised, unweighted, sum of features, as originally proposed by Schapire et al. (1998); the boundedness of these features leads to sub-Gaussian concentration around their mean, similarly to in Theorem 5.5.

Theorem 5.6. *Consider the binary classification setting with $\mathcal{X} \subset \mathbb{R}^{d_{\text{in}}}$ with $\|x\|_\infty \leq 1$ and let $f_{\boldsymbol{w}}(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle$. Fix $\delta \in (0,1)$ and $\gamma > 0$. With probability $\geq 1-\delta$ over the sample simultaneously for all $\|\boldsymbol{w}\|_1 \leq 1$,*

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}}) + \frac{1}{m}, \frac{\lceil 8\gamma^{-2}\log m \rceil \log 2d_{\text{in}} + \log\frac{2\sqrt{m}}{\delta}}{m}\right) + \frac{1}{m}.$$

Since $\gamma \in (0,1)$ for non-vacuous results, a union bound argument can be used to extend the above to fixed-precision γ.

Proof. For simplicity we assume initially that the weights are non-negative; negative weights can later be included through the standard method of doubling the dimension. Our prediction function has the form $f_{\boldsymbol{w}}(x) = \sum_{k=1}^{d_{\text{in}}} w_k x_k$. We set ρ_f to output x_i where the index $i \sim \text{Categ}(\boldsymbol{w})$; π has the same form with a uniform distribution on indices. Since $\|x\|_\infty \leq 1$, this is bounded and therefore 1-sub-Gaussian.

$\mathrm{KL}\left(\rho_f, \pi\right) = \log d_{\text{in}} - H[\boldsymbol{w}]$ where $H[\boldsymbol{w}]$ is the entropy of a categorical variable with (normalised) weights $\boldsymbol{w}$. This expression using $H[\boldsymbol{w}]$ could be explicitly used (or with a non-uniform prior) to improve the bound, as in Seeger et al. (2001); we will ignore this here and just use the upper bound $\mathrm{KL}\left(\rho_f, \pi\right) \leq \log d_{\text{in}}$.

Setting $T = \lceil 8\gamma^{-2}\log m \rceil$ in Theorem 5.2 and using a standard dimension doubling trick to include negative weights gives the final result. □

5.4.3 SHEL Networks

Here we prove generalisation bounds for a one-hidden-layer neural network with a slightly unusual erf activation function that looks much like a tanh or other sigmoidal-type functions as more commonly used. This is inspired by the work of Germain et al. (2009); Letarte et al. (2019) and Biggs and Guedj (2021), which consider averaging over the predictions

of functions like $f_w : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}, x \mapsto \text{sign}(\boldsymbol{w} \cdot \boldsymbol{x})$, where $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{u}, I)$, giving "aggregated" prediction functions of the form

$$\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{u},I)} \text{sign}(\boldsymbol{w}\cdot\boldsymbol{x}) = \text{erf}(\boldsymbol{u}\cdot\boldsymbol{x}/\sqrt{2}\|\boldsymbol{x}\|_2). \tag{5.2}$$

With a clever choice of weight distribution, we can combine the sub-Gaussian concentration of the bounded sign function with its tractable average to get bounds for one-hidden layer networks of the following form.

Definition. Single Hidden Erf Layer (SHEL) Network[4]*: Given* $V \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$ *and* $U \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}$*, this is the neural network* $f_{U,V} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$ *defined by*

$$f_{U,V}(x) = V \,\text{erf}\left(\frac{Ux}{\sqrt{2}\|x\|_2}\right) \tag{5.3}$$

where the erf *activation function is applied elementwise.*

Theorem 5.7. *Fix prior parameters* $U^0 \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}$ *and* $\delta \in (0,1), \alpha > 1$*. With probability* $\geq 1-\delta$ *over any* $\gamma > 0, U \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}, V^{d_{\text{out}} \times d_{\text{hid}}}$*,*

$$\mathcal{L}_0(f_{U,V}) \leq \text{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(f_{U,V}) + \frac{1}{m}, \frac{C(U,V,\gamma) + \log\frac{4\sqrt{m}}{\delta}}{m}\right) + \frac{1}{m}.$$

with $V_\infty := \max_{ij}|V_{ij}|$ *in*

$$C(U,V,\gamma) = 17\left(\frac{\alpha V_\infty d_{\text{hid}}}{\gamma}\right)^2 \left(\frac{\|U-U^0\|_F^2}{2d_{\text{hid}}} + \frac{\|V\|_F^2}{V_\infty^2 d_{\text{hid}}}\log 2\right)\log m + 2\log\left(\frac{\log(\alpha^2 V_\infty d_{\text{hid}}/\gamma)}{\log\alpha}\right).$$

a SHEL network $f_{U,V}$ *as defined in Eq.* (5.3)*.*

Prior parameters. This generalisation bound for depends on a set of prior parameters (or "random features"), U_0, chosen independently of the training data, for example the initialisation of the network (this choice has been extensively discussed in the literature, beginning with Dziugaite and Roy, 2017).

Scaling dependence. At first glance this bound might appear to grow with width, since although the norm terms are usually seen to be roughly constant under increasing d_{hid}, the $\sqrt{d_{\text{hid}}}$ term is obviously not. However, this is not necessarily true: the range of the network (and thus maximum margin) is bounded by $d_{\text{hid}}V_\infty$, so provided the margin per-unit (γ/d_{hid} for the γ used in the bound) remains constant, the bound would actually decrease with d_{hid}.

To emphasise this, we note that the above bound is unchanged under two simple transformations, which ensures dimensional consistency (if it were not, we could perform these operations to obtain a possibly arbitrarily tight bound). (1) Scale V; the bound and

[4]Note that we will use a slightly more general definition in Chapter 6

norm term exactly cancel since we can scale γ by the same amount and obtain the same empirical margin loss. (2) Double the width of network, with exact copies of weights in the copy: we can again double γ for a fixed margin loss, while the squared norms also double.

Empirical evaluation. Although the main contribution of this paper is in the refinement of methods for proving PAC-Bayes margin bounds, in Section 8.5 we also make some empirical evaluations of Theorem 5.7, and a partially derandomised generalisation of it. Since these bounds were in general vacuous, we adopt the procedures of Jiang et al. (2020b) and Dziugaite et al. (2020) to compare such bounds; training to a fixed cross-entropy of 0.3 and setting margin loss $\widehat{\mathcal{L}}_\gamma(f_{U,V}) = 0.2$, we examine changes in the big-O complexity measure in Theorem 5.7 versus generalisation error under different hyperparameter changes. Our complexity measure is predictive under training set size changes and somewhat predictive under learning rate changes, but like most such measures (Dziugaite et al., 2020), it is not predictive under changes of width, implying the per-unit margin decreases significantly with width. We interpret this as follows: at initialisation $\boldsymbol{u}_i \cdot \boldsymbol{x} \sim d_{\text{in}}^{-\frac{1}{2}}$ is small, so if weights stay near their initialisation (as is usual for wider networks trained by SGD), units are less saturated and the per-unit margin decreases. This is avoided in lower dimensions or by scaling up the weight initialisations with d_{in}, but as this is further from the typical SGD training scenario we avoid this.

Optimisation of the prior. We have in the above empirical evaluation neglected to utilise optimised data-dependent priors (as initiated by Ambroladze et al., 2006; Parrado-Hernández et al., 2012), which has been demonstrated to vastly tighten bounds in the case of neural networks due to the stability of training. These ideas have been heavily used in recent papers for neural networks (Dziugaite et al., 2021, for example) and were found to significantly improve the actual bound values in preliminary experiments, in some cases leading to non-vacuous (although loose) results. As our focus is more on the theoretical side of providing a method to prove margin bounds, we decided to focus on the data-independent case for simplicity.

Related Work. Here we mention previous work (Letarte et al., 2019; Biggs and Guedj, 2021, Chapter 4) on PAC-Bayesian neural networks with erf activations, as well as a wide range of results obtaining generalisation bounds for neural networks, in particular Neyshabur et al. (2019) which focuses specifically on one-hidden-layer networks. Banerjee et al. (2020) uses similar methods to ours by looking at Gaussian perturbations to the weights of a deep ReLU network, but their bound relies on the strong assumption of bounds on the Hessian and gradients of the network across weight values, and as formulated is not evaluable nor

decreases with m.

We also highlight an interesting connection to a strand of work (Daxberger et al., 2021; Kristiadi et al., 2020) in the Bayesian neural network literature, where networks involving only some randomised weights (effectively, partially-derandomised networks) were found to offer many of the benefits of more general networks while offering considerable computational saving.

Proof of Theorem 5.7. Let ρ_f be a probability distribution on functions taking the form

$$g(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w} \cdot \boldsymbol{x})\boldsymbol{r},$$

where $(\boldsymbol{w}, \boldsymbol{r}) \in \mathbb{R}^d \times \{+1, -1\}^{d_{\text{out}}}$ is sampled through the following hierarchical procedure: draw a mixture component $k \sim \text{Uniform}(d_{\text{hid}})$; then $\boldsymbol{w} \in \mathbb{R}^d$ from Gaussian $\mathcal{N}(U_{k,\cdot}, I)$ (with mean vector as a row of U); then for each $i \in [d_{\text{out}}]$ draw a component of $\boldsymbol{r} \in \{+1, -1\}^{d_{\text{out}}}$ such that $\mathbb{E}\boldsymbol{r}[i] = V_{ik}/V_\infty$.

It is straightforward to see from Eq. (5.2) that the expectation of this is proportional to $f_{U,V}$,

$$\mathop{\mathbb{E}}_{g\sim\rho_f}[g(\boldsymbol{x})] = \frac{1}{d_{\text{hid}} V_\infty} \sum_{k=1}^{d_{\text{hid}}} V_{ik} \operatorname{erf}\left(\frac{U_{k;\cdot} \cdot x}{\sqrt{2}\|x\|_2}\right) = \frac{1}{V_\infty d_{\text{hid}}} f_{U,V}(\boldsymbol{x}).$$

Further, since $g(\boldsymbol{x})[y] \in [+1, -1]$, ρ_f is 1-sub-Gaussian.

Applying this result in Theorem 5.2 with margin $\theta > 0$ and $T = \lceil 16\theta^{-2} \log m \rceil$ gives

$$\mathcal{L}_0(f_{U,V}) \le \text{kl}^{-1}\left(\widehat{\mathcal{L}}_\theta\left(\frac{f_{U,V}}{V_\infty d_{\text{hid}}}\right) + \frac{1}{m}, \frac{\lceil 16\theta^{-2}\log m\rceil \operatorname{KL}(\rho_f, \pi) + \log\frac{2\sqrt{m}}{\delta}}{m}\right) + \frac{1}{m}.$$

where we used the equivalence $\mathcal{L}_0(f_{U,V}/V_\infty d_{\text{hid}}) = \mathcal{L}_0(f_{U,V})$.

We define a $\pi \in \mathscr{P}(\mathcal{H})$ in the same functional form, but with a data-free matrix U^0 for the Gaussian compontnets and zero mean $V^0 = 0$. Writing $\rho_f(\boldsymbol{w}, \boldsymbol{r})$ for the density on $\boldsymbol{w}, \boldsymbol{r}$ and similarly for the conditionals and marginals, we find by repeatedly using the chain rule for KL divergnce (Cover and Thomas, 2006) that

$$\begin{aligned}
\operatorname{KL}\left(\rho_f(\boldsymbol{w},\boldsymbol{r}), \pi(\boldsymbol{w},\boldsymbol{r})\right) &\le \operatorname{KL}\left(\rho_f(k,\boldsymbol{w},\boldsymbol{r}), \pi(k,\boldsymbol{w},\boldsymbol{r})\right) \\
&= \operatorname{KL}\left(\rho_f(k), \pi(k)\right) + \operatorname{KL}\left(\rho_f(\boldsymbol{w},\boldsymbol{r}|k), \pi(\boldsymbol{w},\boldsymbol{r}|k)\right) \\
&= \operatorname{KL}\left(\rho_f(\boldsymbol{w},\boldsymbol{r}|k), \pi(\boldsymbol{w},\boldsymbol{r}|k)\right) \\
&= \operatorname{KL}\left(\rho_f(\boldsymbol{w}|k), \pi(\boldsymbol{w}|k)\right) + \operatorname{KL}\left(\rho_f(\boldsymbol{r}|k), \pi(\boldsymbol{r}|k)\right) \\
&= \frac{1}{d_{\text{hid}}} \sum_{k=1}^{d_{\text{hid}}} \frac{\|U_{k,\cdot} - U^0_{k,\cdot}\|_2^2}{2} + \frac{1}{d_{\text{hid}}} \sum_{k=1}^{d_{\text{hid}}} \sum_{i=1}^{d_{\text{out}}} h(V_{ik}/V_\infty) \\
&\le \frac{\|U - U^0\|_F^2}{2d_{\text{hid}}} + \frac{\|V\|_F^2}{V_\infty^2 d_{\text{hid}}} \log 2.
\end{aligned}$$

Above we used the KL divergence for a Normal variable and defined $h(x) = \mathrm{KL}(X, \mathrm{Uniform}(\{+1,-1\}))$ for variable X with mean x supported on $\{+1,-1\}$; from Lemma 5.3, $h(x) \leq x^2 \log 2$.

It remains to cover possible values of θ. Firstly we note that for $\theta \geq 1$ the bound is trivially true by the boundedness of $f(\boldsymbol{x})$, and thus we need only consider $\theta^{-2} > 1$. For $\alpha > 1$ and $i = 0, 1, \ldots$, set $\theta_i = \alpha^{-i}$ and $\delta_i = \delta/2(i+1)^2$. Applying the union bound over the above equation with these parameters we get that with probability at least $1 - (\pi^2/6)\delta \geq 1 - 2\delta$ that the above is true for each pair of θ_i and δ_i. We choose the largest θ_i such that $\theta_i \leq \theta < \theta_{i-1}$, so that $i \leq 1 - \log_\alpha(\theta)$. Since $\widehat{\mathcal{L}}_\theta \leq \widehat{\mathcal{L}}_{\theta_i}$ is increasing, $1/\theta_i \leq \alpha/\theta$, and $\log(1/\delta_i) \leq \log(1/\delta) + 2\log(2 + \log_a(1/\theta)) = \log(1/\delta) + 2\log(\log(\alpha^2/\theta)/\log(\alpha))$. Substituting, we obtain the result. □

Generalisation to bounded functions. We note that in the proof of Theorem 5.7 we can replace the sign activation functions used in the proxy function distribution by any bounded activations, for example sigmoid. Indeed, any feature map which is bounded and independent from the final layer is possible. The caveat is that the obtained networks have modified activation functions which may not be analytically tractable.

5.4.4 Feed-forward ReLU Networks

Finally, we give a bound for deep feed-forward ReLU networks, similar in form and proof to that given by Neyshabur et al. (2018). Although the new result shares the same shortcomings (as discussed in, for example, Dziugaite et al., 2020), we hope our simplified proof and unifying perspective will help clear the way for future improvements.

The new bound replaces a factor of d, the number of layers, with one of $\sqrt{\log m}$. This comes from a simplification in the proof where we merely require $\epsilon \leq \mathcal{O}(m^{-1})$ concentration of the margin rather than insisting $\epsilon = 0$ as in the original. Bounding this term for simple Gaussian weights with the same perturbation bound as their proof, gives a simple form for KL divergence. Combination with Theorem 5.2 and a cover of different weight variances and margins completes the proof.

A second difference between Theorem 5.8 and the bound of Neyshabur et al. (2018) is the appearance of prior matrices (to bring the bound into line with others which often set these to the initialisation) and the norm bound $W_\star$. This $W_\star$ term arises from these prior matrices and can be eliminated if the prior matrices are set to zero. This is because re-scaling the weights and margins will then not affect the bound due to the positive homogeneity of the ReLU (so $\|W_i\|_2/\gamma = \|\tilde{W}_i\|_2/\tilde{\gamma}$ and $\|W_i\|_F/\|W\|_2 = \|\tilde{W}_i\|_F/\|\tilde{W}\|_2$ for re-scaled $\tilde{W}_i$ and $\tilde{\gamma}$).

Theorem 5.8. *Let $f_{\boldsymbol{w}} : \mathcal{X} \to \mathbb{R}^{d_{\text{out}}}$ on $\mathcal{X} = \{x \in \mathbb{R}^{d_{\text{in}}} : \|x\|_2 \leq R\}$ be a fully-connected, feed-forward ReLU neural network weights $\boldsymbol{w} = \text{vec}(W_i)$ for W_i with $i \in [L]$ layers and no more than h units per layer. Fix $\delta \in (0,1), W_\star > 0$ and prior weight matrices $\{W_i^0\}_{i=1}^L$. With probability at least $1-\delta$ simultaneously for all $W_i : \|W_i\|_2 \leq W_\star$ and $\gamma > 0$, $\mathcal{L}_0(f_{\boldsymbol{w}}) - \widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}})$ is upper bounded by*

$$\mathcal{O}\left(\sqrt{\frac{h(R\prod_{i=1}^L \|W_i\|_2)^2 \log(mhL)}{\gamma^2 m} \cdot \sum_{i=1}^L \frac{\|W_i - W_i^0\|_F^2}{\|W_i\|_2^2} + \frac{\log\frac{1}{\delta} + L\log\log W_\star}{m}}\right).$$

In order to prove Theorem 5.8we begin with two lemmas used.

Lemma 5.1 (Neyshabur et al., 2018; Lemma 2, Perturbation Bound.)**.** *In the setting of Theorem 5.8, for any layer weights W_i, $x \in \mathcal{X}$ and weight perturbations U_i such that $\|U_i\|_2 \leq L^{-1}\|W_i\|_2$,*

$$\|f_{\boldsymbol{w}}(\boldsymbol{x}) - f_{\boldsymbol{w}+\boldsymbol{u}}(\boldsymbol{x})\|_2 \leq eR\left(\prod_{i=1}^L \|W_i\|_2\right)\sum_{i=1}^L \frac{\|U_i\|_2}{\|W_i\|_2}$$

where $\boldsymbol{w} = \text{vec}(W_i)$ and $\boldsymbol{u} = \text{vec}(U_i)$.

Lemma 5.2. *Let $\boldsymbol{w} = \text{vec}(W_i)$ and $\boldsymbol{u} = \text{vec}(U_i)$ for weights W_i and perturbations U_i. If the perturbations are isotropic Gaussian with per-layer variances σ_i^2, then for all $0 < \gamma < 2\sup_{\boldsymbol{w}\in\mathcal{X}, y\in[d_{\text{out}}]} |f_{\boldsymbol{w}}(\boldsymbol{w})[y]|$,*

$$\mathbb{P}\{|M(f_{\boldsymbol{w}}, z) - M(f_{\boldsymbol{w}+\boldsymbol{u}}, z)| > \gamma/2\} \leq 2h\sum_{i=1}^L \exp\left(-\frac{1}{32h}\left(\frac{\gamma\|W_i\|_2}{\sigma_i eR(\prod_i \|W_i\|_2)}\right)^2\right).$$

Proof. Suppose $\|U_i\|_2 \leq c_i\gamma\|W_i\|_2$ with $c_i^{-1} = 4eRL\prod_i \|W_i\|_2$for every i. Then $\|U_i\|_2 \leq L^{-1}\|W_i\|_2$ if $\gamma < 2eR\prod_i \|W_i\|_2$. This is true for the allowed γ since $|f_{\boldsymbol{w}}(\boldsymbol{x})[y]| \leq R\prod_i \|W_i\|_2$ by the Lipschitz property. For such perturbations the assumptions of Lemma 5.1 are satisfied, and by substitution $\|f_{\boldsymbol{w}}(\boldsymbol{x}) - f_{\boldsymbol{w}+\boldsymbol{u}}(\boldsymbol{x})\|_2 \leq \gamma/4$. Therefore by the logical contrapositive, if $\|f_{\boldsymbol{w}}(\boldsymbol{x}) - f_{\boldsymbol{w}+\boldsymbol{u}}(\boldsymbol{x})\|_2 > \gamma/4$ then for some i we have $\|U_i\|_2 > c_i\gamma\|W_i\|_2$.

If the perturbations are randomised, we can use the above combined with the union bound to show that (letting $y' \neq y$ achieve the maximum margin)

$$\begin{aligned}
&\mathbb{P}\{|M(f_{\boldsymbol{w}}, z) - M(f_{\boldsymbol{w}+\boldsymbol{u}}, z)| > \gamma/2\} \\
&\leq \mathbb{P}\{|f_{\boldsymbol{w}}(\boldsymbol{x})[y] - f_{\boldsymbol{w}}(\boldsymbol{x})[y'] - f_{\boldsymbol{w}+\boldsymbol{u}}(\boldsymbol{x})[y] + f_{\boldsymbol{w}+\boldsymbol{u}}(\boldsymbol{x})[y']| > \gamma/2\} \\
&\leq \mathbb{P}\{2\|f_{\boldsymbol{w}}(\boldsymbol{x}) - f_{\boldsymbol{w}+\boldsymbol{u}}(\boldsymbol{x})\|_\infty > \gamma/2\} \\
&\leq \mathbb{P}\{\|f_{\boldsymbol{w}}(\boldsymbol{x}) - f_{\boldsymbol{w}+\boldsymbol{u}}(\boldsymbol{x})\|_2 > \gamma/4\} \\
&\leq \mathbb{P}\{\exists i : \|U_i\|_2 > c_i\gamma\|W_i\|_2\} \\
&\leq \sum_{i=1}^d \mathbb{P}\{\|U_i\|_2 > c_i\gamma\|W_i\|_2\}.
\end{aligned}$$

We set U_i to be isotropic Gaussian with per-layer variances of σ_i^2. To complete the proof we use a result of Tropp (2012) for such Gaussian random matrices, that

$$\mathbb{P}\{\|U_i\|_2 > t\} \leq 2he^{-t^2/2h\sigma_i^2}.$$

□

Proof of Theorem 5.8. We choose Q in the form $f_{\boldsymbol{w}+\boldsymbol{u}}$ with Gaussian U_i and per-layer variances σ_i. P has the form $f_{\boldsymbol{w}^0+\boldsymbol{u}}$ where $\boldsymbol{w}^0 = \mathrm{vec}(W_i^0)$ for data-independent W_i^0. Suppose $f_{\boldsymbol{w}}$ has weights such that

$$\sigma_i^{-2} \geq 32h\left(\frac{eR(\prod_i \|W_i\|_2)}{\gamma\|W_i\|_2}\right)^2 \log(mhL) \tag{5.4}$$

for every i and the chosen γ. Then from Lemma 5.2 and (a relaxation of) Theorem 5.1 with high probability for all such W_i,

$$\mathcal{L}_0(f_{\boldsymbol{w}}) \leq \widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}}) + \frac{4}{m} + \sqrt{\frac{1}{2m}\left(\sum_{i=1}^{d}\frac{\|W_i - W_i^0\|_F^2}{2\sigma_i^2} + \log\frac{2\sqrt{m}}{\delta}\right)}.$$

This is a generalisation bound holding for any such weights.

We complete the proof by constructing covers for σ_i and γ. We only need to consider $\gamma < R\prod_i \|W_i\|_2 =: C_\gamma$ (an upper bound on the range of the function) as otherwise the $\widehat{\mathcal{L}}_\gamma$ term is 1 and the bound is vacuous. Since $\|W_i\|_2 \leq W_\star$ for all i we have that $\sigma_i^{-2} \geq 32e^2h\|W_i\|_2^{-2} \geq 32e^2hW_\star^{-2}$ and $\sigma_i \leq 15h^{-1/2}W_\star^2 =: C_\sigma$.

For $t = 0, 1, 2, \ldots$ choose margins $\gamma^{(t)} = C_\gamma/2^t$ and let the bound for this margin hold with probability $\delta^{(t)} = \delta/(t+1)^2$, so that taking a union bound the above holds simultaneously for every $\gamma^{(t)}$ with probability at least $1 - \pi^2\delta/6 \geq 1 - 2\delta$. To find a bound holding simultaneously for all γ, we choose the t such that $\gamma^{(t)} \leq \gamma < \gamma^{(t-1)}$, and then replace this term with γ by using the facts that $\widehat{\mathcal{L}}_{\gamma^{(t)}} \leq \widehat{\mathcal{L}}_\gamma$, $1/\gamma^{(t)} \leq 2/\gamma$, and $\log(1/\delta^{(t)}) \leq \log(1/\delta) + 2\log\log_2(4C_\gamma/\gamma)$.

Repeating this same covering process for every choice of σ_i, we obtain with probability at least $1 - 2\delta$ simultaneously for all γ, σ_i (and thus also for the tightest σ_i satisfying Eq. (5.4)) that $\mathcal{L}_0(f_{\boldsymbol{w}}) - \widehat{\mathcal{L}}_\gamma(f_{\boldsymbol{w}})$ is upper bounded by

$$\frac{4}{m} + \sqrt{\frac{1}{2m}\left(4\sum_{i=1}^{L}\frac{\|W_i - W_i^0\|_F^2}{2\sigma_i^2} + \log\frac{2(L+1)\sqrt{m}}{\delta} + 2\log\log_2\frac{4C_\gamma}{\gamma} + \sum_{i=1}^{L} 2\log\log_2\frac{4C_\sigma}{\sigma_i}\right)}$$

$$\in \mathcal{O}\left(\sqrt{\frac{hR^2\left(\prod_{i=1}^{L}\|W_i\|_2^2\right)\log(mhL)}{\gamma^2 m} \cdot \sum_{i=1}^{L}\frac{\|W_i - W_i^0\|_F^2}{\|W_i\|_2^2} + \frac{\log\frac{1}{\delta} + L\log\log W_\star}{m}}\right).$$

□

5.5 Partially-Derandomised Results

Here we consider a slight generalisation of our results. Say we wish to consider margin bounds with a predictive distribution $\widetilde{Q}$, rather than a deterministic predictor f; perhaps $\widetilde{Q}$ is lower-variance than Q, or "partially-derandomises" it. For example, we might want to "stack" our SHEL network on top of a ReLU network with Gaussian weights used as a feature map. Through our extension we can obtain a margin bound for this case where the final layers are deterministic but the ReLU ones are not, adding only a KL term for the randomised ones.

This is interesting because it enables empirical comparisons with deeper networks on more complex datasets without severe overfitting, which we hope can form a stepping stone between totally-randomised PAC-Bayesian bounds and non-random margin bounds, while helping in the understanding of one-hidden-layer network generalisation. This provides a middle ground between a series of works obtaining bounds for stochastic neural networks such as Dziugaite and Roy (2017), and those providing margin bounds for non-stochastic DNNs, such as (in a PAC-Bayesian context) Neyshabur et al. (2018). We discuss this specific case further in Section 8.5.

In general, suppose we have predictive distribution $\widetilde{Q}$; we construct a coupling with Q, $\pi \in \Pi(\widetilde{Q}, Q)$, where by $\Pi(\widetilde{Q}, Q)$ we denote the set of distributions on $\mathcal{H} \times \mathcal{H}$ with marginals $\widetilde{Q}$ and Q, that satisfies the margin condition. Specifically, we redefine

$$\mathcal{P}(\widetilde{Q}, \epsilon, \gamma) = \left\{ Q \in \mathscr{P}(\mathcal{H}) : \inf_{\pi \in \Pi(\widetilde{Q}, Q)} \sup_{z \in \mathcal{Z}, \zeta = \pm 1} \mathbb{P}_{(f,g)\sim\pi} (\zeta(M(f,z) - M(g,z)) > \gamma/2) \leq \epsilon \right\}.$$

It is straightforward to adapt the proof of Theorem 5.1 to replace $\mathcal{L}_0(w)$ by $\mathbb{E}_{w\sim\widetilde{Q}} \mathcal{L}_0(w)$, $\widehat{\mathcal{L}}_\gamma(w)$ by $\mathbb{E}_{w\sim\widetilde{Q}} \widehat{\mathcal{L}}_\gamma(w)$, and use the redefinition of $\mathcal{P}$. Similarly, the sub-Gaussian results Proposition 5.1 and Theorem 5.2 can be adapted in the same way by defining the following slight generalisation of sub-Gaussianity for coupling $\pi \in \Pi(\widetilde{Q}, Q)$:

$$\mathbb{E}_{\pi} \exp(t(f(x)[y] - g(x)[y])) \leq \exp\left(t^2\sigma^2/2\right)$$

and $E_{f\sim\widetilde{Q}} f(x)[y] = E_{g\sim Q} g(x)[y]$, for all $t \in \mathbb{R}$, $(x, y) \in \mathcal{Z}$.

Note that stacking our SHEL network on a randomised feature map with the proxy definition from the original proof satisfies this condition due to the boundedness of the final output. We thus adapt results for SHEL networks to such a partially-derandomised setting, giving experimental evaluations in Section 8.5.

5.6 Conclusion

In this work we have provided a unified framework for derandomising PAC-Bayes bounds using margins. In particular this leads to new bounds or greatly simplified proofs for a

variety of settings. It also enables the novel idea of partial-derandomisation, which provides a halfway house for estimators which cannot be so easily derandomised.

Specifically: we provided in Theorem 5.5 bounds for L_2-regularised linear classification which improve upon classical results and match the state-of-the-art order given in Grønlund et al. (2020) while providing explicit (and small) constants as well as a considerably simplified proof. We then gave further bounds in Theorem 5.7 for the novel setting of single-hidden-erf-layer (SHEL) networks, as well as a bound in Theorem 5.8 that improves slightly on Neyshabur et al. (2018, Theorem 1). We feel that SHEL networks have much potential as a setting to explore margin bounds for deterministic neural networks—matching contemporary practice—through improved concentration techniques and priors. In Section 5.5 we extend our results to the novel situation where we only partially de-randomise, for example by simultaneously learning a (randomised) feature map.

Although we recognise that our final results for linear prediction and deep ReLU networks are relatively small improvements on existing results, we believe our radically simplified proofs and explicit link of derandomisation to concentration (a link which has been occasionally used implicitly in proofs in the literature) are significant and novel contributions to a difficult and central problem in their own right. We show in Section 5.3.3 how reducing PAC-Bayes derandomisation to a covering approach leads to a sub-optimal dependence on the dimension, which is observed in some prior results such as the prior ReLU bound (Neyshabur et al., 2018). We believe that by highlighting this issue we point the way forward to further simplifications and improvements, and hope the machine learning and statistics community will leverage these tools going forward.

Similarly, we feel that a major implication of our work is to show that for non-vacuous neural network margin bounds, we need tighter bounds on the concentration properties of networks. Networks are observed to be quite robust to perturbation in practice, far better than the Lipschitz constant-dependent bounds of our Theorem 5.8 and Neyshabur et al. (2019) would suggest. Tighter concentration bounds would immediately lead to improved margin bounds through our framework and would represent a major contribution to contemporary statistical learning theory.

5.A Additional Technical Lemmas

Lemma 5.3. *Let $X \in \mathscr{P}(\{+1,-1\})$ be a random variable with $\mathbb{E}[X] = x$, and*

$$h(x) := \mathrm{KL}\left(X, \mathrm{Uniform}(\{+1,-1\})\right) = \frac{1}{2}\left[(1+x)\log(1+x) + (1-x)\log(1-x)\right]$$

the KL divergence from a uniform prior. Then

$$h(x) \leq x^2 \log 2.$$

Proof of Lemma 5.3. The first equation is simply an explicit statement of the KL divergence. It is easy to see from convexity that $h(x) \leq x^2$; the improved (and optimal) constant of $\log 2$ requires a more complex argument, as follows.

Calculation gives the Maclaurin series

$$(1+x)\log(1+x) + (1-x)\log(1-x) = x^2 + \sum_{n=2}^{\infty} \frac{x^{2n}}{n(2n-1)}$$

which has a radius of convergence of 1. Therefore

$$h(x)/x^2 = \frac{1}{2} + \frac{1}{2}\sum_{n=2}^{\infty} \frac{x^{2n}}{(n+1)(2n+1)}$$

which is an increasing function on $(0,1)$ with supremum $\log 2$. From the definition, $x \in [-1,1]$. A similar argument applies for $(-1,0)$ and equality is achieved at $x = 0$, so the inequality holds (and is the tightest possible). □

Lemma 5.4. *For all $\epsilon \in [0, \frac{1}{2}], p \in [0,1], p > \epsilon$ (with the final condition ensuring the left hand side is well-defined),*

$$\mathrm{kl}(\epsilon : p - \epsilon) \geq p + 4\epsilon \log \epsilon.$$

Proof. Firstly, by Theorem 3.4

$$\sup_{C>0} \left[C\Phi_C(p-\epsilon) - C\epsilon\right] = \mathrm{kl}(\epsilon : p - \epsilon).$$

Note that $-\log(p-\epsilon) \geq 0$ if $p \leq 1$ and thus $\epsilon \log \frac{\epsilon}{R-\epsilon} \geq \epsilon \log \epsilon$. Using the bound $\log x \leq x - 1$ we also have that $(1-\epsilon)\log \frac{1-\epsilon}{1+\epsilon-R} \geq R - 2\epsilon$. Combining these results, $\mathrm{kl}(\epsilon : p-\epsilon) \geq p + \epsilon(\log \epsilon - 2)$; combination with the bound $\epsilon(\log \epsilon - 2) \geq 4\epsilon \log \epsilon$ in the specified range to completes the proof. □

Lemma 5.5. *Let $\tilde{\mu}$ be a probability distribution supported only on A, then for any other probability distribution ν*

$$\mathrm{KL}\left(\tilde{\mu}, \nu\right) \geq -\log \nu(A).$$

Proof. For the case $\nu(A) = 0$ or where $\tilde{\mu}$ is not absolutely continuous w.r.t. ν the above holds trivially as the KL is infinite.

Thus assume $\nu(A) > 0$ and $\tilde{\mu} \ll \nu$. Define the restriction (or conditional distribution) of ν to A as

$$\tilde{\nu} = \begin{cases} \nu/\nu(A) & \text{on A} \\ 0 & \text{else.} \end{cases}$$

Given the above assumptions, we have

$$\frac{d\tilde{\mu}}{d\nu} = \frac{d\tilde{\mu}}{d\tilde{\nu}} \frac{1}{\nu(A)}$$

so from the definition of and non-negativity of KL divergence,

$$\mathrm{KL}\left(\tilde{\mu}, \nu\right) = \log \frac{1}{\nu(A)} + \mathrm{KL}\left(\tilde{\mu}, \tilde{\nu}\right) \geq -\log \nu(A).$$

□

5.B Comparison of Theorem 5.5 to Existing Bounds

Here we discuss how our results in Theorem 5.5 compare to existing results. All of the following will be in the setting of this theorem with $R = 1$ (since all bounds only depend on R through the scaling R/γ).

5.B.1 Hard Margin Case

The best existing result for this case is the following, from Bartlett and Shawe-Taylor (1998, Theorem 4.17):

$$\mathcal{L}_0 \leq \frac{1}{m}\left(2k \log \frac{8em}{k} \log 32m + 2 \log \frac{8m}{\delta}\right) \in \mathcal{O}\left(\frac{1}{m}\left(\gamma_\star^{-2} \log^2 m + \log \frac{1}{\delta}\right)\right)$$

where $k = \lfloor 577 \gamma_*^{-2} \rfloor$.

From this we see that not only does Theorem 5.5 improve in order by removing a factor of $\log m$, but also considerably improves the constant factors.

Grønlund et al. (2020) show that there exists a dataset, and an estimator with $\|w\|_2 \leq 1$, such that:

$$\mathcal{L}_0 \geq \Omega\left(\frac{\log m}{m\gamma_\star^2}\right)$$

which is matched by our Theorem 5.5 and confirms it cannot be improved in order without additional assumptions.

5.B.2 Soft Margin

Several somewhat different results appear in this case. Using Rademacher complexity, Theorem 21 of Bartlett and Mendelson (2002) implies the following (where the big-O follows by combination with the trivial bound $\mathcal{L}_0 \leq 1$):

$$\mathcal{L}_0 \leq \widehat{\mathcal{L}}_\gamma + \frac{4}{\sqrt{m}\gamma} + \left(\frac{8}{\gamma} + 2\right)\sqrt{\frac{\log \frac{4}{\delta}}{m}} = \widehat{\mathcal{L}}_\gamma + \mathcal{O}\left(\sqrt{\frac{\gamma^{-2} + \log(1/\delta)}{m}}\right).$$

Based on a more complex bound in Langford and Shawe-Taylor (2003), McAllester (2003) gives the bound (for $m \geq 4$):

$$\mathcal{L}_0 \leq \widehat{\mathcal{L}}_\gamma + \frac{8}{m\gamma^2} + 2\sqrt{2\left(\frac{\widehat{\mathcal{L}}_\gamma}{m\gamma^2} + \frac{4}{m^2\gamma^4}\right)\log\frac{m\gamma^2}{4}} + \mathcal{O}\left(\sqrt{\frac{\log m + \log(1/\delta)}{m}}\right)$$
$$= \widehat{\mathcal{L}}_\gamma + \mathcal{O}\left(\frac{\log m}{m\gamma^2} + \sqrt{\frac{\log m}{m\gamma^2}\widehat{\mathcal{L}}_\gamma} + \sqrt{\frac{\log m + \log(1/\delta)}{m}}\right).$$

The above also leads to a hard-margin formulation which is however weaker than the Bartlett and Shawe-Taylor (1998) bound for all but very tiny margins, as pointed out by Grønlund et al. (2020).

The state-of-the-art, nearly tight bound given by Grønlund et al. (2020) is the following, not given with any constants,

$$\mathcal{L}_0 \leq \widehat{\mathcal{L}}_\gamma + \mathcal{O}\left(\frac{\gamma^{-2}\log m + \log(1/\delta)}{m} + \sqrt{\frac{\gamma^{-2}\log m + \log(1/\delta)}{m}\cdot\widehat{\mathcal{L}}_\gamma}\right)$$

which is shown in the same paper to be nearly-tight, in the existential sense that there exist data distributions for which it cannot be improved.

This matches exactly the bound given in Theorem 5.5 in order, but we emphasise both the simplicity of our proof and that we give constants, making it actually evaluable in practice.

5.C Empirical Evaluation of Theorem 5.7

5.C.1 Experimental setup

All experiments were performed using the Tensorflow 2 library (Abadi et al., 2015) in Python, on a single workstation with a Nvidia RTX 2080 Ti GPU. Code for the results is licensed under an MIT license is available in the supplementary material for the published version of the paper (Biggs and Guedj, 2022a).

We train SHEL networks and a partially-aggregated variation thereof under different hyperparameter configurations. We use this to compare changes in the generalisation error (the difference between test and train misclassification errors) with the complexity term from Theorem 5.7 given by

$$\frac{\sqrt{d_{\text{hid}}}}{\gamma\sqrt{m}}\left(V_\infty\|U - U^0\|_F + \|V\|_F\right). \tag{5.5}$$

Following previous empirical evaluations of such complexity terms, we train to a fixed value of cross-entropy; see Jiang et al. (2020b) for further discussion. The margin γ is set as that giving a fixed $\mathcal{L}_\gamma(f_{U,V}) = 0.2$, or $E_{f\sim\widetilde{Q}}\mathcal{L}_\gamma(f_{U,V}) = 0.2$ for the partially aggregated version.

For the partially-aggregated version, we include a feature map of three additional dense ReLU layers with Gaussian weight matrices with independent components, means $\{W_i\}_{i=1}^3$

and variances of σ. Again using the initialisation as a prior, this adds a term of

$$\sqrt{\sum_{i=1}^{3} \|W_i - W_i^0\|_F^2 / 4m\sigma^2}$$

to the right hand side of the bound. To enable comparison, we set σ to make this term constant and equal to a half when calculating $E_{f\sim\widetilde{Q}}\mathcal{L}_\gamma(f)$. This is done during the evaluation phase, and training is performed on the non-stochastic version (weights as means) as in Dziugaite and Roy (2017).

These experiments aim to evaluate the predictive ability of this complexity measure under changes of procedures. To this end we provide plots of the generalisation, $G(\omega)$, and complexity measure, $C(\omega)$, for trained parameters ω versus some change in hyperparameter value.

We also provide evaluations using the sign-error, a measure of predictive power defined in Dziugaite et al. (2020) as

$$\frac{1}{2}\mathop{\mathbb{E}}_{\omega,\omega'}[1 - \text{sign}(C(\omega') - C(\omega)) \cdot \text{sign}(G(\omega') - G(\omega))]$$

where ω and ω' are parameters obtained through training with one changed hyperparameter between them. The maximum over such pairs of hyperparameter settings is a measure of the robustness of predictions made about the generalisation based on the complexity measure; if this value is low, the complexity measure makes robust predictions. We provide this maximum, the median, and the mean of the above (as in Jiang et al., 2020b) for different setups and allowing different hyperparameters to vary.

5.C.2 SHEL Network

On the MNIST (LeCun et al., 2010) dataset, we examine the following hyperparameter settings, finding through the sign error (Table 5.1) that predictions under changes of training size are quite robust, while those under changes of learning rate or width are poor. We additionally provide plots (Figs. 5.1 to 5.3) for some selected hyperparameter values to verify the above. This poor prediction under such changes is unfortunately a feature of many such complexity measures (Dziugaite et al., 2020).

- Learning rate $\in \{10^{-3}, 3\times10^{-3}, 10^{-2}, 3\times10^{-2}, 10^{-1}\}$.
- Train set sizes $\in \{60\,000, 30\,000, 15\,000, 7\,500\}$.
- Width $\in \{50, 100, 200, 400, 800\}$.
- Batch size $= 200$.
- Learning algorithm SGD with momentum parameter $= 0.9$.

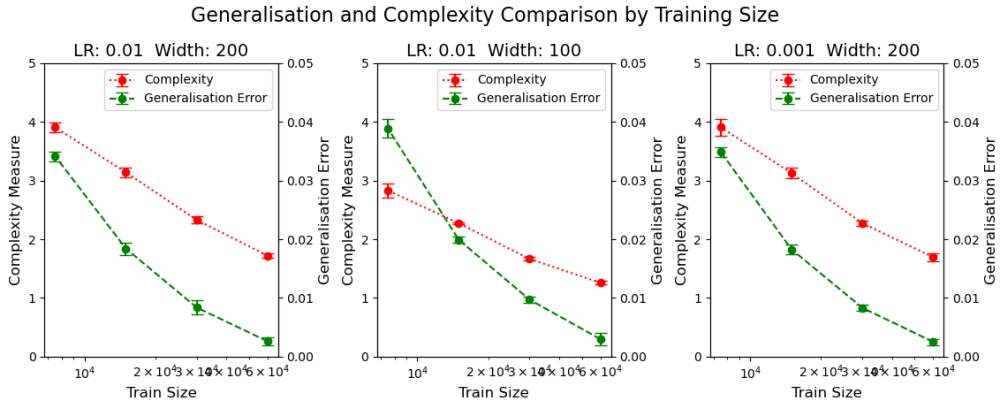

Figure 5.1: Changes in complexity measure and generalisation error versus training set size under fixed other hyperparameters, for a SHEL network trained on MNIST.

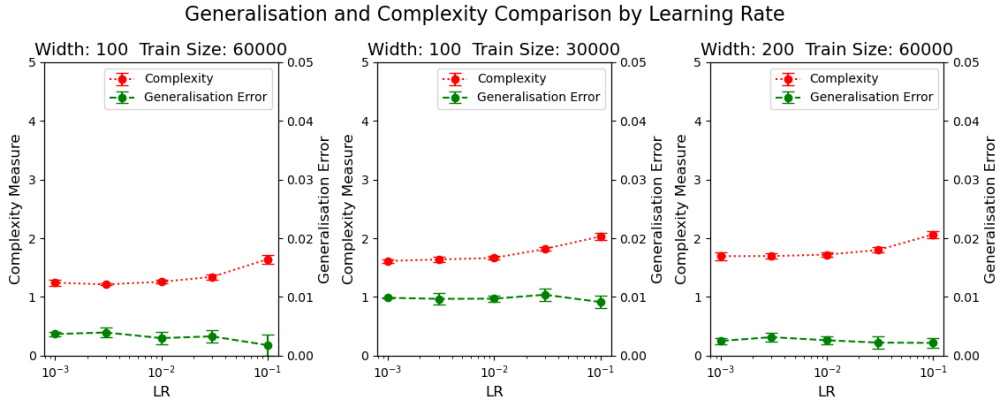

Figure 5.2: Changes in complexity measure and generalisation error versus learning rate under fixed other hyperparameters, for a SHEL network trained on MNIST.

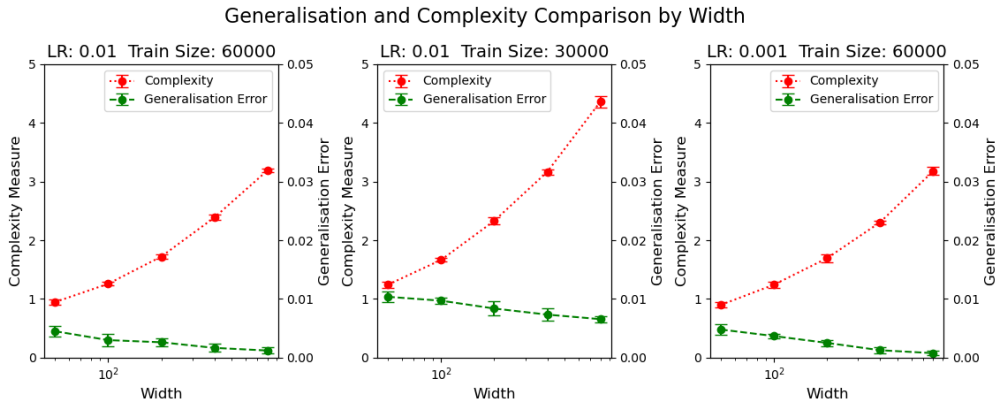

Figure 5.3: Changes in complexity measure and generalisation error versus width under fixed other hyperparameters, for a SHEL network trained on MNIST.

Variable Hyperparameter	Max SE	Median SE	Mean SE
Learning Rate	1.0	0.60	0.56
Width	1.0	1.0	0.90
Train Size	0.2	0.0	0.00
All	1.0	0.60	0.53

Table 5.1: Statistics of the sign error, SE, under different varying hyperparameters for a SHEL network trained on MNIST.

5.C.3 Partially-Derandomised SHEL

Again on the MNIST dataset, we evaluate the partially-derandomised version of the above under the same hyperparameter values, excluding learning rates of 0.1 and 0.03 which sometimes led to numerical instability. Figures 5.4 to 5.6 provide sample results and the sign-error results are reported in Table 5.2.

These sign error results show that predictions under changes of training size are completely robust, while those under changes of learning rate or width are still poor. The predictions for width are somewhat improved, though we note that our estimate of this quantity may be somewhat noisy as the generalisation error appears largely independent of width.

Variable Hyperparameter	Max SE	Median SE	Mean SE
Learning Rate	1.0	0.60	0.49
Width	1.0	0.40	0.46
Train Size	0.0	0.0	0.0
All	1.0	0.20	0.31

Table 5.2: Statistics of the sign error, SE, under different varying hyperparameters for a partially-derandomised SHEL network trained on MNIST.

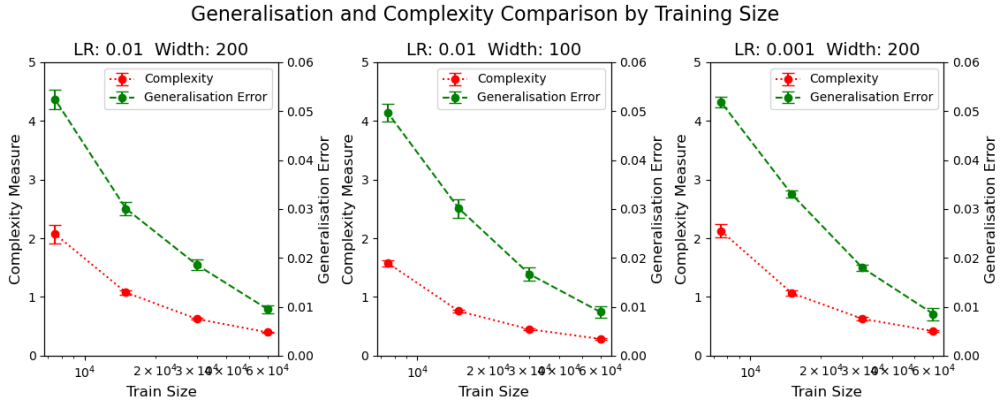

Figure 5.4: Changes in complexity measure and generalisation error versus training set size under fixed other hyperparameters, for a partially-derandomised SHEL network trained on MNIST.

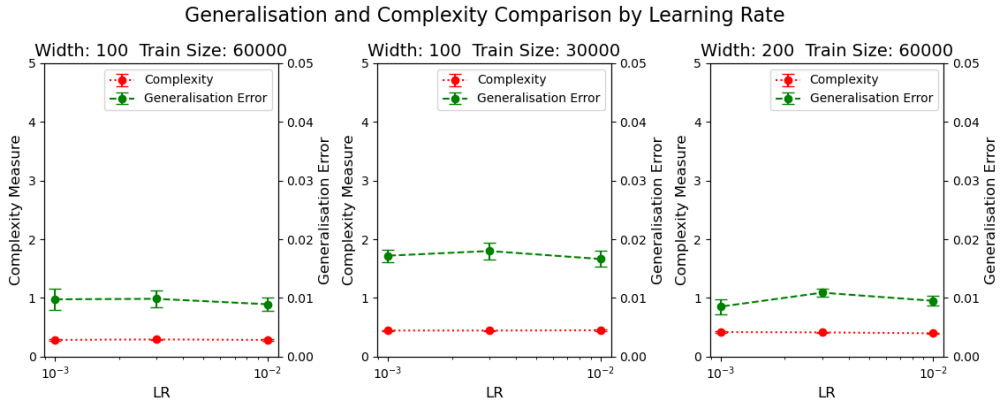

Figure 5.5: Changes in complexity measure and generalisation error versus learning rate under fixed other hyperparameters, for a partially-derandomised SHEL network trained on MNIST.

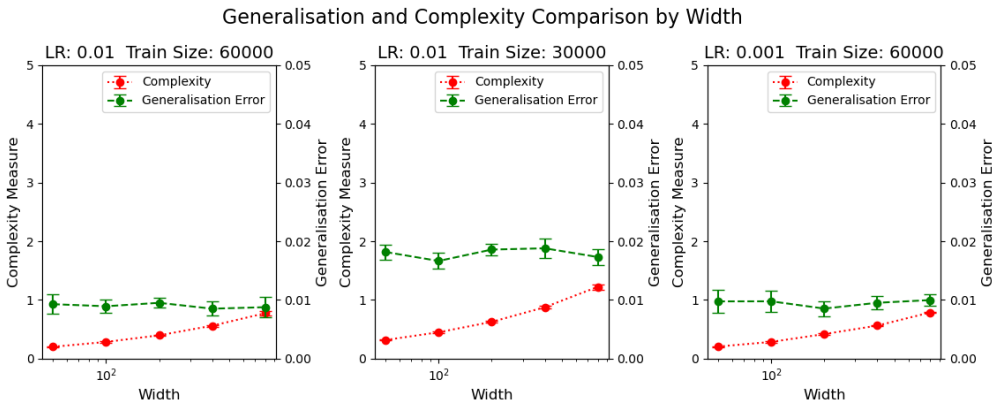

Figure 5.6: Changes in complexity measure and generalisation error versus width under fixed other hyperparameters, for a partially-derandomised SHEL network trained on MNIST.

Chapter 6

Non-Vacuous Generalisation Bounds for Shallow Networks

We focus on a specific class of shallow neural networks with a single hidden layer, namely those with L_2-normalised data and either a sigmoid-shaped Gaussian error function ("erf") activation or a Gaussian Error Linear Unit (GELU) activation. For these networks, we derive new generalisation bounds through the PAC-Bayesian theory; unlike most existing such bounds they apply to neural networks with *deterministic* rather than randomised parameters. Our bounds are empirically non-vacuous when the network is trained with vanilla stochastic gradient descent on MNIST, Fashion-MNIST, and binary classification versions of the above.

6.1 Introduction

The study of generalisation properties of deep neural networks is arguably one of the topics gaining most traction in deep learning theory (see, *e.g.*, the recent surveys Jiang et al., 2020b; Kawaguchi et al., 2020). In particular, a characterisation of out-of-sample generalisation is essential to understand where trained neural networks are likely to succeed or to fail, as evidenced by the recent NeurIPS 2020 competition "Predicting Generalization in Deep Learning" (Jiang et al., 2020a). One stream of this joint effort, which the present paper contributes to, is dedicated to the study of shallow neural networks, potentially paving the way to insights on deeper architectures.

Despite numerous efforts in the past few years, non-vacuous generalisation bounds for *deterministic* neural networks with many more parameters than data remain generally elusive. Those few non-vacuous bounds that exist primarily report bounds for networks with randomised parameters, for example Gaussian weights, which are re-drawn for every prediction (a non-exhaustive list of references would begin with Dziugaite and Roy, 2017, 2018; Hellström and Durisi, 2021; Neyshabur et al., 2017, 2018), or for compressed versions

of the trained networks (Zhou et al., 2019). While these undoubtedly advanced knowledge on generalisation in deep learning theory, this is far from contemporary practice which generally focuses on deterministic networks obtained directly through stochastic gradient descent (SGD), as we do.

The PAC-Bayesian theory is thus far the only framework within which non-vacuous bounds have been provided for networks trained on common classification tasks. Given its focus on randomised or "Gibbs" predictors, the aforementioned lack of results for deterministic networks is unsurprising. However, the framework is not limited to such results: one area within PAC-Bayes where deterministic predictors are often considered lies in a range of results for the "majority vote", or the expected overall prediction of randomised predictors, which is itself deterministic.

Computing the average output of deep neural networks with randomised parameters is generally intractable: therefore most such works have focused on cases where the average output is simple to compute, as for example when considering linear predictors. Here, building on ideas from Biggs and Guedj (2022a), we show that provided our predictor structure factorises in a particular way, more complex majority votes can be constructed. In particular, we give formulations for randomised predictors whose majority vote can be expressed as a deterministic single-hidden-layer neural network. Through this, we obtain classification bounds for these *deterministic* predictors that are non-vacuous on the celebrated baselines MNIST (LeCun et al., 2010), Fashion-MNIST (Xiao et al., 2017), and binarised versions of the above. We believe these are the first such results.

Our work fundamentally relates to the question: what kind of properties or structures in a trained network indicate likely generalisation to unseen data? It has been shown by Zhang et al. (2017) that neural networks trained by SGD can perfectly overfit large datasets with randomised labels, which would indicate a lack of capacity control, while simultaneously generalising well in a variety of scenarios. Thus, clearly any certification of generalisation must involve extracting additional information other than the train loss—for example, the specific final network chosen by SGD. How do the final parameters of a neural network trained on an "easy" data distribution as opposed to a pathological (*e.g.*, randomised label) one differ? A common answer to this has involved the return of capacity control and the norms of the weight matrices, often measured as a distance to the initialisation (as done, *e.g.*, in Bartlett et al., 2017; Dziugaite and Roy, 2017; Neyshabur et al., 2018).

We suggest, following insights from Dziugaite et al. (2021), that a better answer lies in utilising the empirically-observed stability of SGD on easy datasets. We give bounds that are tightest when a secondary run of SGD on some subset of the training set gives final weights that are close to the full-dataset derived weights. This idea combines naturally in the PAC-Bayes framework with the requirement of perturbation-robustness of the weights—related to

the idea of flat-minima (Hinton and van Camp, 1993; Hochreiter and Schmidhuber, 1997)—to normalise the distances between the two runs. By leveraging this commonly-observed empirical form of stability we effectively incorporate information about the inherent easiness of the dataset and how adapted our neural network architecture is to it. Although it is a deep and interesting theoretical question as to when and why such stability occurs under SGD, we believe that by making the link to generalisation explicit we solve some of the puzzle.

Setting. We consider d_{out}-class classification on a set $\mathcal{X} \subset \mathbb{R}^{d_{\text{in}}}$ with "score-output" predictors returning values in $\hat{\mathcal{Y}} \subset \mathbb{R}^{d_{\text{out}}}$ with multi-class label space $\mathcal{Y} = [d_{\text{out}}]$, or in $\hat{\mathcal{Y}} = \mathbb{R}$ with binary label space $\mathcal{Y} = \{+1, -1\}$. We recall that the prediction is the argmaximum or sign of the output and the misclassification loss is defined as $\ell_0(f, (\boldsymbol{x}, y)) = \mathbf{1}\{\text{argmax}_{k \in [d_{\text{out}}]} f(\boldsymbol{x})[k] \neq y\}$ or $\ell_0(f, (\boldsymbol{x}, y)) = \mathbf{1}\{y f(\boldsymbol{x}) \leq 0\}$ respectively. It is will prove useful that scaling does not enter into these losses and thus the outputs of classifiers can be arbitrarily re-scaled by $c > 0$ without affecting the predictions. We write $\mathcal{L}_0(f) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell_0(f, (\boldsymbol{x}, y))$ and $\widehat{\mathcal{L}}_0(f) := m^{-1} \sum_{(\boldsymbol{x},y)\in S} \ell_0(f, (\boldsymbol{x}, y))$ for the risk and empirical risk of the predictors with respect to data distribution $\mathcal{D}$ and i.i.d. m-sized sample $S \sim \mathcal{D}^m$.

Overview of our contributions. We derive generalisation bounds for a single-hidden-layer neural network $F_{U,V}$ with first and second layer weights U and V respectively taking the form

$$F_{U,V}(\boldsymbol{x}) = V \phi\left(\beta \frac{U\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\right)$$

with ϕ being an element-wise activation. If the data is normalised to have $\|\boldsymbol{x}\|_2 = \beta$ these are simply equivalent to one-hidden-layer neural networks with activation ϕ and the given data norm. We provide high-probability bounds on $\mathcal{L}_0(F_{U,V})$ of the approximate form

$$2 \mathop{\mathbb{E}}_{f\sim Q} \widehat{\mathcal{L}}_0(f) + \mathcal{O}\left(\frac{\beta\|U - U^n\|_F + \|V - V^n\|_F}{\sqrt{m-n}}\right),$$

where Q is a distribution over predictors f, which depends on U and V but does not necessarily take the form of a neural network. The construction of this randomised proxy Q is central to our PAC-Bayes derived proof methods. The bounds hold uniformly over any choice of weight matrices, but for many choices the bounds obtained will be vacuous; what is interesting is that they are non-vacuous for SGD-derived solutions on some real-world datasets. U^n and V^n are matrices constructed using some subset $n < m$ of the data. Since we consider SGD-derived weights, we can leverage the empirical stability of this training method (through an idea introduced by Dziugaite et al., 2021) to construct U^n, V^n which are quite close to the final true SGD-derived weights U, V, essentially by training a prior on the n-sized subset in the same way.

Outline. In Section 6.2 we give an overview of results from previous works which we use. In Section 6.3 we give a bound on the generalisation error of binary classification SHEL networks, which are single hidden layer networks with "erf" activations. In Section 6.4 we extend to multi-class classification using a simple assumption, giving a general formulation as well as results for "erf"- and GELU-activated networks. In Section 6.5 we discuss our experimental setting and give our numerical results, which we discuss along with future work in Section 6.6.

6.2 Background and Related Work

We use Theorem 3.12 (Maurer's inverse-kl bound) as our foundational PAC-Bayes bound. We again note that although the bound holds over all "posterior" distributions Q, a poor choice (for example, one over-concentrated on a single predictor) will lead to a vacuous bound. We will use the relaxation $\mathrm{kl}^{-1}(t,c) \leq t + \sqrt{c/2}$ which gives an idea of the behaviour of Theorem 3.12; however in the case of t close to 0 the original formulation is considerably tighter.

Data-Dependent Priors. A careful choice of the prior is essential to the production of sharp PAC-Bayesian results. A variety of works going back to Ambroladze et al. (2006) and Parrado-Hernández et al. (2012) (and further developed by Dziugaite and Roy, 2018; Dziugaite et al., 2021; Pérez-Ortiz et al., 2021a,c; Rivasplata et al., 2018, among others) have considered dividing the training sample into two parts, one to learn the prior and another to evaluate the bound. Formally, we divide $S = S^{\mathrm{prior}} \cup S^{\mathrm{bnd}}$ and use S^{prior} to learn a prior P^n where $n = |S^{\mathrm{prior}}|$, then apply the PAC-Bayesian bound using sample S^{bnd} to a posterior Q learned on the entirety of S. The resulting bound replaces $\widehat{\mathcal{L}}_0$ by $\widehat{\mathcal{L}}_0^{\mathrm{bnd}}$, P by the data-dependent P^n, and m by $m - n = |S^{\mathrm{bnd}}|$; thus the KL complexity term may be reduced at the cost of a smaller dataset to apply the bound to.

Dziugaite et al. (2021) used this when considering training neural networks by constructing a so-called "coupled" prior P^n which is trained in the same way from the same initialisation as the posterior Q by stochastic gradient descent with the first n examples from the training set forming one epoch. Due to the stability of gradient descent, the weights of P^n and Q evolve along similar trajectories; thus stability of the training algorithm is leveraged to tighten bounds without explicit stability results being required (and we do not study the conditions under which SGD provides such solutions). In many ways this can be seen as an extension of previous work such as Dziugaite and Roy (2017) relating generalisation to the distance from initialisation rather than total weight norms.

Majority Votes. Since PAC-Bayesian bounds generally consider the risk of randomised predictors, a natural question is whether prediction accuracy can be improved by "voting" many independently drawn predictions; such a majority vote predictor takes the deterministic form $\mathrm{MV}_Q(\boldsymbol{x}) := \operatorname{argmax}_k \mathbb{P}_{f\sim Q}(\operatorname{argmax} f(\boldsymbol{x}) = k)$. Several strategies have been devised to obtain bounds for these predictors via PAC-Bayesian theorems, with the simplest (and often most successful) being the unattributed first-order bound $\ell_0(\mathrm{MV}_Q, (\boldsymbol{x}, y)) \leq 2\,\mathbb{E}_{f\sim Q}\,\ell_0(f, (\boldsymbol{x}, y))$ valid for all $(\boldsymbol{x}, y)$, called the "folk theorem" by Langford and Shawe-Taylor (2003) and the *first-order* bound elsewhere. This can be substituted directly into PAC-Bayesian theorems such as Theorem 3.12 above to obtain bounds for the majority vote at a de-randomisation cost of a factor of two. This is the result we use, since across a variety of preliminary experiments we found other strategies including the tandem bound of Masegosa et al. (2020)[1] and the C-bound of Lacasse et al. (2006) were uniformly worse, as also discussed by Zantedeschi et al. (2021).

Gaussian Sign Aggregation. To exploit the useful relationship above, Germain et al. (2009) considered aggregating a kind of linear prediction function of the form $f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w} \cdot \boldsymbol{x})$ with $\boldsymbol{w} \sim Q = \mathcal{N}(\boldsymbol{u}, I)$. In this case the aggregation can be stated in closed form using the Gaussian error function "erf" as

$$\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{u},I)} \operatorname{sign}(\boldsymbol{w}\cdot\boldsymbol{x}) = \operatorname{erf}\left(\frac{\boldsymbol{u}\cdot\boldsymbol{x}}{\sqrt{2}\|\boldsymbol{x}\|_2}\right). \tag{6.1}$$

This closed-form relationship has been used since by Letarte et al. (2019) and Biggs and Guedj (2021) in a PAC-Bayesian context for neural networks with sign activation functions and Gaussian weights; Biggs and Guedj (2022a) used it to derive a generalisation bound for SHEL (single hidden erf layer) networks, which have a single hidden layer with erf activation function. We will consider deriving a different PAC-Bayesian bound for this same situation and develop this method further in this work.

Other Approaches. A wide variety of other works have derived generalisation bounds for deterministic neural networks without randomisation. We note in particular the important works of Bartlett et al. (2017), Neyshabur et al. (2017) (using PAC-Bayesian ideas in their proofs) and Arora et al. (2018), but contrary to us, they do not provide empirically non-vacuous bounds. Nagarajan and Kolter (2019) de-randomise PAC-Bayesian bounds by leveraging the notion of noise-resilience (how much the training loss of the network changes with noise injected into the parameters), but they note that in practice their bound would

[1] Masegosa et al. (2020) observe that these alternative bounds are often prefereable to the first order bound as optimisation objectives, but in this work we are not considering bounds as objectives at all, rendering this a moot point here.

be numerically large. Many of these approaches utilise uniform convergence, which may lead to shortcomings as discussed at length by Nagarajan and Kolter (2019); we emphasise that the bounds we give are non-uniform and avoid these shortcomings. Finally, we also highlight the works of Neyshabur et al. (2015, 2019) which specifically consider single-hidden-layer networks as we do – as in the recent study from Tinsi and Dalalyan (2021). Overall we emphasise that, to the best of our knowledge, all existing bounds for deterministic networks are vacuous when networks are trained on real-world data.

6.3 Binary SHEL Network

We begin by giving a bound for binary classification by a single hidden layer neural network with error function ("erf") activation. Binary classification takes $\mathcal{Y} = \{+1, -1\}$, with prediction the sign of the prediction function. The specific network takes the following form with output dimension $d_{\text{out}} = 1$. Although the erf activation function is not a commonly-used one, it is very close in value to the more common tanh activation. It can also be rescaled to a Gaussian CDF activation, which is again very close to the classical sigmoid activation (and is itself the CDF of the probit distribution).

Definition. SHEL Network. *(Biggs and Guedj, 2022a) For $U \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}$, $V \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{out}}}$, and $\beta > 0$, a β-normalised single hidden erf layer (SHEL) network is defined by*

$$F^{\text{erf}}_{U,V}(\boldsymbol{x}) := V \cdot \text{erf}\left(\beta \frac{U\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\right).$$

The above is a single-hidden-layer network with a first normalisation layer, or if the data is already normalised the overall scaling $\|\boldsymbol{x}\|_2$ can be absorbed into the β parameter. This parameter β could easily be absorbed into the matrix U and mainly has the effect of scaling the relative learning rate for U versus V when training by gradient descent, as shown by looking at $\frac{\partial}{\partial U} F^{\text{erf}}_{U,V}(\boldsymbol{x})$, something which would normally be affected by the scaling of data. A higher β means more "feature learning" takes place as U has a relatively larger learning rate.

For binary classification, the majority vote of distribution Q is $\text{MV}_Q(\boldsymbol{x}) = \text{sign}(\mathbb{E}_{f\sim Q} \text{sign}(f(\boldsymbol{x})))$. By expressing the (binary classification) SHEL network directly as the majority vote of a randomised prediction function, we can prove a PAC-Bayesian generalisation bound on its error using the first-order bound. The misclassification error of the randomised function can further be stated in closed form using the Binomial cumulative distribution function (CDF), giving rise to a bound where the distribution Q does not appear directly.[2]

[2]We note that the bound with a Binomial expansion used in the binary case is closely related to the very generic "binomial bound" used earlier by Lacasse et al. (2010).

Theorem 6.1. *In the binary setting, fix network prior parameters* $\boldsymbol{u}_1^0, \ldots, \boldsymbol{u}_{d_{\text{hid}}}^0 \in \mathbb{R}^{d_{\text{in}}}, \boldsymbol{v}^0 \in \mathbb{R}^{d_{\text{hid}}}$, $\beta > 0$, *bound tuning parameter* $T \in \mathbb{N}^+$, *and data distribution* $\mathcal{D}$. *For* $\delta \in (0,1)$, *with probability at least* $1-\delta$ *under the sample* $S \sim \mathcal{D}^m$, *simultaneously for any* $U \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{out}}}, \boldsymbol{v} \in \mathbb{R}^{d_{\text{hid}}}$,

$$\mathcal{L}_0(F^{\text{erf}}_{U,\boldsymbol{v}}) \leq 2\,\text{kl}^{-1}\left(\frac{1}{m}\sum_{(\boldsymbol{x},y)\in S} \text{Bin}\left(\frac{T}{2}; T, \frac{1}{2}\left(1 + \frac{yF_{U,\boldsymbol{v}}(\boldsymbol{x})}{\|\boldsymbol{v}\|_1}\right)\right), \frac{T\kappa + \log\frac{2\sqrt{m}}{\delta}}{m}\right).$$

Here $F^{\text{erf}}_{U,\boldsymbol{v}}$ *is a SHEL network with* β*-normalised activation,*

$$\kappa := \sum_{k=1}^{d_{\text{hid}}} \frac{|\boldsymbol{v}_k|}{\|\boldsymbol{v}\|_1}\left(\beta^2\|\boldsymbol{u}_k - \boldsymbol{u}_k^0\|_2^2 + \log\left(2\frac{|\boldsymbol{v}_k|/\|\boldsymbol{v}\|_1}{|\boldsymbol{v}_k^0|/\|\boldsymbol{v}^0\|_1}\right)\right)$$

and $\text{Bin}(k; r, p)$ *is the CDF of a Binomial distribution with parameters* r, p.

6.4 Multi-class Networks

We now go further and show that various single-hidden-layer multi-class neural networks can also be expressed as the expectation of randomised predictors. We show specific results for multi-class SHEL networks as well as GELU-activation (Hendrycks and Gimpel, 2016) networks as defined below. We also give a more general form of the result as a aggregation of individual aggregated predictors which allows these results to be extended further.

We make a simple assumption based on the first-order bound to extend PAC-Bayesian bounds to this case. This is necessary because under certain choices of PAC-Bayes posterior Q, the *majority vote* does not give the same prediction as the *expected vote* as was the case in Section 6.3, i.e. there exist Q such that $\text{argmax}_k \mathbb{E}_{f\sim Q} f(\boldsymbol{x})[k] \neq \text{MV}_Q(\boldsymbol{x})$ at certain adversary-chosen values of $\boldsymbol{x}$. Thus we assume that $\mathcal{L}_0(\mathbb{E}_{f\sim Q} f(\boldsymbol{x})) \leq 2\,\mathbb{E}_{f\sim Q}\mathcal{L}_0(f)$, (denoted $\star$), which follows from the first order bound in the case $\mathbb{E}_Q f(\boldsymbol{x}) \approx \text{MV}_Q(\boldsymbol{x})$, which we later verify empirically.[3]

6.4.1 SHEL Networks

Here we give a generalisation bound for a multi-class variant of the SHEL network using the above assumption. The proof is slightly different from the binary case, but still relies on the useful fact that the SHEL network can be written as the expectation of a randomised predictor. This predictor however takes a slightly different form to that in the binary case.

Theorem 6.2. *In the multi-class setting, fix prior parameters* $U^0 \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}$ *and* $V^0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$, $\sigma_V > 0$, $\beta > 0$, *and data distribution* $\mathcal{D}$. *For* $\delta \in (0,1)$, *with probability at least*

[3] The following is a counterexample to this assumption in 3-class classification: suppose that at a fixed $\boldsymbol{x}$, $f(\boldsymbol{x}) = [1,0,0]^T$ with probability 3/4 under $f \sim Q$, and $f(\boldsymbol{x}) = [0,4,0]^T$ otherwise. The majority vote predicts the first class, while the expectation $\mathbb{E}_Q f(\boldsymbol{x})$ predicts the second class. However for the majority of $\boldsymbol{x}$ the assumption holds.

$1-\delta$ under the sample $S \sim \mathcal{D}^m$, simultaneously for any $U \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}, V \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$ such that assumption ($\star$) is satisfied,

$$\mathcal{L}_0(F^{\text{erf}}_{U,V}) \leq 2\,\text{kl}^{-1}\left(\mathop{\mathbb{E}}_{f\sim Q}\widehat{\mathcal{L}}_0(f),\ \frac{\kappa + \log\frac{2\sqrt{m}}{\delta}}{m}\right).$$

Here $F^{\text{erf}}_{U,V}$ is a SHEL network with β-normalised activation,

$$\kappa := \beta^2\|U-U^0\|_F^2 + \frac{\|V-V^0\|_F^2}{2\sigma_V^2},$$

and

$$\mathop{\mathbb{E}}_{f\sim Q}\widehat{\mathcal{L}}_0(f) := \frac{1}{m}\sum_{(\boldsymbol{x},y)\in S}\mathop{\mathbb{P}}_{W_1,W_2}(\text{argmax}\,[W_2\,\text{sign}(W_1\boldsymbol{x})] \neq y),$$

with the probability $\mathbb{P}_{W_1,W_2}$ over draws of $\text{vec}(W_2) \sim \mathcal{N}(\text{vec}(V), \sigma_V^2 I), \text{vec}(W_1) \sim \mathcal{N}(\text{vec}(U), \frac{1}{2}\beta^{-2}I)$. Note that vec *is the vectorisation operator and* sign *is applied element-wise.*

Differences to Biggs and Guedj (2022a). In their Theorem 5, Biggs and Guedj (2022a) give a bound for generalisation in SHEL networks, with $\mathcal{L}_0(F^{\text{erf}}_{U,V})$ upper bounded under similar conditions to Theorem 6.2 by

$$\widehat{\mathcal{L}}_\gamma(F^{\text{erf}}_{U,V}) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{d_{\text{hid}}}}{\gamma\sqrt{m}}\left(V_\infty\|U-U^0\|_F + \|V\|_F\right)\right),$$

where $\widehat{\mathcal{L}}_\gamma(g) = m^{-1}|\{(\boldsymbol{x},y)\in S : g(\boldsymbol{x})[y] - \max_{k\neq y} g(\boldsymbol{x})[k] \leq \gamma\}|$, the proportion of γ-margin errors in the training set, and $V_\infty := \max_{ij}|V_{ij}|$. Thus a margin loss of the actual predictor used rather than a stochastic one appears. A tighter formulation more similar to Theorem 6.2 is also given in an appendix and the bound could be similarly adapted to a data-dependent prior.

The derivation of the bound is quite different from ours, relying on a quite differently-constructed randomised version of Q (which is however constructed to have mean $F^{\text{erf}}_{U,V}$), and a de-randomisation procedure relying on margins and concentration rather than a majority vote bound. Both the form of Q used and the de-randomisation step lead to issues which we have addressed through our alternative formulation of Q and a majority vote bound: de-randomisation requires a very low variance Q, leading to the $\sqrt{d_{\text{hid}}}/\gamma$ term in the bound, which is empirically very large for low margin losses. Thus as demonstrated in their experiments, the big-O term increases with widening networks. Finally we note the most important distinction to our work: contrary to the present work, Biggs and Guedj (2022a) do not obtain non-vacuous bounds in practice.

6.4.2 GELU Networks

The Gaussian Error Linear Unit is a commonly-used alternative to the ReLU activation defined by $\text{GELU}(t) := \Phi(t)t$ where $\Phi(t)$ is the standard normal CDF. Far from the origin,

the $\Phi(t)$ is saturated at zero or one so it looks much like a smoothed ReLU or SWISH activation (defined by Ramachandran et al., 2018 as $t/(1+e^{-ct})$ for some $c>0$). It was introduced to lend a more probabilistic interpretation to activation functions, and fold in ideas of regularisation by effectively averaging the output of adaptive dropout (Ba and Frey, 2013); its wide use reflects excellent empirical results in a wide variety of settings.

Definition. GELU Network. *For $U \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}$, $V \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{out}}}$, and $\beta > 0$, a β-normalised single hidden layer GELU network is defined by*

$$F_{U,V}^{\text{GELU}}(\boldsymbol{x}) := V \cdot \text{GELU}\left(\beta \frac{U\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\right)$$

where $\text{GELU}(t) := \Phi(t)\, t$.

Theorem 6.3. *In the multi-class setting, fix prior parameters $U^0 \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}$ and $V^0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$, $\sigma_V > 0$, $\sigma_U > 0$ $\beta > 0$, and data distribution $\mathcal{D}$. For $\delta \in (0,1)$, with probability at least $1-\delta$ under the sample $S \sim \mathcal{D}^m$, simultaneously for any $U \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}, V \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$ such that assumption ($\star$) is satisfied,*

$$\mathcal{L}_0(F_{U,V}^{\text{GELU}}) \leq 2\,\text{kl}^{-1}\left(\mathop{\mathbb{E}}_{f\sim Q} \widehat{\mathcal{L}}_0(f),\ \frac{\kappa + \log \frac{2\sqrt{m}}{\delta}}{m}\right). \tag{6.2}$$

Here $F_{U,V}^{\text{GELU}}$ is a single-hidden-layer GELU network with β-normalised activation,

$$\kappa := \left(\beta^2 + \frac{1}{\sigma_U^2}\right)\frac{\|U-U^0\|_F^2}{2} + \frac{\|V-V^0\|_F^2}{2\sigma_V^2},$$

and $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0(f)$ is

$$\frac{1}{m}\sum_{(\boldsymbol{x},y)\in S} \mathop{\mathbb{P}}_{W_1,W_2}\left(\text{argmax}\left[W_2(\mathbf{1}_{W_1\boldsymbol{x}} \otimes (W_1'\boldsymbol{x}))\right] \neq y\right),$$

with the probability $\mathbb{P}_{W_1,W_2}$ is over draws of $\text{vec}(W_2) \sim \mathcal{N}(\text{vec}(V), \sigma_V^2 I), \text{vec}(W_1), \sim \mathcal{N}(\text{vec}(U), \beta^{-2} I)$ and $\text{vec}(W_1') \sim \mathcal{N}(\text{vec}(V), \sigma_U^2 I)$. Here vec is the vectorisation operator and the indicator function $\mathbf{1}_y$ is applied element-wise.

Although the proof method for Theorem 6.3 and the considerations around the hyper-parameter β are the same as for Theorem 6.2 and SHEL networks, one notable difference is the inclusion of the σ_U parameter. When this is very small, the stochastic predictions are effectively just a linear two-layer network with adaptive dropout providing the non-linearity. The ability to adjust the variability of the stochastic network hidden layer and thus $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0(f)$ is a major advantage over the SHEL network; in SHEL networks this variability can only be changed through β, which is a fixed parameter related to the deterministic network, not just a quantity appearing only in the bound.

6.4.3 General Form

Both of the above bounds can effectively be derived from the same formulation, as both take the form

$$F(\boldsymbol{x}) := \mathop{\mathbb{E}}_{f\sim Q} f(\boldsymbol{x}) = \sum_{k=1}^{d_{\text{hid}}} v_k H_k(\boldsymbol{x}), \tag{6.3}$$

where $\boldsymbol{v}_k \in \mathbb{R}^{d_{\text{out}}}$ are the column vectors of a matrix $V \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$ and $H_k : \mathcal{X} \to \mathbb{R}$ is itself a predictor of a form expressible as the expectation of another predictor. This means that there exists a distribution on functions $Q^k \in \mathscr{P}(\mathcal{F})$ such that for each $\boldsymbol{x} \in \mathcal{X}$, $H_k(\boldsymbol{x}) = \mathbb{E}_{h\sim Q^k}[h(\boldsymbol{x})]$. The bound on the generalisation of such predictors takes essentially the same form those given in the rest of this section.

Theorem 6.4. *Fix a set of priors $P^k \in \mathscr{P}(\mathcal{F})$ with $k \in [d_{\text{hid}}]$, a prior weight matrix $V^0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$, $\sigma_V > 0$, $\delta \in (0,1)$. With probability at least $1-\delta$ under the sample $S \sim \mathcal{D}^m$ simultaneously for any $V \in \mathbb{R}^{d_{\text{out}} \times d_{\text{hid}}}$ and set of $Q^k \in \mathscr{P}(\mathcal{F})$ such that assumption ($\star$) holds,*

$$\mathcal{L}_0(F) \leq 2\,\text{kl}^{-1}\left(\mathop{\mathbb{E}}_{f\sim Q} \widehat{\mathcal{L}}_0(f),\ \frac{\kappa + \log\frac{2\sqrt{m}}{\delta}}{m}\right) \tag{6.4}$$

where F is the deterministic predictor given in Eq. (6.3),

$$\kappa := \sum_{k=1}^{d_{\text{hid}}} \text{KL}\left(Q^k, P^k\right) + \frac{\|V - V^0\|_F^2}{2\sigma_V^2},$$

and

$$\mathop{\mathbb{E}}_{f\sim Q} \widehat{\mathcal{L}}_0(f) := \frac{1}{m} \sum_{(\boldsymbol{x},y)\in S} \mathop{\mathbb{P}}_{\boldsymbol{w}^\cdot, h^\cdot}\left(\text{argmax}\left[\sum_{k=1}^{d_{\text{hid}}} \boldsymbol{w}^k h^k(\boldsymbol{x})\right] \neq y\right)$$

is the stochastic predictor sample error where the probability is over independent draws of $\boldsymbol{w}^k \sim \mathcal{N}(\boldsymbol{v}_k, \sigma_V^2 I), h^k \sim Q^k$ for all $k \in [d_{\text{hid}}]$.

6.5 Numerical Experiments

For numerical evaluation and the tightest possible values of bounds, a few further ingredients are needed, which are here described. We also give the specific way these are evaluated in our later experiments.

Bounding the empirical error term. We note that there is rarely a closed form expression for $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0(f)$, as there is in the binary SHEL bound. In the multi-class bounds, this term must be estimated and bounded by making many independent draws of the parameters and using the fact that the quantity is bounded in $[0,1]$ to provide a concentration bound through, for example, Hoeffding's inequality. This adds a penalty to the bound which reduces with the number of independent draws and thus the amount of computing time invested in calculating the bound, but this is not a theoretical drawback of the bound. We give here a

form which is useful in the neural network setting, where it is computationally efficient to re-draw predictors for every prediction, but we make T passes through the dataset to ensure a tight bound. This formulation is considerably more computationally efficient than drawing a single h for every pass of the dataset.[4]

Theorem 6.5 (Train Set Bound)**.** *Let $\ell \in [0,1]$, Q be some distribution over predictors and $h^{i,t} \overset{\text{iid}}{\sim} Q$ be i.i.d. draws for $i \in [m], t \in [T]$. Then with probability at least $1-\delta'$,*

$$\mathop{\mathbb{E}}_{f\sim Q} \widehat{\mathcal{L}}_0(f) \leq \frac{1}{mT} \sum_{i=1}^{m} \sum_{t=1}^{T} \ell(h^{i,t}, (\boldsymbol{x}_i, y_i)) + \sqrt{\frac{\log \frac{1}{\delta'}}{2mT}}.$$

In our results, we will set $\delta' = 0.01$ (zero in the binary SHEL case), $T = 20$, and the generalisation bound $\delta = 0.025$; combining them our overall results will hold with probability at least $\delta + \delta' = 0.035$, as in Dziugaite and Roy (2017).

Variance Parameters β and σ. The parameters β, σ_V and σ_U control the variances of the weights in the stochastic estimator defined by Q, but fulfil different functions. The β parameter appears in the non-stochastic shallow network $F_{U,V}$ and thus affects the final predictions made and the training by SGD, and can be related to data normalisation as discussed above. We therefore set it to the fixed value of $\beta = 5$ in all our experiments.

However the σ parameters appear only on the right hand side of the bounds for multi-class SHEL and GELU, and can be tuned to provide the tightest bounds—as they grow the KL term reduces but the performance of Q will degrade. We therefore optimise the final bounds over a grid of σ values as follows: choose a prior grid of σ_V values, $\sigma_V \in \{\sigma_V^1, \ldots, \sigma_V^r\}$, and combine via a union bound argument to add a $\log(r)$ term to κ where r is the number of grid elements. The same practice is applied to σ_U in the GELU case. In practice we use a grid $\sigma \in \{0.05, 0.06, \ldots, 0.2\}$ for both. Thus the tuning of σ_U and σ_V is not a feature of the network like β, but rather a tool to optimise the tightness of the bounds.

The parameter T appearing in Theorem 6.1 fufils a similar function, trading off the performance of $\widehat{\mathcal{L}}_0(Q^{\otimes T})$ versus the complexity term, but we do not optimise it like the above in our experiments, fixing it to $T = 500$ in all our results.

Coupling Procedure. We adopt a 60%-prefix coupling procedure for generating the prior weights U^n, V^n (rather than U^0, V^0, and similarly in the binary case) as in Dziugaite et al. (2021). This works by taking the first 60% of training examples used in our original SGD run and looping them in the same order for up to 4000 epochs. Note that this also replaces m by $m-n$ and S by S^{bnd} in the bounds, so we are making a trade off between optimising

[4] Note the result below is based on a version of Hoeffding's bound which does not require identical means for different random variables. See also Biggs (2022) for a more sophisticated version of the same idea.

the prior and the tightness of the bound (affected by $m-n$). These are used to train a prior model of the same architecture with the same learning rate from the same initialisation (this is valid because the initialisation is data-independent). The best bound from the generated prior weights was chosen (with a small penalty for this choice added to the bound via a union argument).

Numerical Results. In order to evaluate the quality of the bounds provided, we made many evaluations of the bound under many different training scenarios. In particular we show that the bound behaves in similar ways to the test error on changes of the width, learning rate, training set size and random relabelling of the data.

The following results follow by training β-normalised SHEL and GELU networks with stochastic gradient descent on the cross-entropy loss to a fixed cross entropy value of 0.3 for Fashion-MNIST and 0.1 for MNIST. When evaluating the binary SHEL bound (Theorem 6.1) we use binarised versions of the datasets where the two classes consist of the combined classes $\{0,\ldots,4\}$ and $\{5,\ldots,9\}$ respectively (following Dziugaite and Roy, 2017; Letarte et al., 2019), training to cross-entropy values of 0.2 for Bin-F (binarised Fashion-MNIST) and 0.1 for Bin-M (binarised MNIST) respectively. We trained using SGD with momentum $=0.9$ (as suggested by Hendrycks and Gimpel, 2016 and following Biggs and Guedj, 2022a) and a batch size of 200, or without momentum and a batch size of 1000 (with this larger batch size stabilising training). We evaluated for ten different random seeds, a grid search of learning rates $\in\{0.1, 0.03, 0.01\}$ without momentum, and additionally $\in\{0.003, 0.001\}$ with momentum (where small learning rate convergence was considerably faster), and widths $\in\{50, 100, 200, 400, 800, 1600\}$ to generate the bounds in Table 6.1.

From these results we also show plots in Fig. 6.1 of the test error, stochastic error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ and best prior bound versus width for the different dataset/activation combinations, with more plots given in the appendix. We also note here that in all except the width $=50$ case, our neural networks have more parameters than there are train data points (60000). Using the test set, we also verified that assumption $(\star)$ holds in all cases in which it is used to provide bounds.

6.6 Discussion

In Table 6.1 we have given the first non-vacuous bounds for two types of deterministic neural networks trained on MNIST and Fashion-MNIST through a standard SGD learning algorithm, both with and without momentum. The coupled bounds are in all cases far from vacuous, with even the full bounds being non-vacuous in most cases, particularly on the easier MNIST task. Further, Figs. 6.1 and 6.2 show that the bounds are robustly non-vacuous across a range of widths and learning rates. Since these are direct bounds on $\mathcal{L}_0(F_{U,V})$ rather than

Best Coupled Bounds with Momentum

	Data	Test Err	Full Bnd	Coupled Bnd
SHEL	Bin-M	0.038	0.837	0.286
SHEL	Bin-F	0.085	0.426	0.297
SHEL	MNIST	0.046	0.772	0.490
SHEL	Fashion	0.150	0.984	0.727
GELU	MNIST	0.043	0.693	0.293
GELU	Fashion	0.153	0.976	0.568

Best Coupled Bounds without Momentum

	Data	Test Err	Full Bnd	Coupled Bnd
SHEL	Bin-M	0.037	0.835	0.286
SHEL	Bin-F	0.085	0.425	0.300
SHEL	MNIST	0.038	0.821	0.522
SHEL	Fashion	0.136	1.109	0.844
GELU	MNIST	0.036	0.742	0.317
GELU	Fashion	0.135	1.100	0.709

Table 6.1: Results for β-normalised (with $\beta = 5$) SHEL and GELU networks trained with and without momentum SGD on MNIST, Fashion-MNIST and binarised versions of the above, after a grid search of learning rates and widths as described above. Results shown are those obtaining the tightest coupled bound (calculated using Theorem 6.2 and Theorem 6.3 for the multi-class datasets, and Theorem 6.1 for the binary datasets), with the accompanying full train set bound and test error for the same hyper-parameter settings.

the usual PAC-Bayes $\mathbb{E}_{f\sim Q}\,\mathcal{L}_0(f)$, we emphasise that (for fixed hyper-parameters) no trade off is made between the tightness of the bound and the real test set performance, which is usually worse for a higher-variance (and thus more tightly bounded) Q.

Stability and Robustness Trade-Off. The two main contributions to the bound are the empirical error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0(f)$ and the KL divergence incorporated in κ. $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0(f)$ can be seen roughly as measuring a combination of the difficulty of the task for our predictor $F_{U,V}$ combined with some kind of perturbation resistance of its weights (like the idea of a flat minimum originated in Hinton and van Camp, 1993 and discussed at length by Dziugaite and Roy, 2017); while κ is here an empirical measure of the stability of the training method, scaled by the inverse width of the perturbation robustness.

When optimising the trade-off between these terms through a choice of σ_U, σ_V values,

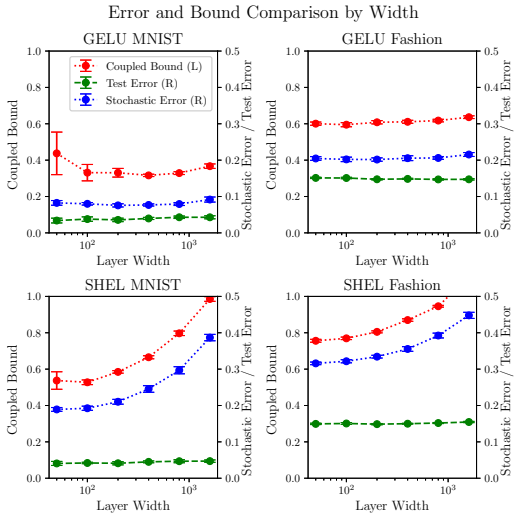

Figure 6.1: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus width for SHEL and GELU networks trained with momentum SGD and learning rate 0.01 on Fashion-MNIST and MNIST. Error bars show 1 standard deviation from ten different random seeds. The different scales are chosen so the trade-off between $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ and complexity terms can be seen more easily by neglecting the overall factor of 2, and the trends can be seen more clearly. We include an option in our code to generate these figures with a common scaling instead.

we find that the complexity contribution to the bound remains relatively consistent across datasets and architectures, while it is the stochastic error that varies. This is especially true of multi-class SHEL networks as seen in Fig. 6.1, perhaps since there is no easy way to set the stochastic error small by adjusting the variability of the Q hidden layer. This is in direct contrast to many works (Dziugaite et al., 2020; Jiang et al., 2020b) evaluating the predictive ability of PAC-Bayesian bounds for generalisation on hyper-parameter changes, which fix the weight variances as the largest leading to a bound on $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0(f)$ of a fixed value, say 0.1. Our results show that this approach may be sub-optimal for predicting generalisation, if as in our results the optimal trade-off tends to fix the κ term and trade off the size of $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0(f)$ instead of the reverse[5].

Width Comparison. For the width comparisons we note that it is difficult to discern the real trend in the out-of-sample error of our trained networks. The test sets only have 10000 examples and thus any test-set estimate of $\mathcal{L}_0(F_{U,V})$ is subject to error; if the differences between test errors of two networks of different widths is smaller than about 0.02 (obtained through a Hoeffding bound) it is not possible to say if generalisation is better or worse. It is

[5]The use of bi-criterion plots as suggested by Neyshabur et al. (2017) may therefore offer an better alternative when comparing vacuous bounds.

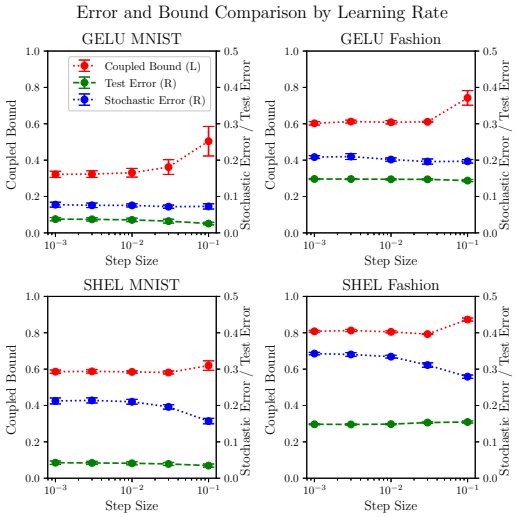

Figure 6.2: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus learning rate for width 200 SHEL and GELU networks trained with momentum SGD on Fashion-MNIST and MNIST. Scales are as in Fig. 6.1.

therefore possible that the pattern of weaker bounds for wider SHEL networks seen is a strong amplification of an existing trend, but it seems more likely it is an artefact of the bound shared with that of Biggs and Guedj (2022a). Assuming the latter conclusion that the trained network true error really is relatively width-independent, the GELU bound does better matching this prediction (with this also being true in the momentum-free case, see appendix). The value of $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ stays roughly constant as width increases, while we observe that the optimal bound σ_U tends to decrease with increasing width. We attribute to this the tighter bounds for wide GELU networks, since the SHEL network has no comparable way to reduce the randomness of the hidden layer in Q, as we discuss at the end of Section 6.4.2.

Lower-Variance Stochastic Predictions. Following from the above, we note that in general $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ is smaller for comparably-trained GELU networks than the SHEL networks. We speculate that this arises from the increased randomness of the hidden layer of Q in Theorem 6.2: the sign activation is only $\{+1,-1\}$-valued and the amount of information coming through this layer is therefore more limited; and a $\{+1,-1\}$-valued random variable has maximum variance among $[+1,-1]$-bounded variables of given mean. In future work we will explore whether variance reduction techniques such as averaging multiple samples for each activation can improve the tightness of the bounds, but we also emphasise both that the bounds are still non-vacuous across a range of widths, and that the ability to adjust this variability is a central advantage of our new GELU formulation.

Learning Rate Comparison and Stability. In the case of training with momentum SGD we see that a very large learning rate leads to weaker and higher-variance bounds, with significantly larger norm contribution in κ. We speculate this arises because of the reduced stability at such high rates: we found in general that small batch sizes (particularly under vanilla SGD) and fast learning rates caused the training trajectory of U^n, V^n to diverge more greatly from that of U, V.

Improving Prior Coupling. With the instability of high learning rates and the empirical observation that in many cases $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0(f)$ was very close to $\mathbb{E}_{f\sim Q}\,\mathcal{L}_0(f)$ (as estimated from the test set), we see that there is a degree of slackness in the bound arising from the κ term. We speculate that it may be possible to make more efficient use of the sample S in constructing U^n, V^n to reduce this term further. This might be possible through an improved coupling scheme, or through extra side-channel information from S^{bnd} which can be compressed (as per Zhou et al., 2019) or is utilised in a differentially-private manner (as by Dziugaite and Roy, 2018).

Majority Votes. In our results we rely on the novel idea of randomised single-hidden-layer neural networks as the expectation or majority vote of randomised predictors for de-randomisation of our PAC-Bayes bound. For the multi-class bounds we rely on an additional assumption, so a first step in future work could be providing further conditions under which this assumption can be justified without relying on a test set. Next, we found empirically (similarly to many PAC-Bayesian works) that $\mathbb{E}_{f\sim Q}\,\mathcal{L}_0(f) > \mathcal{L}_0(F_{U,V})$, in other words the derandomised predictor was better than the stochastic version on the test set. By de-randomising through the first order bound, we introduce a factor of 2 which cannot be tight in such cases. Removal of this term would lead to considerably tighter bounds and even non-vacuous bounds for CIFAR-10 (Krizhevsky, 2009), based on preliminary experiments, where the training error for one-hidden-layer networks on CIFAR-10 was greater than 0.5 so such bounds could not be non-vacuous, but the final bounds were only around $1.1-1.2$. Improved bounds for the majority vote have been the focus of a wide variety of PAC-Bayesian works (Lacasse et al., 2006; Masegosa et al., 2020), and can theoretically give tighter results for $\mathcal{L}_0(\text{MV}_Q)$ than $\mathbb{E}_{f\sim Q}\,\mathcal{L}_0(f)$, but these are not yet competitive. They universally led to inferior or vacuous results in preliminary experiments. However, there is still much scope for exploration here: alternative formulations of the oracle C-bound lead to different empirical bounds, and improvement of the KL term (which appears more times in an empirical C-bound than Theorem 3.12) may improve these bounds more than the first order one. We also hope that offering this new perspective on one-hidden-layer networks as majority votes can lead to better understanding of their properties, and perhaps even of closely-related Gaussian

processes (Neal, 1996).

Deeper networks and convolutions. An extremely interesting question whether this approach will generalise to convolutions or deeper networks. For convolutions, the parameter sharing is not a problem as separate samples can be taken for each convolution kernel position (although potentially at a large KL divergence cost that might be mitigated through the use of symmetry). For deeper networks the answer is less clear, but the empirically-observed stability of most trained networks to weight perturbation would suggest that the mode of a Bayesian neural network may at least be a close approximation to its majority vote, a connection that could lead to further results.

Summary. We have provided non-vacuous generalisation bounds for shallow neural networks through novel methods that make a promising new link to majority votes. Although some aspects of our approach have recently appeared in the PAC-Bayesian literature on neural networks, we note that all previous results obtaining non-vacuous generalisation bounds only apply to randomised versions of neural networks. This often leads to degraded test set performance versus a deterministic predictor. By providing bounds directly on the deterministic networks we provide a setting through which the impact of robustness, flat-minima and stability on generalisation can be explored directly, without making potentially sub-optimal trade-offs or invoking stringent assumptions.

In future work we intend to address two main potential sources of improvement: through progress in majority votes to tighten the step from stochastic to deterministic predictor; and through development of the prior (perhaps thorough improved utilisation of data), a strand running parallel to much PAC-Bayesian research on neural networks.

6.A Proofs

Proof of Theorem 6.1. Consider randomised functions $\operatorname{sign}(\boldsymbol{w}\cdot\boldsymbol{x})$ with $\boldsymbol{w}\sim\rho$. ρ is a mixture of Gaussians distribution with $2d_{\text{hid}}$ components. We denote by $\rho(k)=\text{Categ}(q)$ the distribution over the choice of component k, with weights

$$q=\frac{1}{\|\boldsymbol{v}\|_1}[\max(0,\boldsymbol{v}_1),\dots,\max(0,\boldsymbol{v}_{d_{\text{hid}}}),\max(0,-\boldsymbol{v}_1),\dots,\max(0,-\boldsymbol{v}_{d_{\text{hid}}})].$$

$\rho(\boldsymbol{w}|k)$ denotes the corresponding components, with distributions $\rho(\boldsymbol{w}|k)=\mathcal{N}(\boldsymbol{u}_k,\frac{1}{2}\beta^{-2}I)$ for $k\in\{1,\dots,d_{\text{hid}}\}$, and $\rho(\boldsymbol{w}|k)=\mathcal{N}(-\boldsymbol{u}_k,\frac{1}{2}\beta^{-2}I)$ for $k\in\{d_{\text{hid}}+1,\dots,2d_{\text{hid}}\}$. $\boldsymbol{u}_k$ are the rows of U. This dimension-doubling trick accommodates the use of negative final-layer weights.

It is easy to show that $\mathbb{E}_{\boldsymbol{w}\sim\rho}\operatorname{sign}(\boldsymbol{w}\cdot\boldsymbol{x})=\frac{1}{\|\boldsymbol{v}\|_1}F(\boldsymbol{x})$, where we abbreviate by $F(\boldsymbol{x})=F^{\text{erf}}_{U,\boldsymbol{v}}(\boldsymbol{x})$ the SHEL network with parameters $U,\boldsymbol{v}$ as given above. This follows using the expectation of a mixture followed by using the aggregation of a sign function under a Gaussian weight given in Eq. (6.1), which gives

$$\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\rho}\operatorname{sign}(\boldsymbol{w}\cdot\boldsymbol{x})=\sum_{k=1}^{d_{\text{hid}}}q_k\operatorname{erf}\left(\beta\frac{\boldsymbol{u}_k\cdot\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\right)+\sum_{k=d_{\text{hid}}+1}^{2d_{\text{hid}}}q_k\operatorname{erf}\left(\beta\frac{-\boldsymbol{u}_k\cdot\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\right)=\frac{F(\boldsymbol{x})}{\|\boldsymbol{v}\|_1}$$

To obtain a PAC-Bayes bound in full, we also choose a set of prior weights $U^0,\boldsymbol{v}^0$ to define a π distribution that takes the same structure as ρ. The mixture index distribution is $\pi(k)=\text{Categ}(p)$ with

$$p=\frac{1}{2\|\boldsymbol{v}^0\|_1}[|\boldsymbol{v}^0_1|,\dots,|\boldsymbol{v}^0_{d_{\text{hid}}}|,|\boldsymbol{v}^0_1|,\dots,|\boldsymbol{v}^0_{d_{\text{hid}}}|],$$

and component distributions, $\pi(\boldsymbol{w}|k)$, defined as per $\rho(\boldsymbol{w}|k)$ but with weights $\boldsymbol{u}^0_k$ instead. Using the chain rule for KL divergence (Cover and Thomas, 2006) twice and the non-negativity of the KL,

$$\operatorname{KL}(\rho(\boldsymbol{w}),\pi(\boldsymbol{w}))\leq\operatorname{KL}(\rho(\boldsymbol{w},k),\pi(\boldsymbol{w},k))=\operatorname{KL}(\rho(\boldsymbol{w}|k),\pi(\boldsymbol{w}|k))+\operatorname{KL}(\rho(k),\pi(k)) \tag{6.5}$$

where $\rho(\boldsymbol{w},k)$ and $\pi(\boldsymbol{w},k)$ are the joint distributions on $\boldsymbol{w}$ and mixture index k. From the definitions of the KL divergence for categorical and Gaussian distributions in the above,

$$\operatorname{KL}(\rho(\boldsymbol{w}),\pi(\boldsymbol{w}))\leq\sum_{k=1}^{d_{\text{hid}}}q_k\beta\|\boldsymbol{u}_k-\boldsymbol{u}^0_k\|_2^2+\sum_{k=1}^{d_{\text{hid}}}q_k\log\frac{q_k}{p_k}=\kappa.$$

To move to a PAC-Bayes bound on the SHEL network we consider averaging copies of the above, so our overall posterior takes the form $f(\boldsymbol{x})=\frac{1}{T}\sum_{t=1}^{T}\operatorname{sign}(\boldsymbol{w}^t\cdot\boldsymbol{x})\sim Q$ with $\boldsymbol{w}^1,\dots,\boldsymbol{w}^T\overset{\text{iid}}{\sim}\rho$. The prior is defined analogously using π. $\operatorname{KL}(Q,P)\leq T\operatorname{KL}(\rho(\boldsymbol{w}),\pi(\boldsymbol{w}))$ by the i.i.d. nature of the $\boldsymbol{w}^t$.

The predictions of the SHEL network, $\operatorname{sign}F(\boldsymbol{x})$, are equivalent to a majority vote of $f(\boldsymbol{x})$, since $\text{MV}(\boldsymbol{x})=\operatorname{sign}(\mathbb{E}\operatorname{sign}(f(\boldsymbol{x})))$ is 1 if $\mathbb{E}f(\boldsymbol{x})\propto F(\boldsymbol{x})\geq 0$ and vice-versa for -1. Therefore

the first order bound can be used to see that $\ell_0(F,(\boldsymbol{x},y)) \le 2\mathbb{E}_Q \ell_0(f,(\boldsymbol{x},y))$. Combining with Theorem 3.12, the following holds with probability $\ge 1-\delta$

$$\mathcal{L}_0(F^{\mathrm{erf}}_{U,\boldsymbol{v}}) \le 2\,\mathrm{kl}^{-1}\left(\mathop{\mathbb{E}}_{f\sim Q}\widehat{\mathcal{L}}_0(f),\ \frac{T\kappa+\log\frac{2\sqrt{m}}{\delta}}{m}\right).$$

To complete the result we also note the closed form for the Q-expected empirical misclassification loss:

$$\begin{aligned}
\mathop{\mathbb{E}}_{Q}\ell_0(f,(\boldsymbol{x},y)) &= \mathop{\mathbb{P}}_{Q}(yf(\boldsymbol{x})\le 0)\\
&= \mathop{\mathbb{P}}_{Q}\left(\sum_{t=1}^{T} y\,\mathrm{sign}(\boldsymbol{w}^t\cdot\boldsymbol{x})\le 0\right)\\
&= \mathop{\mathbb{P}}_{Q}\left(\sum_{t=1}^{T}\frac{1}{2}(y\,\mathrm{sign}(\boldsymbol{w}^t\cdot\boldsymbol{x})+1)\le\frac{1}{2}T\right)\\
&= \mathop{\mathbb{P}}_{Q}\left(\sum_{t=1}^{T}\mathbf{1}_{y=\mathrm{sign}(\boldsymbol{w}^t\cdot\boldsymbol{x})}\le\frac{1}{2}T\right)\\
&= \mathrm{Bin}\left(\frac{T}{2};T,\mathop{\mathbb{P}}_{\boldsymbol{w}\sim\rho}(y=\mathrm{sign}(\boldsymbol{w}\cdot\boldsymbol{x}))\right)\\
&= \mathrm{Bin}\left(\frac{T}{2};T,\frac{1}{2}\left(1+\frac{yF(\boldsymbol{x})}{\|v\|_1}\right)\right)
\end{aligned}$$

where we have interchanged $\mathbf{1}_{y=\mathrm{sign}(\boldsymbol{w}\cdot\boldsymbol{x})}=\frac{1}{2}(y\,\mathrm{sign}(\boldsymbol{w}\cdot\boldsymbol{x})+1)$.

All of the above can be readily extended to the data-dependent prior case, replacing $U^0\to U^n$, $\boldsymbol{v}^0\to\boldsymbol{v}^n$, $m\to m-n$, and $\widehat{\mathcal{L}}_0\to\widehat{\mathcal{L}}_0^{\mathrm{bnd}}$. □

Proof of Theorem 6.4. We are considering a distribution on functions of the form $\sum_k \boldsymbol{w}^k h^k(\boldsymbol{x})$ where for each index $k\in[d_{\mathrm{hid}}]$ we have $\boldsymbol{w}^k\sim\mathcal{N}(\frac{1}{\sigma_V}\boldsymbol{v}_k,I)$ and $h^k\sim Q^k$. This slightly different formulation can take advantage of the scaling-invariance of the final layer to the misclassification loss when $V^0=0$, so we can then choose $\sigma_V>0$ arbitrarily. The expectation of this takes the form given in Eq. (6.3) scaled by $1/\sigma_V$ and leads to the empirical loss above.

Given another distribution P taking a similar form with $\boldsymbol{w}^k\sim\mathcal{N}(\frac{1}{\sigma_V}\boldsymbol{v}_k^0,I)$ and components P^k, the KL divergence can be expressed (using the chain rule for KL divergence) as

$$\mathrm{KL}\,(Q,P)\le\sum_{k=1}^{d_{\mathrm{hid}}}\mathrm{KL}\left(Q^k,P^k\right)+\frac{\|V-V^0\|_F^2}{2\sigma_V^2}.$$

We prove the overall bound by combining Theorem 3.12 with the assumption ($\star$). □

Proof of Theorem 6.2. Apply the bound from Theorem 6.4 with the individual units as $h^k(\boldsymbol{x})=\mathrm{sign}(\boldsymbol{w}^k\cdot\boldsymbol{x})$ and $\boldsymbol{w}^k\sim\mathcal{N}(\boldsymbol{u}_k,\frac{1}{2}\beta^{-2}I)$ alongside Theorem 6.4. The aggregated form of the sign activation function is given in (6.1). The prior takes the same form as the posterior with weight means U^0,V^0 and the same variances, leading to the form of KL divergence for Gaussian weights given in κ. □

Proof of Theorem 6.3. The proof takes the same form as that of Theorem 6.2. We note that the expectation under the given probability distributions of $\mathbb{E}[W_2(\mathbf{1}_{W_1\boldsymbol{x}} \otimes (W_1'\boldsymbol{x}))] = \|\boldsymbol{x}\|_2 F_{U,V}^{\mathrm{GELU}}(\boldsymbol{x})$, but since the misclassification loss is scaling-invariant this gives equivalent results. Choosing appropriate prior forms as in Theorem 6.2 gives the KL divergence which we substitute into Theorem 6.4. □

Proof of Theorem 6.5. Define $\xi = \sum_{t=1}^{T}\sum_{i=1}^{m} \frac{1}{mT}\ell(h^{i,t},(\boldsymbol{x}_i,y_i))$ which has expectation $\mathbb{E}_Q\,\xi = \mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0(f)$. This quantity is a sum of mT independent random variables in $\{0,1/mT\}$, where there are T variables with mean $\mathbb{E}_Q \frac{1}{mT}\ell(h,(\boldsymbol{x}_i,y_i))$ for each i. Hoeffding's bound (which does not require identical means) then gives the result. □

6.B Additional Results and Code

We provide all of our results and code to reproduce them along with the figures (including with the option of using the same scaling for the bound and errors, as described in Fig. 6.1) in the supplementary material for the original publication Biggs and Guedj (2022b). We also note here that the "erf" function is included in a wide variety of common deep learning libraries.

Here we also provide Figs. 6.3 and 6.4 similar to Figs. 6.1 and 6.2 for GELU and SHEL networks trained without momentum and with a batch size of 1000, as described in Section 6.5. We then also provide further similar plots for networks trained with momentum and a batch size of 200 as in Section 6.5 with different learning rates and widths, to show the similar behaviour across a variety of regimes.

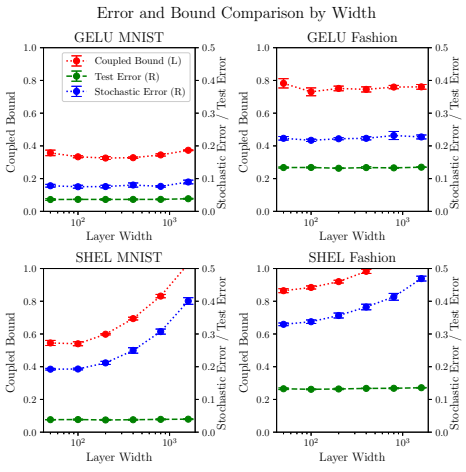

Figure 6.3: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus width for SHEL and GELU networks trained with vanilla SGD and learning rate 0.01 on Fashion-MNIST and MNIST. Scales are as in Fig. 6.1.

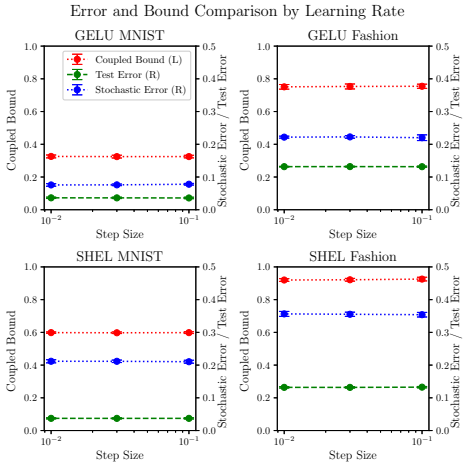

Figure 6.4: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus learning rate for width 200 SHEL and GELU networks trained with vanilla SGD on Fashion-MNIST and MNIST. Scales are as in Fig. 6.1.

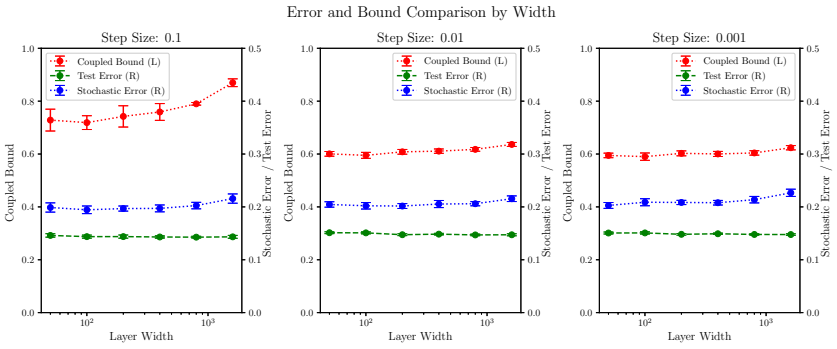

Figure 6.5: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a GELU network trained with momentum on Fashion-MNIST.

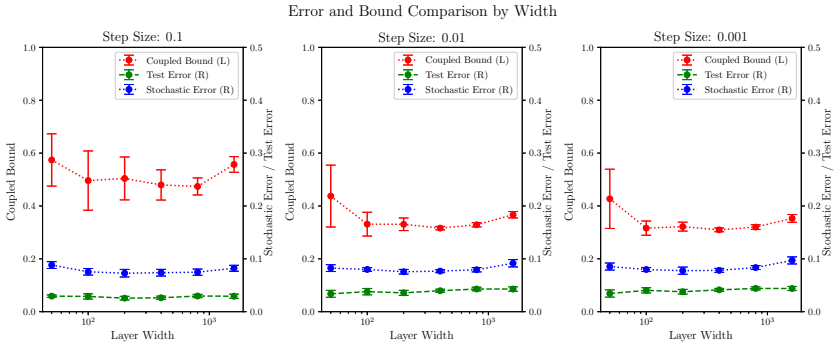

Figure 6.6: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a GELU network trained with momentum on MNIST.

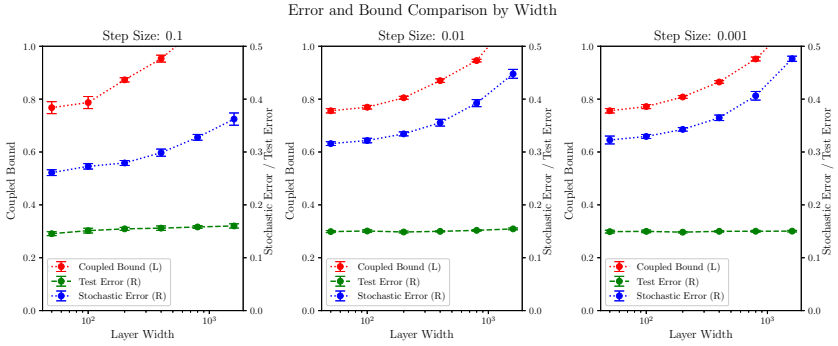

Figure 6.7: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a SHEL network trained with momentum on Fashion-MNIST.

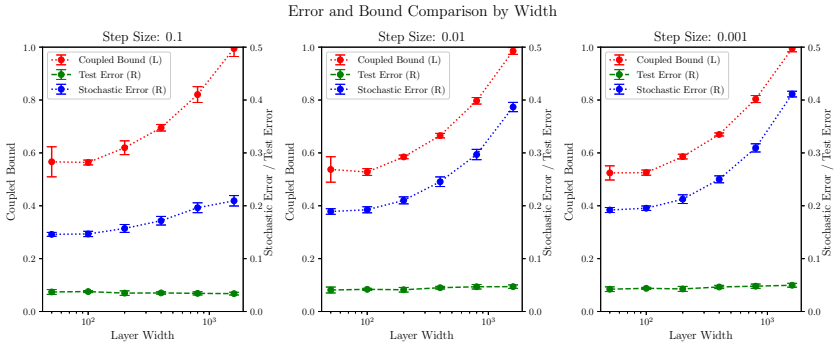

Figure 6.8: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a SHEL network trained with momentum on MNIST.

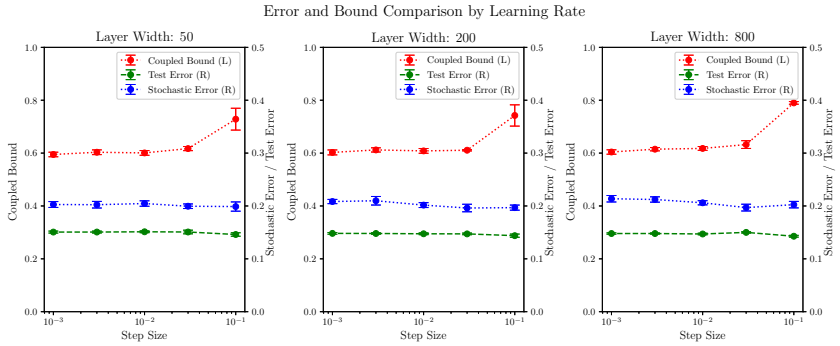

Figure 6.9: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a GELU network trained with momentum on Fashion-MNIST.

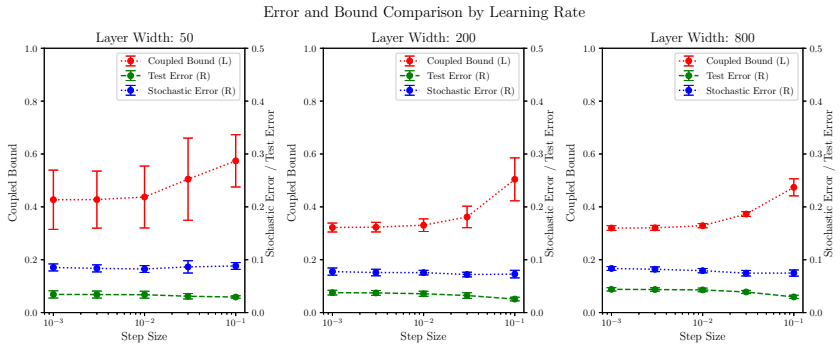

Figure 6.10: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a GELU network trained with momentum on MNIST.

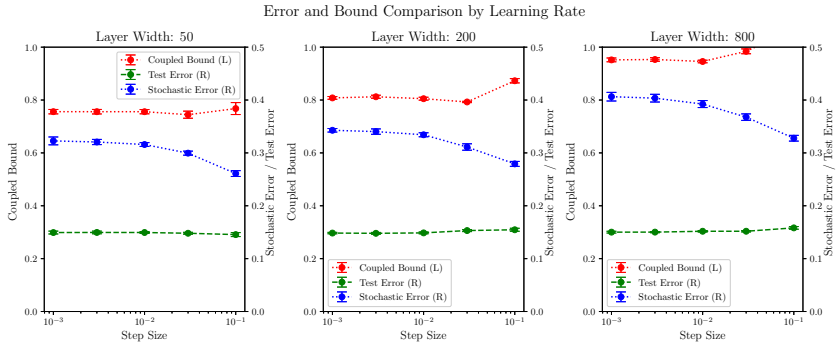

Figure 6.11: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a SHEL network trained with momentum on Fashion-MNIST.

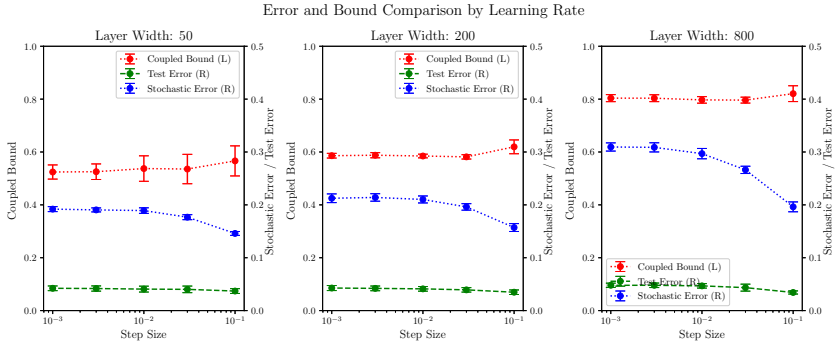

Figure 6.12: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\mathbb{E}_{f\sim Q}\,\widehat{\mathcal{L}}_0^{\mathrm{bnd}}(f)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a SHEL network trained with momentum on MNIST.

Chapter 7

On Margins and Generalisation for Voting Classifiers

We study the generalisation properties of majority voting on finite ensembles of classifiers, proving margin-based generalisation bounds via the PAC-Bayes theory. These provide state-of-the-art guarantees on a number of classification tasks. Our central results leverage the Dirichlet posteriors studied recently by Zantedeschi et al. (2021) for training voting classifiers; in contrast to that work our bounds apply to non-randomised votes via the use of margins. Our contributions add perspective to the debate on the "margins theory" proposed by Schapire et al. (1998) for the generalisation of ensemble classifiers.

7.1 Introduction

Weighted ensemble methods are among the most widely-used and effective algorithms known in machine learning. Variants of boosting (Chen and Guestrin, 2016; Freund and Schapire, 1997) are state-of-the-art in a wide variety of tasks (Nielsen, 2016; Shwartz-Ziv and Armon, 2022) and methods such as random forest (Breiman, 2001) are among the most commonly-used in machine learning competitions (see, *e.g.*, Bell and Koren, 2007; Uriot et al., 2021), valued both for their excellent results and interpretability. Even when these algorithms do not directly produce the best learners for a task, the best performance in competitions is often obtained by an ensemble of "strong learners"—the output of a collection of different algorithms trained on the data—contrasted to the weak learners usually considered in the ensemble learning literature.

Among the oldest ideas to explain the performance of ensemble classifiers, and machine learning methods in general, is the concept of margins. First introduced to analyse the Perceptron algorithm (Novikoff, 1962), margins relate closely to the idea of confidence in predictions in ensemble learning, with a large margin implying that a considerable weighted

fraction of voters chose the same answer. This was first leveraged to obtain early margin-based generalisation bounds for ensembles by Schapire et al. (1998), in an attempt to understand the excellent generalisation of boosting, a surprising result given classical Vapnik-Chervonenkis theory. This "margins theory" was explored further in a number of works (Gao and Zhou, 2013; Grønlund et al., 2020; Wang et al., 2008) and is among the leading explanations for the success of such methods and boosting in particular.

The same thread of margin bounds for ensemble methods has also been taken up in parallel in PAC-Bayes theory by Biggs and Guedj (2022a); Langford and Seeger (2001). PAC-Bayes provides a natural framework both for deriving margin bounds, and for considering ensemble methods in general, particularly majority votes where the largest-weighted ensemble prediction is taken. Within the framework, the weightings are typically considered as the parameter of a categorical distribution over individual voters. PAC-Bayes theorems then directly provide generalisation bounds for the performance of this "randomised" proxy for the majority vote, *a.k.a.* Gibbs classifier. These can then be de-randomised by such margin-based techniques, or through a variety of oracle bounds (Lacasse et al., 2010; Langford and Shawe-Taylor, 2003; Masegosa et al., 2020; Shawe-Taylor and Hardoon, 2009), motivating new learning algorithms (Germain et al., 2015; Lacasse et al., 2006; Laviolette et al., 2017; Lorenzen et al., 2019; Roy et al., 2011; Viallard et al., 2021; Wu et al., 2021).

Uniquely among PAC-Bayesian approaches, Zantedeschi et al. (2021) instead consider Dirichlet distributions over the voters. Any sample from this distribution already implies a vector of voting weights, and it is on the performance and optimisation of these "stochastic majority votes" they primarily focus. As an aside, they provide an oracle result which allows their bounds to be de-randomised, but this introduces an irreducible factor such that the bound on the true fixed vote can never be less than double that of the stochastic version. It also neglects to leverage the generally high confidence of predictions obtained by their algorithm.

Our contribution. By combining tools from margin bounds and the use of Dirichlet majority votes, we provide a new margin bound for non-randomised majority votes. This is in contrast to Zantedeschi et al. (2021) which primarily considers stochastic majority votes. Our bound empirically compares very favourably to existing margin bounds and in contrast to them are applicable to multi-class classification. Remarkably, our empirical results are also sharper than existing PAC-Bayesian ones, even when the algorithm optimising those bounds is used.

Our primary tool is a new result relating the margin loss of these stochastic votes to the misclassification loss of the non-randomised ones in a surprisingly sharp way. This tool can additionally be utilised alongside a further idea from Zantedeschi et al. (2021) to obtain an alternative form of the bound which is more amenable to optimisation. Through this work

we provide further support to the margins theory for ensembles, showing that near-sharp bounds based on margins alone can be obtained on a variety of real-world tasks.

Outline. The rest of this section introduces the problem setup, notation and summarises main results. Section 7.2 provides background on PAC-Bayes, Dirichlet majority votes and margin bounds, relating them to our new results. Section 7.3 states and summarises our new theoretical results, giving the most relevant proofs (all remaining proofs are deferred to appendices). Section 7.4 empirically evaluates these new results before we conclude with an overall discussion in Section 7.5.

7.1.1 Notation and setting

We recall the definition of majority voting algorithms on a finite set of "base" classifiers, $\mathcal{H}_{\text{base}}$, from $\mathcal{X}$ (which will generally be arbitrary here, since we do not need assumptions about the base classifiers) to $\mathcal{Y} = [d_{\text{out}}] := \{1, \ldots, d_{\text{out}}\}$. The base classifiers $h_i \in \mathcal{H}_{\text{base}}$ take the form $h_i : \mathcal{X} \to \mathcal{Y}$ for $i \in [d_{\text{vot}}]$ so that $|\mathcal{H}_{\text{base}}| = d_{\text{vot}}$. Majority votes on a finite set consider weightings $\boldsymbol{\theta}$ in $\Delta_{d_{\text{vot}}-1}$, the simplex, and return the highest-weighted overall prediction. This can be expressed as

$$\text{MV}_{\boldsymbol{\theta}}(x) = \text{argmax}_{k \in \mathcal{Y}} \sum_{i \in [d_{\text{vot}}]} \theta_i \mathbf{1}_{h_i(x)=k} \,.$$

We are primarily interested in learning a weighting $\boldsymbol{\theta}$ with small misclassification risk based on the sample S.

The margin of majority vote $\text{MV}_{\boldsymbol{\theta}}$ on example (x, y) is derived from the minimal gap between the total weight assigned to the true class y and to any other predicted class:

$$M(\boldsymbol{\theta}, x, y) := \sum_{i:h_i(x)=y} \theta_i - \max_{k \neq y} \sum_{i:h_i(x)=k} \theta_i .$$

The corresponding margin loss is $\ell_\gamma(\text{MV}_{\boldsymbol{\theta}}, (x, y)) := \mathbf{1}_{M(\boldsymbol{\theta},x,y) \leq \gamma}$ for margin $\gamma \geq 0$.

7.1.2 Overview of results

Our main result is a margin bound of the following form: with high probability $\geq 1 - \delta$ over the sample, simultaneously for any $\boldsymbol{\theta} \in \Delta_{d_{\text{vot}}-1}$ and $K > 0$,

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \leq \mathcal{O}\left(\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}}) + e^{-K\gamma^2} + \frac{\mathbb{D}_{\text{Dir}}(K\boldsymbol{\theta}, \mathbf{1}) + \log \frac{m}{\delta}}{m} \right) \tag{7.1}$$

where $\mathbb{D}_{\text{Dir}}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the KL divergence between Dirichlet random vectors with parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, with $\mathbf{1}$ a vector of ones implying a uniform Dirichlet prior distribution on the simplex. The term $e^{-K\gamma^2}$ is a de-randomisation penalty. The parameter K is chosen freely in an arbitrary data-dependent way[1] to balance the requirements of the different terms: it must

[1] This is possible because K arises as a parameter of the Dirichlet posterior considered in our proof, rather than as a hyper-parameter of the bound.

be large enough to decrease this exponential term, while too-large a parameter increases the KL divergence from the uniform prior. This result is surprisingly strong; in particular there is no dependence on the dimensionality (*i.e.*, number of voters d_{vot}) in the exponential term, an advantage discussed further in Section 7.3.2.

In Eq. (7.1), $\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}})$ is the 0-1 valued γ-margin loss which enables comparison with existing margin bounds for trained weighted ensembles. We further consider a second scenario, where the generalisation bound is also used to train the model itself. We note that the γ-margin loss $\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}})$ appearing in Eq. (7.1) has null gradients, so the bound cannot be directly optimised by gradient descent. To rectify this we also prove a variation of the bound, replacing the above loss by its expectation under a Dirichlet stochastic vote, $\mathbb{E}_{\boldsymbol{\xi}\sim\text{Dir}(K\boldsymbol{\theta})}\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\xi}})$, which is bounded in differentiable closed form to give an alternative, optimisation-friendly bound.

In our evaluations we focus on these two complementary scenarios, obtaining state-of-the-art empirical results. Across different scenarios and tasks our results outperform both existing margin bounds (including a sharpened version of the result from Biggs and Guedj (2022a) which may be of independent interest), and PAC-Bayes bounds, even when it is not used as the objective. Further, in contrast to existing margin bounds our results also hold for multi-class majority votes.

7.2 Background

7.2.1 PAC-Bayes bounds

We use the PAC-Bayes bound Theorem 3.12 as the basis for our results, leading to bounds involving the inverse small-kl function. We recall that $\text{kl}^{-1}(u,c) \in \mathcal{O}(u+c)$[2], giving Eq. (7.1) from Theorem 7.1 when using a uniform prior.

7.2.2 Margin bounds

In the learning theory literature there exists a rich tradition of using the concept of a margin, which quantifies the confidence of predictions, to explain generalisation. This is particularly evident in the case of voting algorithms such as boosting, where traditional Vapnik-Chervonenkis based techniques predict classical overfitting which is not ultimately observed. The "margins theory" was developed by Schapire et al. (1998) to explain this discrepancy. By considering the weightings $\boldsymbol{\theta}$ as the parameter of a categorical distribution, they proved a bound of the form (holding with probability greater than $1-\delta$ over the sample, as for all bounds in this section) $\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \leq \mathcal{L}_\gamma(\text{MV}_{\boldsymbol{\theta}}) + \widetilde{\mathcal{O}}\left(\frac{1}{\gamma\sqrt{m}}\right)$. Although there was initially some debate about the validity of the theory (Breiman, 1999), eventually Gao and

[2]I.e. $\text{kl}^{-1}(u,c) \leq C(u+c)$ for some constant C. Note we also have $\text{kl}^{-1}(u,c) \leq u + C'\sqrt{c})$ for a different constant C'; for small u the former bound can be tighter.

Zhou (2013, Theorem 4) provided the following improved bound which further supported that a large-margin voting classifier could generalise: simultaneously for any $\gamma > \sqrt{8/d_{\text{vot}}}$ and $\boldsymbol{\theta} \in \Delta_{d_{\text{vot}}-1}$ in binary classification,

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \leq \text{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}}), \frac{1}{m}\left(\frac{8\log(2d_{\text{vot}})}{\gamma^2}\log\frac{2m^2}{\log d_{\text{vot}}} + \log\frac{md_{\text{vot}}}{\delta}\right)\right) + \frac{\log d_{\text{vot}}}{m}. \quad (7.2)$$

More recently, a similar bound (proved through a PAC-Bayesian method based on Seeger et al., 2001 and also valid for only binary classification) was proved in Biggs and Guedj (2022a, Theorem 8). Here we give the improved variant Theorem 5.6 appearing in Chapter 5 that was provided as an intermediate step in the original proof and is strictly (and empirically considerably) sharper than the final result: for any fixed margin $\gamma > 0$, simultaneously for any $\boldsymbol{\theta} \in \Delta_{d_{\text{vot}}-1}$

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \leq \text{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}}) + \frac{1}{m}, \frac{1}{m}\left(\left\lceil 8\gamma^{-2}\log m\right\rceil \log d_{\text{vot}} + \log\frac{2\sqrt{m}}{\delta}\right)\right) + \frac{1}{m}. \quad (7.3)$$

Since $\gamma \in (0,1)$ for non-vacuous results, a union bound argument can be used to extend the above to fixed-precision γ, and this result has the advantage of being valid for small γ as are often observed empirically.

Our contributions. Firstly we mention the smaller contribution of the improved form of the bound from Biggs and Guedj (2022a) given in Eq. (7.3); a proof is given in Section 7.B alongside further refinements and evaluation. However we show that in many cases even this improved version and Eq. (7.2) give weak or vacuous results. As a result of this weakness (and thus perhaps null result for the margins theory applied to voting classifiers) we present a new margin bound in Theorem 7.1 based on Dirichlet distributions as a theoretical intermediate step. This is also valid in the multi-class case, unlike the above results which are only for binary classification. Empirically the bound is observed to give an enormous improvement in tightness than the existing margin bounds and in some cases is near-sharp.

7.2.3 Dirichlet stochastic majority votes

In most results from the PAC-Bayes framework, and in the proof of the existing results given in Section 7.2.2, the majority vote weightings $\boldsymbol{\theta}$ are considered the parameters of a categorical distribution over voters. Zantedeschi et al. (2021) instead consider PAC-Bayesian bounds (specifically, Theorem 3.12 with the misclassification loss) applied to a hypothesis class of majority votes of the form $\text{MV}_{\boldsymbol{\xi}}$, where $\boldsymbol{\xi} \sim \text{Dir}(\boldsymbol{\alpha})$ is drawn from a Dirichlet distribution with parameter $\boldsymbol{\alpha}$. This distribution has mean $\mathbb{E}\boldsymbol{\xi} = \boldsymbol{\alpha}/\sum_{i=1}^{d_{\text{vot}}}\alpha_i$ with a larger sum $\sum_{i=1}^{d_{\text{vot}}}\alpha_i$ giving a more concentrated or peaked distribution (see Section 7.A for more details).

Since $\boldsymbol{\xi}$ is randomised, the bounds from Zantedeschi et al. (2021) apply to "stochastic majority votes" rather than the more typical deterministic ones we consider here. However, the use of such Dirichlet distributions over voters in the PAC-Bayes bounds rather than the

more usual categorical ones is a major step forward as it allows the correlation between voters to be more carefully considered. This is because with a categorical distribution, the expected Gibbs risk is simply an average of the losses of individual predictors, without taking into account how well the combination of their predictions performs. Conversely, the Dirichlet distribution gives a (stochastic) majority vote of predictors, so if the errors of base voters are de-correlated, the better performance that results from their combination can be accounted for in the bound. We will utilise and de-randomise these stochastic majority votes as a stepping stone to bounds for deterministic predictors $\mathrm{MV}_{\boldsymbol{\theta}}$ directly.

As is common in the PAC-Bayes literature, Zantedeschi et al. (2021) use their new bound as an optimisation objective to obtain a new algorithm, here using stochastic gradient descent. The bound with Dirichlet posterior obtained directly from Theorem 3.12 includes the expected misclassification loss with respect to the Dirichlet parameters, $\mathbb{E}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}\,\ell(\mathrm{MV}_{\boldsymbol{\xi}},(x,y))$, which has null gradient for any sampled $\boldsymbol{\xi}$. They therefore additionally upper bound this term by the differentiable closed form

$$\mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}\ell(\mathrm{MV}_{\boldsymbol{\xi}},(x,y)) \leq I_{\frac{1}{2}}\left(\sum_{i:h_i(x)=y}\alpha_i,\ \sum_{i:h_i(x)\neq y}\alpha_i\right), \tag{7.4}$$

where $I_z(a,b)$ is the regularised incomplete beta function, which has a sigmoidal shape. The inequality is sharp in the binary classification case, and is used in the training objective and final evaluation of their method. As an aside, Zantedeschi et al. (2021) also proved an oracle bound which allows their result to be de-randomised, but this introduces a irreducible factor of two. This bound, which holds with probability at least $1-\delta$ over the sample for any $\boldsymbol{\theta}\in\Delta_{d_{\mathrm{vot}}-1}, K>0$ is given by

$$\mathcal{L}_0(\mathrm{MV}_{\boldsymbol{\theta}}) \leq 2\,\mathrm{kl}^{-1}\left(\mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(K\boldsymbol{\theta})}\widehat{\mathcal{L}}_0(\mathrm{MV}_{\boldsymbol{\xi}}),\ \frac{\mathbb{D}_{\mathrm{Dir}}(K\boldsymbol{\theta},\boldsymbol{\beta})+\log\frac{2\sqrt{m}}{\delta}}{m}\right).$$

Our contributions. Firstly, we provide a new margin bound for majority vote algorithms utilising Dirichlet posteriors as a theoretical stepping stone. We show that this bound gives sharper bounds on the misclassification loss than the bound from Zantedeschi et al. (2021), doing better than the irreducible factor, even when applied to the output of their algorithm. We show further that the bound is also tighter when applied to the outputs of other PAC-Bayes algorithms derived from "categorical"-type posteriors. Finally, we give an altered form of the bound involving the expectation of the margin loss $\mathbb{E}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}\,\ell_\gamma(\mathrm{MV}_{\boldsymbol{\xi}},(x,y))$ and a result analogous to Eq. (7.4) for this case. Through this we are able to obtain a new PAC-Bayes objective which is compared to existing PAC-Bayes optimisation methods.

7.3 Main results

Our main results use the idea of Dirichlet stochastic majority votes from Zantedeschi et al. (2021) as an intermediate step to prove new margin bounds for deterministic majority votes. In this section, first we give our main result in Theorem 7.1 and discuss further. In Section 7.3.1 we give an alternative bound obtained by a very similar method which is more amenable to optimisation, and we provide proofs for these results in Section 7.3.2.

The central step in these proofs is in constructing a proxy Dirichlet distribution $\boldsymbol{\xi} \sim \mathrm{Dir}(K\boldsymbol{\theta})$ over voters, the loss of which is bounded à la PAC-Bayes, and de-randomised using margins to obtain bounds directly for $\mathrm{MV}_{\boldsymbol{\theta}}$. The primary complexity term appearing in our bounds is therefore $\mathbb{D}_{\mathrm{Dir}}(K\boldsymbol{\theta}, \boldsymbol{\beta})$, the KL divergence between Dirichlet distributions with parameters $K\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ respectively. As with PAC-Bayes priors, $\boldsymbol{\beta}$ can be chosen in arbitrary sample-independent fashion, but we typically choose it as a vector of ones, giving a uniform distribution on the simplex as prior as in Eq. (7.1). The bounds also involve a de-randomisation penalty of $\mathcal{O}(e^{-K\gamma^2})$ where γ is the margin appearing in the loss; this term upper bounds the difference between our randomised proxy $\boldsymbol{\xi}$ and its mean $\boldsymbol{\theta}$ and gets smaller with K as the distribution concentrates tightly around its mean. This parameter K can be optimised in any data-dependent way to obtain the tightest final bound.

Theorem 7.1. *For any $\mathcal{D} \in \mathscr{P}(\mathcal{X} \times \mathcal{Y})$, $m \in \mathbb{N}_+$, margin $\gamma > 0$, $\delta \in (0,1)$, and prior $\boldsymbol{\beta} \in \mathbb{R}_+^{d_{\mathrm{vot}}}$, with probability at least $1-\delta$ over the sample $S \sim \mathcal{D}^m$ simultaneously for every $\boldsymbol{\theta} \in \Delta_{d_{\mathrm{vot}}-1}$ and $K > 0$,*

$$\mathcal{L}_0(\mathrm{MV}_{\boldsymbol{\theta}}) \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(\mathrm{MV}_{\boldsymbol{\theta}}) + e^{-\frac{1}{4}(K+1)\gamma^2}, \frac{\mathbb{D}_{\mathrm{Dir}}(K\boldsymbol{\theta}, \boldsymbol{\beta}) + \log\frac{2\sqrt{m}}{\delta}}{m}\right) + e^{-\frac{1}{4}(K+1)\gamma^2}.$$

Theorem 7.1 differs from the existing margin bounds of Eqs. (7.2) and (7.3), and Schapire et al. (1998) in a specific and significant way, with $\boldsymbol{\theta}$ appearing not only in the loss function $\widehat{\mathcal{L}}_\gamma(\mathrm{MV}_{\boldsymbol{\theta}})$, but *also* in the KL complexity term. Empirically we find our bound to be an improvement but it is possible to generate scenarios where the pre-existing bounds are non-vacuous while ours is not, since the KL divergence is unbounded for certain choices of $\boldsymbol{\theta}$, for example when one of the components is exactly zero. This difference arises because the existing bounds all use the idea of a categorical distribution with parameter $\boldsymbol{\theta}$ in their proofs (which has KL divergence from a uniform prior upper bounded by $\log d_{\mathrm{vot}}$), while we use a Dirichlet. This gains us the surprisingly tight de-randomisation result (Theorem 7.3) used in all proofs.

7.3.1 PAC-Bayes bound as objective

We note here that it is non-trivial to directly obtain a training objective for optimisation from Theorem 7.1, due to the non-differentiability of the margin loss $\widehat{\mathcal{L}}_\gamma(\mathrm{MV}_{\boldsymbol{\theta}})$. Therefore, in order to compare results with a wide variety of methods that optimise PAC-Bayes bounds

(including those used by Zantedeschi et al., 2021, as baselines), we obtain a relaxed and differentiable formulation in Theorem 7.2 for direct optimisation.

Theorem 7.2. *Under the conditions of Theorem 7.1 the following bound also holds*

$$\mathcal{L}_0(\mathrm{MV}_{\boldsymbol{\theta}}) \leq \mathrm{kl}^{-1}\left(\mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(K\boldsymbol{\theta})} \widehat{\mathcal{L}}_\gamma(\mathrm{MV}_{\boldsymbol{\xi}}),\ \frac{\mathbb{D}_{\mathrm{Dir}}(K\boldsymbol{\theta},\boldsymbol{\beta}) + \log\frac{2\sqrt{m}}{\delta}}{m}\right) + e^{-(K+1)\gamma^2}.$$

Using the incomplete Beta function $I_z(a,b)$ we also have the following result, which is sharp in the binary classification case,

$$\mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})} \ell_\gamma(\mathrm{MV}_{\boldsymbol{\xi}},(x,y)) \leq I_{\frac{1+\gamma}{2}}\left(\sum_{i:h_i(x)=y}\alpha_i,\ \sum_{i:h_i(x)\neq y}\alpha_i\right).$$

Theorem 7.2 has a stronger PAC-Bayesian flavour than Theorem 7.1, with an expected loss under some distribution appearing (complicating the final optimisation of K), while Theorem 7.1 takes a form much closer to that of a classical margin bound. The second part of the result is analogous to Eq. (7.4) used by Zantedeschi et al. (2021). We combine both parts to calculate the overall bound in closed form and obtain gradients for optimisation.

7.3.2 Proof of main results

The proof of Theorems 7.1 and 7.2 essentially follow from applying a simple PAC-Bayesian bound in combination with the key Theorem 7.3 below. In some sense this is our most important and novel result. Our whole approach is largely motivated by its surprising tightness; in particular there is no dependence on the dimension, which is avoided by careful use of the aggregation property of the Dirichlet distribution. This surprising tightness arises because to obtain a tightly concentrated Dirichlet distribution on $\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})$, the concentration parameter $K=\sum_{i=1}^{d_{\mathrm{vot}}}\alpha_i$ must grow linearly with the dimension. In fact, even a uniform distribution (which will be less peaked than our final posterior) has $\sum_{i=1}^{d_{\mathrm{vot}}}\alpha_i = d_{\mathrm{vot}}$, so the de-randomisation step below, which is tighter for higher K, is effectively very cheap in higher dimensions.

Theorem 7.3. *Let $\boldsymbol{\theta}\in\Delta_{d_{\mathrm{vot}}-1}$ and $K>0$. Then for any $\gamma>0$ and (x,y),*

$$\ell_0(\mathrm{MV}_{\boldsymbol{\theta}},(x,y)) \leq \mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(K\boldsymbol{\theta})} \ell_\gamma(\mathrm{MV}_{\boldsymbol{\xi}},(x,y)) + e^{-(K+1)\gamma^2},$$

$$\mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(K\boldsymbol{\theta})} \ell_\gamma(\mathrm{MV}_{\boldsymbol{\xi}},(x,y)) \leq \ell_{2\gamma}(\mathrm{MV}_{\boldsymbol{\theta}},(x,y)) + e^{-(K+1)\gamma^2}.$$

For our proofs we first recall the aggregation property of the Dirichlet distribution: if $(\xi_1,\ldots,\xi_{d_{\mathrm{vot}}})\sim\mathrm{Dir}((\alpha_1,\ldots,\alpha_{d_{\mathrm{vot}}}))$, then $(\xi_1,\ldots,\xi_{d_{\mathrm{vot}}-1}+\xi_d)\sim\mathrm{Dir}((\alpha_1,\ldots,\alpha_{d_{\mathrm{vot}}-1}+\alpha_{d_{\mathrm{vot}}}))$. We further note the following crucial concentration-of-measure result. The aforementioned lack of dimensionality in Theorem 7.3 is possible because Theorem 7.4 depends only on $\sum_{i=1}^{d_{\mathrm{vot}}}\alpha_i$, and this value is unchanged by aggregation, which avoids the dimension dependence that would otherwise be introduced by the requirement $\|\boldsymbol{u}\|_2=1$ below.

Theorem 7.4 (Marchal and Arbel, 2017). *Let* $\boldsymbol{X} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, $t > 0$, *and* $\boldsymbol{u} \in \mathbb{R}^d$ *with* $\|\boldsymbol{u}\|_2 = 1$. *Then*

$$\mathop{\mathbb{P}}_{\boldsymbol{X}}\{\boldsymbol{u}\cdot(\boldsymbol{X}-\mathbb{E}\boldsymbol{X}) > t\} \le \exp\left(-2\left(\sum_{i=1}^{d}\alpha_i + 1\right)t^2\right).$$

Proof of Theorem 7.1 and Theorem 7.2. The proof of our main results is completed by applying the PAC-Bayes bound Theorem 3.12 with the γ-margin loss to a Dirichlet prior and posterior with parameters $\boldsymbol{\beta}$ and $K\boldsymbol{\theta}$ respectively. Substituting the first part of Theorem 7.3 gives the first part of Theorem 7.2, and additionally substituting the second part and re-scaling $\gamma \to \gamma/2$ gives Theorem 7.1.

For the second part of Theorem 7.2, define correct labels $c = \{i : h_i(x) = y\}$ for fixed (x, y) so that the (randomised) weight of correct labels is $W := \sum_{i\in c}\xi_i \sim \mathrm{Beta}\left(\sum_{i\in c}\alpha_i, \sum_{i\notin c}\alpha_i\right)$ by the aggregation property of the Dirichlet distribution. Then

$$\mathop{\mathbb{E}}_{\boldsymbol{\xi}}\ell_\gamma(\mathrm{MV}_{\boldsymbol{\xi}}, (x, y)) \le \mathop{\mathbb{E}}_{\boldsymbol{\xi}}\left\{W \le \frac{1+\gamma}{2}\right\} = I_{\frac{1+\gamma}{2}}\left(\sum_{j\in c}\alpha_j, \sum_{i\notin c}\alpha_j\right)$$

using $\mathbf{1}\left\{a_i - \max_{j\neq i} a_j \le \gamma\right\} \le \mathbf{1}\left\{a_i - \sum_{j\neq i} a_j \le \gamma\right\} = \mathbf{1}\left\{a_i \le \frac{1+\gamma}{2}\right\}$ for $\boldsymbol{a} \in \Delta_{d_{\mathrm{out}}-1}$ (with equality for $d_{\mathrm{out}} = 2$ classes), and that $I_z(a, b)$ is the CDF of a Beta distribution with parameters (a, b). □

Proof of Theorem 7.3. Define $\gamma_2 > \gamma_1$ such that $\gamma := \gamma_2 - \gamma_1$, and $\boldsymbol{\alpha} = K\boldsymbol{\theta}$. From the trivial inequality $\mathbf{1}_{x\in A} - \mathbf{1}_{x\in B} \le \mathbf{1}_{x\in A}\mathbf{1}_{x\notin B}$ we derive

$$\begin{aligned}
\Delta &:= \ell_{\gamma_1}(\mathrm{MV}_{\boldsymbol{\theta}}, (x, y)) - \mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}\ell_{\gamma_2}(\mathrm{MV}_{\boldsymbol{\xi}}, (x, y)) = \mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\mathbf{1}_{M(\boldsymbol{\theta},x,y)\le\gamma_1} - \mathbf{1}_{M(\boldsymbol{\xi},x,y)\le\gamma_2}] \\
&\le \mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\mathbf{1}_{M(\boldsymbol{\theta},x,y)\le\gamma_1}\mathbf{1}_{M(\boldsymbol{\xi},x,y)>\gamma_2}] \le \mathop{\mathbb{E}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\mathbf{1}_{M(\boldsymbol{\xi},x,y)-M(\boldsymbol{\theta},x,y)>\gamma}] \\
&= \mathop{\mathbb{P}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}\left\{\sum_{i:h_i(x)=y}\xi_i - \max_{j'\neq y}\sum_{i:h_i(x)=k'}\xi_i - \sum_{i:h_i(x)=y}\theta_i + \max_{j\neq y}\sum_{i:h_i(x)=k}\theta_i > \gamma\right\} \\
&\le \mathop{\mathbb{P}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}\left\{\sum_{i:h_i(x)=y}\xi_i - \sum_{i:h_i(x)=k}\xi_i - \sum_{i:h_i(x)=y}\theta_i + \sum_{i:h_i(x)=k}\theta_i > \gamma\right\}
\end{aligned}$$

where in the last inequality we set $k = \mathrm{argmax}_{k\neq y}\sum_{i:h_i(x)=k}\theta_i$, and use that $\max_j \sum_{i:h_i(x)=j}\theta_i - \max_j \sum_{i:h_i(x)=j}\xi_i \le \max_j \sum_{i:h_i(x)=j}\theta_i - \sum_{i:h_i(x)=k}\xi_i$ for any k. We rewrite the above in vector form (with inner product denoted $\boldsymbol{u}\cdot\boldsymbol{v}$) as

$$\Delta \le \mathop{\mathbb{P}}_{\boldsymbol{\xi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}\left\{\underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\\0\end{bmatrix}}_{\boldsymbol{u}} \cdot \left(\underbrace{\begin{bmatrix}\sum_{i:h_i(x)=y}\xi_i\\ \sum_{i:h_i(x)=k}\xi_i\\ \sum_{i:h_i(x)\notin\{k,y\}}\xi_i\end{bmatrix}}_{\tilde{\boldsymbol{\xi}}} - \underbrace{\begin{bmatrix}\sum_{i:h_i(x)=y}\theta_i\\ \sum_{i:h_i(x)=k}\theta_i\\ \sum_{i:h_i(x)\notin\{k,y\}}\theta_i\end{bmatrix}}_{\mathbb{E}\tilde{\boldsymbol{\xi}}}\right) > \frac{1}{\sqrt{2}}\gamma\right\}$$

$$= \mathop{\mathbb{P}}_{\tilde{\boldsymbol{\xi}} \sim \mathrm{Dir}(\tilde{\boldsymbol{\alpha}})} \left\{ \boldsymbol{u} \cdot (\tilde{\boldsymbol{\xi}} - \mathbb{E}\tilde{\boldsymbol{\xi}}) > \frac{1}{\sqrt{2}}\gamma \right\}$$

where by the aggregation property of the Dirichlet distribution $\tilde{\boldsymbol{\xi}} \sim \mathrm{Dir}(\tilde{\boldsymbol{\alpha}})$ with

$$\tilde{\boldsymbol{\alpha}} := \left[\sum_{i:h_i(x)=y} \alpha_i, \sum_{i:h_i(x)=k} \alpha_i, \sum_{i:h_i(x)\notin\{k,y\}} \alpha_i \right]^T .$$

Applying Theorem 7.4 we obtain $\Delta \leq e^{-(\sum_i \tilde{\alpha}_i + 1)\gamma^2} = e^{-(\sum_{i=1}^{d_{\mathrm{vot}}} \alpha_i + 1)\gamma^2}$. This gives the first inequality by setting $\gamma_1 = 0, \gamma_2 = \gamma$. Setting $\gamma_1 = \gamma, \gamma_2 = 2\gamma$ and swapping $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ gives an almost identical proof (with some signs reversed) of the second inequality. □

7.4 Empirical evaluation

In this section we empirically validate our results against existing PAC-Bayesian and margin bounds on several classification datasets from UCI (Dua and Graff, 2017), LIBSVM[3] and Zalando (Xiao et al., 2017). Since our main result in Theorem 7.1 is not associated with any particular algorithm, we use $\boldsymbol{\theta}$ outputted from PAC-Bayes-derived algorithms to evaluate this result against other margin bounds (Fig. 7.1) and PAC-Bayes bounds (Fig. 7.2). We then compare optimisation of our secondary result Theorem 7.2 with optimising those PAC-Bayes bounds directly (Fig. 7.3). All generalisation bounds given are evaluated with a probability $1-\delta$=0.95. Further details not provided here including tabulated results, description of datasets, training mechanisms and compute are provided in Section 7.C. The code for reproducing the results is available at https://github.com/vzantedeschi/dirichlet-margin-bound.

Strong and weak voters. Similarly to Zantedeschi et al. (2021) we consider both using data-independent and data-dependent voters. This brings our experimental setup in line with a common workflow for machine learning practicioners: the training set is sub-divided into a set for training several different strong algorithms, and a second set on which the weightings of these are optimised. More specifically, the weak voter setting, used only for binary classification, uses axis-aligned decision stumps (denoted *stumps*), with thresholds evenly spread over the input space (6 per feature and per class). The stronger voters (denoted *rf*) are learned from half of the training data, while the other half is used for evaluating and optimising the different generalisation bounds (note this reduces m). These take the form of random forests (Breiman, 2001) of M=10 trees optimising Gini impurity score on $\frac{n}{2}$ bagged samples and $\sqrt{d_{\mathrm{in}}}$ drawn features for each tree, with unbounded maximal depth.

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Optimising γ and K in bounds. In reporting margin bounds we optimise over a grid of margin γ values in $(0,1)$, and additionally over K for Theorem 7.1. Since Theorem 7.1 and Eq. (7.3) as stated require a fixed margin, we apply a union bound over the values in the grid, replacing δ in these bounds with δ/N where N is the number of grid points.

Existing PAC-Bayes bounds. We compare to state-of-the-art PAC-Bayesian bounds (and derived algorithms) for weighted majority vote classifiers: the First Order (Langford and Shawe-Taylor, 2003), the Second Order (Masegosa et al., 2020), Binomial (Lacasse et al., 2010) (with the number of voters set to 100) and the two Chebyshev-Cantelli-based (Wu et al., 2021) empirical bounds from categorical-type Gibbs classifiers with parameter $\boldsymbol{\theta}$, and we refer to these as *FO*, *SO*, *Bin*, *CCPBB* and *CCTND* respectively (more details are given in Section 7.C). We denote by *f2* the factor two bound derived in Zantedeschi et al. (2021, Annex A.4) from Dirichlet majority votes. All prior distributions for PAC-Bayes bounds, including ours, are set to uniform. We also refer by the same names to the outputs of optimising these bounds with stochastic gradient descent; details on training and initialisation are given in Section 7.C.

Description of figures. In Fig. 7.1 we compare Theorem 7.1 with the existing margin bound of Eq. (7.2) and the improved Biggs and Guedj (2022a) bound given in Eq. (7.3) (which in this thesis is Theorem 5.6, proved in Chapter 5). Since Eq. (7.3) is strictly better than the original result and the latter was vacuous in almost all cases considered (see Section 7.B), we do not include it. All datasets are for binary classification as the existing results only cover this case, and the $\boldsymbol{\theta}$ values considered are the outputs of either the FO- or f2-optimisation using either the weak or the strong voters described above. Figure 7.2 extends this evaluation of Theorem 7.1 to improve generalisation results, by applying it to the models optimised with the PAC-Bayes bounds *FO*, *SO*, *Bin* and *f2* as objective. In this case, we consider both binary and multi-class datasets. In Fig. 7.3 we directly compare the outputs of optimising state-of-the-art PAC-Bayesian bounds with our optimisation-ready variant result Theorem 7.2. These experiments were carried out on strong voters, as standard in the literature (*e.g.* Lorenzen et al., 2019; Masegosa et al., 2020; Wu et al., 2021).

7.5 Discussion and conclusion

We observe overall that in many cases the existing margin and PAC-Bayes bounds are insufficient to explain the generalisation observed, while our new bound is consistently tight, and sometimes sharp (*i.e.* it approaches the true test error).

Figure 7.1 demonstrates that existing margin bounds can be insufficient to explain the generalisation observed, which could be construed as a null result for the "margins theory".

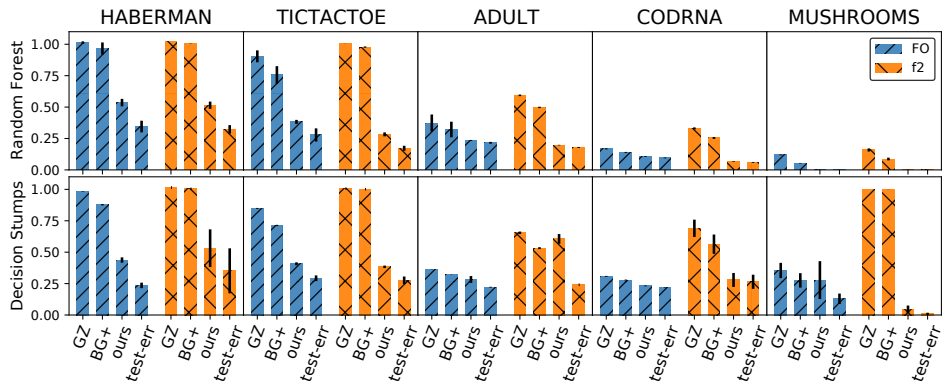

Figure 7.1: Theorem 7.1 (**ours**) compared with the margin bounds of Eq. (7.3) (**BG+**), Eq. (7.2) (**GZ**), and the test error. Settings are *rf* (first row) and *stumps* (second row) on the given datasets, with $\boldsymbol{\theta}$ output by optimising either *FO* or *f2* (first and second column groupings respectively).

However, our new bound obtains empirically very sharp results in almost all cases, reaffirming to the theory. Note that due to the non-convexity of our bound, the reported values are local minima and can potentially be improved by applying a thorougher search for the optimal K, still giving a similarly valid bound. For instance, simply by enlarging the search space for K our bound drops to 0.36 ± 0.10 on *ADULT* with decision stumps as voters, beating existing bounds also in this setting. Unlike the existing results, $\boldsymbol{\theta}$ also arises in the complexity (KL divergence) term and so the bound is not equally tight for every $\boldsymbol{\theta}$ at fixed margin loss. Further examination of this property could add additional nuance and perspective to the theory.

When comparing to existing PAC-Bayes bounds in Fig. 7.2, remarkably Theorem 7.1 is *always* tighter than just using the bound which is being optimised. We speculate that this arises partially due to the irreducible factors appearing in those bounds; for example the *FO* or *f2* bounds can never be tighter than twice the train loss of the associated Gibbs classifier, while ours has no such limitation. This result is quite valuable as it demonstrates that Theorem 7.1 can be readily used in an algorithm-free manner: the choice of learning algorithm is up to the practitioner, but the bound will then often provide an excellent guarantee on the obtained weights $\boldsymbol{\theta}$.

Finally, in Fig. 7.3, our optimisation-friendly variant bound Theorem 7.2 is seen to be competitive in terms of test error while giving an improved-or-equal final bound on all datasets. When considering the less-common setting of binary stumps (see Section 7.C) we found that sometimes this objective converged to a sub-optimal local minimum. We speculate that this arises due to the highly non-convex nature of the objective combined with a strong K-inflating gradient signal from the $\mathcal{O}(e^{-K\gamma^2})$ term. Thus future work to improve these

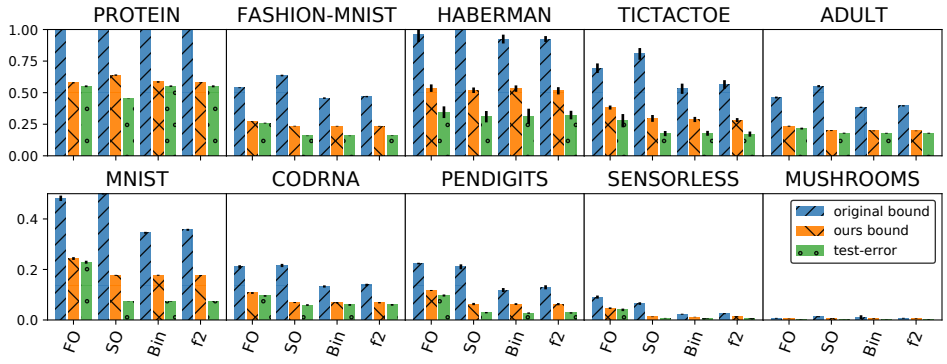

Figure 7.2: Theorem 7.1 (*our bound*) compared with the bounds of *FO*, *SO*, *Bin* or *f2* (*original bound*), and test errors. For each column grouping, $\boldsymbol{\theta}$ is the output from optimising the corresponding PAC-Bayes bound (as named underneath) for *rf* on the given dataset. The blue column is the final value of the bound used as objective, the green is the test error, and the orange is the value of our bound when $\boldsymbol{\theta}$ is plugged into it (so that our bound is not used as an objective here).

results even further could start with the use of the quasi-convex small-kl relaxation from Thiemann et al. (2017). We note however that this is overall less important than our main results, as both our bounds are still extremely tight when used in an algorithm-free way and applied to the output of another algorithm as discussed above.

Overall, we note that in many cases (a majority in Fig. 7.2) our main bound of Theorem 7.1 is very close to the test set bound and thus cannot actually be improved any further, with the problem of providing sharp guarantees based on the training data alone effectively solved in many cases.

Conclusion. We obtain empirically very strong generalisation bounds for voting classifiers using margins. We believe these are highly relevant to the community, since voting-based classifiers and margin-maximising algorithms are among the most popular and influential in machine learning. Dirichlet majority votes have already obtained excellent results in the stochastic setting (Zantedeschi et al., 2021), but our new result in Theorem 7.3 showing they are well-approximated by their mean should open new directions in the more conventional deterministic setting.

Our results also have practical relevance: for example, in the strong voter machine learning workflow described above, instead of setting data aside as a test set, this data can be freed up to learn even stronger voters, since a strong out-of-sample ensemble guarantee can still be provided even *without* a test set.

In future work we hope to expand these results further to other (non-majority) voting

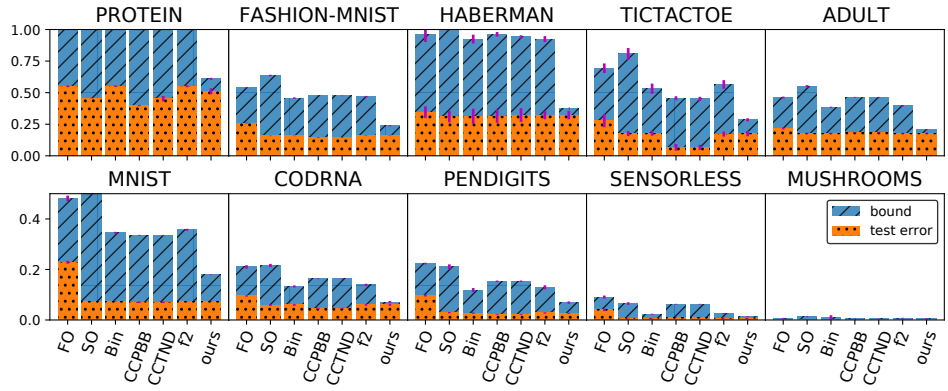

Figure 7.3: Theorem 7.2 (**ours**) as optimisation objective compared to other PAC-Bayes results (*FO*, *SO*, *Bin*, *CCPBB* and *CCTND*) as objectives in the *rf* setting. For each objective the test error and bound associated with the objective is shown.

schemes like those with score-output voters (as in *e.g.* Schapire et al., 1998), and ensembles of voters with finite VC dimension.

7.A Properties of the Dirichlet distribution

The Dirichlet measure has probability density function w.r.t. Lebesgue measure given by:

$$f(x_1,\dots,x_d;\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})}\prod_{i=1}^{d} x_i^{\alpha_i-1}$$

where $B(\boldsymbol{\alpha})$ is the multivariate Beta function,

$$B(\boldsymbol{\alpha}) := \frac{\prod_{i=1}^{d}\Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{d}\alpha_d)}.$$

The mean of a Dirichlet is $\mathbb{E}_{\xi\sim\mathrm{Dir}(\boldsymbol{\alpha})}\boldsymbol{\xi} = \boldsymbol{\alpha}/\sum_{i=1}^{d}\alpha_i$. The KL divergence between two Dirichlet distributions is the following, given in *e.g.* Zantedeschi et al. (2021):

$$\mathbb{D}_{\mathrm{Dir}}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \log\frac{B(\boldsymbol{\beta})}{B(\boldsymbol{\alpha})} + \sum_{i=1}^{d}(\alpha_i-\beta_i)(\psi(\alpha_i)-\psi(\alpha_0)) = \log B(\boldsymbol{\beta}) - \mathbb{H}_{\mathrm{Dir}}(\boldsymbol{\alpha}).$$

7.B Additional details on margin bounds

We firstly note that the definition of the margin given in Gao and Zhou (2013) and Biggs and Guedj (2022a) is slightly different from our own in that it allows for negative weights. Biggs and Guedj (2022a) uses a dimension doubling trick to allow these negative weights (as they consider only the binary case), which we remove in Eq. (7.3) to replace the factor $\log(2d_{\mathrm{vot}})$ by $\log d_{\mathrm{vot}}$. This definition leads however to identical definitions in the case of the majority vote for binary classification, where $w \geq 0$ and $\|w\|_1 = 1$.

Technical, the above papers consider prediction functions like $F_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{d_{\mathrm{vot}}}\theta_i h_i(x)$ with output set $\mathcal{Y} = \{+1,-1\}$. The functions $h_i(x)$ can be positive or negative. The margin is defined as as $yF_{\boldsymbol{\theta}}(x)$. We translate this into our equivalent formulations as

$$yF_{\boldsymbol{\theta}}(x) = y\left(\sum_{i:h_i(x)=1}\theta_i - \sum_{i:h_i(x)=-1}\theta_i\right) = \sum_{i:h_i(x)=y}\theta_i - \sum_{i:h_i(x)=-y}\theta_i \tag{7.5}$$

which is the binary margin as we define it in Section 7.1.1. Thus $\ell_\gamma(\mathrm{MV}_{\boldsymbol{\theta}},(x,y)) = \mathbf{1}_{M(\boldsymbol{\theta},x,y)\leq\gamma} = \mathbf{1}_{yF_{\boldsymbol{\theta}}\geq\gamma}$.

Here we also note the original result from Biggs and Guedj (2022a) that is adapted and improved in Eq. (7.3) and Chapter 5; since this is obtained by applying an upper bound to the inverse small-kl and an additional step, it is strictly looser than the result we give in Eq. (7.3). We compare the these different margin bounds as a function of γ below.

Theorem 7.5. *For any margin $\gamma > 0$, $\delta \in (0,1)$, sample size $m \in \mathbb{N}$, each of the following results holds with probability at least $1-\delta$ over the sample $S \sim \mathcal{D}^m$ simultaneously for any $\boldsymbol{\theta} \in \Delta_{d_{\mathrm{vot}}-1}$,*

$$\mathcal{L}_0(\mathrm{MV}_{\boldsymbol{\theta}}) \leq \widehat{\mathcal{L}}_\gamma(\mathrm{MV}_{\boldsymbol{\theta}}) + \sqrt{\frac{C}{m}\cdot\widehat{\mathcal{L}}_\gamma(\mathrm{MV}_{\boldsymbol{\theta}})} + \frac{C+\sqrt{C}+2}{m}, \tag{7.6}$$

where $C := 2\log(2/\delta) + 19\gamma^{-2}\log d_{\mathrm{vot}}\log m$.

7.B.1 Further improvement to Eq. (7.3)

A question which naturally arises from looking at the proof of Eq. (7.3) and Theorem 7.5 is whether we can do better by choosing T in a more optimal way, rather than just setting it to $\lceil 8\gamma^{-2}\log m\rceil$. We thus prove a bound here which is valid for the optimal choice of T; in practice this is seen to be slightly tighter than Eq. (7.3), although the improvement from Theorem 7.5 to that result is far greater.

For any $\boldsymbol{\theta}\in\Delta_{d_{\text{vot}}-1}$ with probability at least $1-\delta$ over the sample,

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \le \inf_{T\in\mathbb{N}_+}\left[\text{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}})+e^{-\frac{1}{8}T\gamma^2}, \frac{T\log d_{\text{vot}}+\log\frac{m}{\delta}}{m}\right)+e^{-\frac{1}{8}T\gamma^2}\right] \tag{7.7}$$

A slightly weaker version of this result, with an extra $m^{-1}\log(2\sqrt{m})$ term, could be proved from

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \le \text{kl}^{-1}\left(\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}})+e^{-\frac{1}{8}T\gamma^2}, \frac{T\log d_{\text{vot}}+\log\frac{2\sqrt{m}}{\delta}}{m}\right)+e^{-\frac{1}{8}T\gamma^2},$$

which is an intermediate step in the proof of Eq. (7.3). We note however that the optimal T depends on the data only through $\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}})\in\{0,m^{-1},2m^{-1},\dots,1\}$. The last possibility gives a trivial bound. A union bound over the m non-vacuous possibilities gives Eq. (7.7) with the extra logarithmic factor.

In order to remove this term, we use a slightly more sophisticated argument applied to a different PAC-Bayes bound (Catoni's bound, Theorem 3.9). We recall the function $\Phi_C(p)$ which appears in this bound, and that it can be related to the small-kl inversion via Theorem 3.4.

Proof of Eq. (7.7). Following the proof steps of Theorems 5.1 and 5.2 with PAC-Bayes bound Theorem 3.9, then using the 1-sub-Gaussianity of the predictors as in the proof of Theorem 5.6 we obtain for any data-independent $C>0, T\in\mathbb{N}_+, \gamma>0$ that

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}})-e^{-\frac{1}{8}T\gamma^2} \le \Phi_C^{-1}\left(\frac{k}{m}+e^{-\frac{1}{8}T\gamma^2}+\frac{T\log d_{\text{vot}}+\log\frac{1}{\delta}}{Cm}\right).$$

where $k:=m\widehat{\mathcal{L}}_\gamma(\text{MV}_{\boldsymbol{\theta}})$ is the number of margin errors.

Since the only quantity on the left hand side in this bound unknown before we see data is the value of k, there exists a C_k dependent on the value of k that optimises the bound, and a T_k that depends on this pair. Since there are only m such values giving non-vacuous bounds ($k=m$ is trivially vacuous), we can apply a union bound over all these bounds with $\delta=\delta/m$ to give the following with probability $\ge 1-\delta$:

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \le \min_{T\in\mathbb{N}_+}\min_{C>0}\left[e^{-\frac{1}{8}T\gamma^2}+\Phi_C^{-1}\left(\frac{k}{m}+e^{-\frac{1}{8}T\gamma^2}+\frac{T\log d_{\text{vot}}+\log\frac{m}{\delta}}{Cm}\right)\right].$$

Applying the inversion of Theorem 3.4 gives the result. □

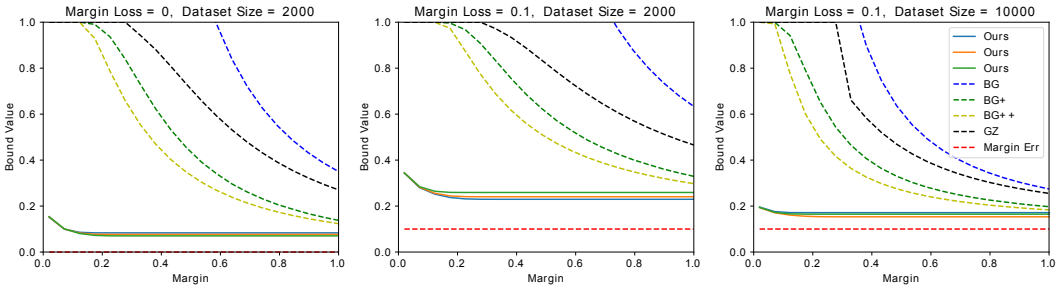

Figure 7.4: Values of different bounds versus margin at margin error $\widehat{\mathcal{L}}_\gamma$ (0 or 0.2). Dimension $d_{\text{vot}} = 100$, probability $\delta = 0.5$ and dataset size m (2000 or 10000) are also fixed. The bounds are Theorem 7.1 (**ours**) with three different samples $\boldsymbol{\theta} \sim \text{Uniform}(\Delta_{d_{\text{vot}}-1})$, compared with the margin bounds of Theorem 7.5 (**BG**), Eq. (7.3) (**BG+**), Eq. (7.7) (**BG++**), Eq. (7.2) (**GZ**), and the margin error $\widehat{\mathcal{L}}_\gamma$.

7.B.2 Comparison of margin bounds

In Fig. 7.4 we compare the various bounds given above in a non-experimental way, fixing the margin loss $\widehat{\mathcal{L}}_\gamma$ to a particular value and seeing how the bounds change if that value of the loss is achieved for different values of the margin $\gamma \in (0,1)$. Since (uniquely among the bounds), the value of $\boldsymbol{\theta}$ appears in our bound Theorem 7.1, we show three different sampled possible values, drawn uniformly from the simplex.

The results for "categorical"-based bounds demonstrate that the refined bounds Eqs. (7.3) and (7.7) are much tighter that the result as given in Theorem 7.5 by Biggs and Guedj (2022a). Both these refinements are also tighter than Eq. (7.2) from Gao and Zhou (2013). We used Eq. (7.3) in the main paper because it is closer to an exiting result (as it appears in the proof from Biggs and Guedj, 2022a), and is not much worse than the refinement Eq. (7.7), particularly when compared to our far stronger new result Theorem 7.1.

This figure also shows that, at least for some values of $\boldsymbol{\theta}$, this new bound can be far tighter than all the existing bounds. One interesting facet of this is that the bound is improved very little for γ above a certain point, quite a different behaviour to the other bounds. Empirically this was seen too in our other experiments, with the optimised γ often being quite small. Of course, for some values of $\boldsymbol{\theta}$ this bound will be weaker, but we observe the same kind of results in our main experimental results, where this is a learned value.

7.C Additional experimental details and evaluations

Dataset descriptions. We provide the description of the classification datasets considered in our empirical evaluation.

***Haberman* (UCI)** prediction of survival of $n = 306$ patients who had undergone surgery

from $d_{\text{in}} = 3$ anonymized features.

***TicTacToe* (UCI)** determination of a win for player x at TicTacToe game of any of the $n = 958$ board configurations ($d_{\text{in}} = 9$ categorical states).

***Mushrooms* (UCI)** prediction of edibility of $n = 8,124$ mushroom sample, given their $d_{\text{in}} = 22$ categorical features describing their aspect.

***Adult* (LIBSVM a1a)** determining whether a person earns more than 50K a year ($n = 32,561$ people and $d_{\text{in}} = 123$ binary features).

***CodRNA* (LIBSVM)** detection of non-coding RNAs among $n = 59,535$ instances and from $d_{\text{in}} =$ features.

***Pendigits* (UCI)** recognition of hand-written digits (10 classes, $d_{\text{in}} = 9$ features and $n = 12,992$).

***Protein* (LIBSVM)** $d_{\text{in}} = 357$ features, $n = 24,387$ instances and 3 classes.

***Sensorless* (LIBSVM)** prediction of motor condition ($n = 58,509$ instances and 11 classes), with intact and defective components, from $d_{\text{in}} = 48$ features extracted from electric current drive signals.

***MNIST* (LIBSVM)** prediction of hand-written digits ($n = 70,000$ instances and 10 classes) from $d_{\text{in}} = 28 \times 28$ gray-scale images.

***Fashion-MNIST* (Zalando)** prediction of cloth articles ($n = 70,000$ instances and 10 classes) from $d_{\text{in}} = 28 \times 28$ gray-scale images.

In all experiments, we convert all categorical features to numerical using an ordinal encoder and we standardize all features using the statistics of the training set.

Baseline descriptions. We report the generalization bounds of the literature used for training weighted majority vote classifiers in our comparison. We additionally note: $\text{Cat}(\boldsymbol{\theta})$ the categorical distribution over the base classifiers (with θ_i the weight associated to voter $h_i \in \mathcal{H}_{\text{base}}$), and $\mathbb{D}_{\text{Cat}}(\boldsymbol{\theta}, \boldsymbol{\pi})$ the KL-divergence between two categorical distribution with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$; $\ell_{\text{TND}}(h, h', x, y) := \mathbf{1}\{h(x) \neq y\}\mathbf{1}\{h'(x) \neq y\}$ the tandem loss proposed in Masegosa et al. (2020) and $\widehat{\mathcal{L}}_{\text{TND}}(h, h') := \mathbb{E}_{(x,y)\sim\text{Uniform}(S)}\, \ell_{\text{TND}}(h, h', x, y)$ its in-sample estimate; $\ell_{\text{Bin}}(\boldsymbol{\theta}, T, x, y) := \sum_{k=\frac{T}{2}}^{T} \binom{T}{k} (1-\theta_y)^k \theta_y^{(T-k)}$ for $\theta_y = \sum_{h_i(x)=y}$, which is the probability that among T voters randomly drawn from $\text{Cat}(\boldsymbol{\theta})$ at least $\frac{T}{2}$ of them are incorrect, as defined in Lacasse et al. (2010).

- First Order (FO, Langford and Shawe-Taylor, 2003):
  For any $\mathcal{D} \in \mathscr{P}(\mathcal{X} \times \mathcal{Y})$, $m \in \mathbb{N}_+$, $\delta \in (0,1)$, and prior $\boldsymbol{\pi} \in \Delta_{d_{\text{vot}}-1}$, with probability at least $1-\delta$ over the sample $S \sim \mathcal{D}^m$ simultaneously for every $\boldsymbol{\theta} \in \Delta_{d_{\text{vot}}-1}$,

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \leq 2\,\text{kl}^{-1}\left(\mathop{\mathbb{E}}_{h \sim \text{Cat}(\boldsymbol{\theta})} \widehat{\mathcal{L}}_0(h),\ \frac{\mathbb{D}_{\text{Cat}}(\boldsymbol{\theta}, \boldsymbol{\pi}) + \log \frac{2\sqrt{m}}{\delta}}{m} \right).$$

- Second Order (SO, Masegosa et al., 2020):
  For any $\mathcal{D} \in \mathscr{P}(\mathcal{X} \times \mathcal{Y})$, $m \in \mathbb{N}_+$, $\delta \in (0,1)$, and prior $\boldsymbol{\pi} \in \Delta_{d_{\text{vot}}-1}$, with probability at least $1-\delta$ over the sample $S \sim \mathcal{D}^m$ simultaneously for every $\boldsymbol{\theta} \in \Delta_{d_{\text{vot}}-1}$,

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \leq 4\,\text{kl}^{-1}\left(\mathop{\mathbb{E}}_{h \sim \text{Cat}(\boldsymbol{\theta}), h' \sim \text{Cat}(\boldsymbol{\theta})} \widehat{\mathcal{L}}_{\text{TND}}(h, h'),\ \frac{2\mathbb{D}_{\text{Cat}}(\boldsymbol{\theta}, \boldsymbol{\pi}) + \log \frac{2\sqrt{m}}{\delta}}{m} \right).$$

- Binomial (Bin, Lacasse et al., 2010):
  For any $\mathcal{D} \in \mathscr{P}(\mathcal{X} \times \mathcal{Y})$, $m \in \mathbb{N}_+$, $T \in \mathbb{N}_+$, $\delta \in (0,1)$, and prior $\boldsymbol{\pi} \in \Delta_{d_{\text{vot}}-1}$, with probability at least $1-\delta$ over the sample $S \sim \mathcal{D}^m$ simultaneously for every $\boldsymbol{\theta} \in \Delta_{d_{\text{vot}}-1}$,

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \leq 2\,\text{kl}^{-1}\left(\mathop{\mathbb{E}}_{(x,y) \sim \text{Uniform}(S)} \ell_{\text{Bin}}(\boldsymbol{\theta}, T, x, y),\ \frac{T\mathbb{D}_{\text{Cat}}(\boldsymbol{\theta}, \boldsymbol{\pi}) + \log \frac{2\sqrt{m}}{\delta}}{m} \right).$$

- Chebyshev-Cantelli tandem loss bound (CCTND, Wu et al., 2021, Theorem 12);

- Chebyshev-Cantelli tandem loss bound with an offset (CCPBB, Wu et al., 2021, Theorem 15);

- Dirichlet Factor-Two (f2, Zantedeschi et al., 2021):
  For any $\mathcal{D} \in \mathscr{P}(\mathcal{X} \times \mathcal{Y})$, $m \in \mathbb{N}_+$, $\delta \in (0,1)$, and prior $\boldsymbol{\beta} \in \mathbb{R}_+^{d_{\text{vot}}}$, with probability at least $1-\delta$ over the sample $S \sim \mathcal{D}^m$ simultaneously for every $\boldsymbol{\theta} \in \Delta_{d_{\text{vot}}-1}$ and $K > 0$,

$$\mathcal{L}_0(\text{MV}_{\boldsymbol{\theta}}) \leq 2\,\text{kl}^{-1}\left(\mathop{\mathbb{E}}_{\boldsymbol{\xi} \sim \text{Dir}(K\boldsymbol{\theta})} \widehat{\mathcal{L}}(\text{MV}_{\boldsymbol{\xi}}),\ \frac{\mathbb{D}_{\text{Dir}}(K\boldsymbol{\theta}, \boldsymbol{\beta}) + \log \frac{2\sqrt{m}}{\delta}}{m} \right).$$

Optimisation of PAC-Bayesian bounds. To optimize the baselines *CCPBB* and *CCTND*, we rely on the code released by its authors [4], with the Gradient Descent option and building random forests as described in our main text. When optimising the PAC-Bayesian bounds *FO*, *SO*, *Bin*, *f2* and ours, we initialize $\boldsymbol{\theta}$'s to be uniform, *i.e.* $\theta_i = 1/d_{\text{vot}}$, and $K = 2$. We then optimise the posterior parameters of the method ($\alpha = K\theta$ for Dirichlet, and θ for

[4] https://github.com/StephanLorenzen/MajorityVoteBounds/tree/44cec987865ddce01cd27076019394538cee85ca/NeurIPS2021

Categorical distributions) with the Adam optimiser (Kingma and Ba, 2014) with running average coefficients $(0.9, 0.999)$, batch size equal to 100 and learning rate set to 0.1. All methods are run for a maximum of 100 epochs with patience of 25 epochs for early stopping and a learning rate scheduler reducing it by a factor of 10 with 2 epochs patience.

At each run of an algorithm, we randomly split a dataset into training and test sets of sizes $80\% - 20\%$ respectively, and optimise/evaluate the bounds only with the half of the training set that was not used for learning the voters (in the case of data-dependent ones). Note that we do not make use of a validation set, as we use the risk certificates as estimate of the test error for model selection. Finally, we report the value of Seeger's "small-kl" bound of Theorem 3.12, even when a different type of bound has been optimised (*e.g.* for the *CCPBB* and *CCTND* baselines), and we average all results over 5 different trials.

Margin bound comparison. Given a pre-trained model, hence fixed θ and initial K_{init} (which is different from 1. only for the models trained via Dirichlet bounds), we search for its optimal risk certificate by evaluating a given bound at $1,000$ values of γ, spaced evenly on a log scale with base 10 and in the interval $[10^{-4}, 0.5)$. For our margin bound, for each of these γ values we also optimise $K \in [K_{init}, K_{init}\, 2^{16}]$ using the golden-section search technique to obtain the tightest upper bound. Notice that this does not add significant computational overhead to the search. Also for these experiments, the bounds are evaluated with the portion of training data that was not used for learning the voters.

Compute. All experiments were run on a virtual machine with 16 vCPUs and $128Gb$ of RAM.

7.C.1 Additional results

In Fig. 7.5, Fig. 7.6 and Fig. 7.7 we report the results from Fig. 7.1, Fig. 7.2 and Fig. 7.3 in the main text. Here we deploy a different scale per dataset so that they can be easily read, also when the bounds and test errors are very small. Additionally, in Fig. 7.8 we provide the test errors and risk certificates obtained by optimising the generalization bounds with decision stumps as voters. Although our certificates are always the tightest, we found that in some cases our method converges to sub-optimal solutions. We speculate that this arises due to the highly non-convex nature of the objective combined with a strong K-inflating gradient signal from the $\mathcal{O}(e^{-K\gamma^2})$ term. Thus future work to improve these results even further could start with the use of the quasi-convex small-kl relaxation from Thiemann et al. (2017). We note however that this is overall less important than our main results, as both our bounds are still extremely tight when used in an algorithm-free way and applied to the output of another algorithm.

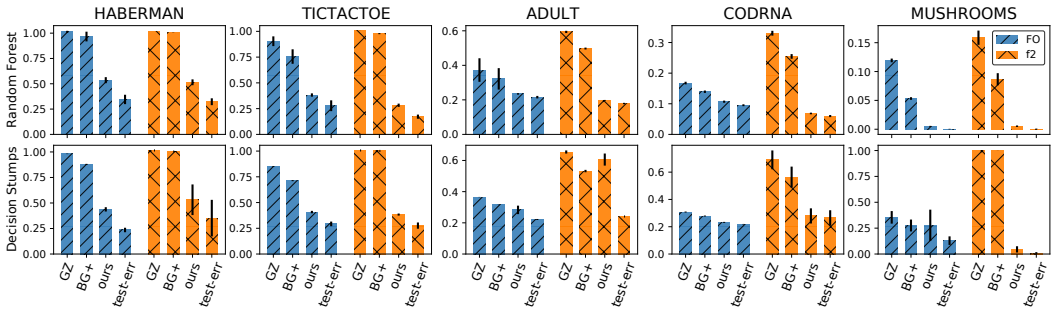

Figure 7.5: Theorem 7.1 (**ours**) compared with the margin bounds of Eq. (7.3) (**BG+**), Eq. (7.2) (**GZ**), and the test error. Settings are *rf* (first row) and *stumps* (second row) on the given datasets, with $\boldsymbol{\theta}$ output by optimising either *FO* or *f2* (first and second column groupings respectively).

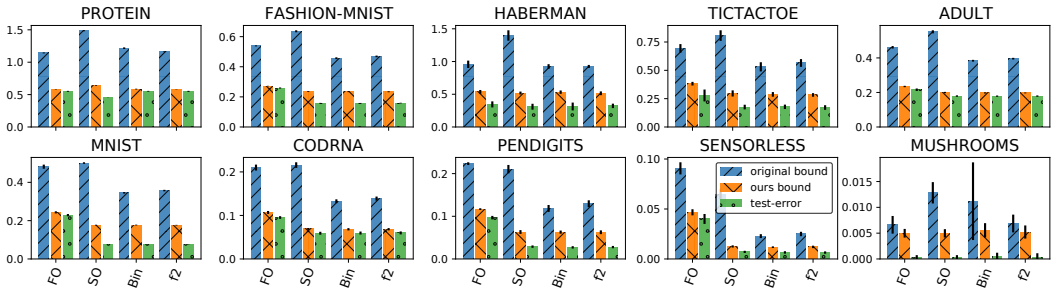

Figure 7.6: Theorem 7.1 (*our bound*) compared with the bounds of *FO*, *SO*, *Bin* or *f2* (*original bound*), and test errors. For each column grouping, $\boldsymbol{\theta}$ is the output from optimising the corresponding PAC-Bayes bound for *rf* on the given dataset.

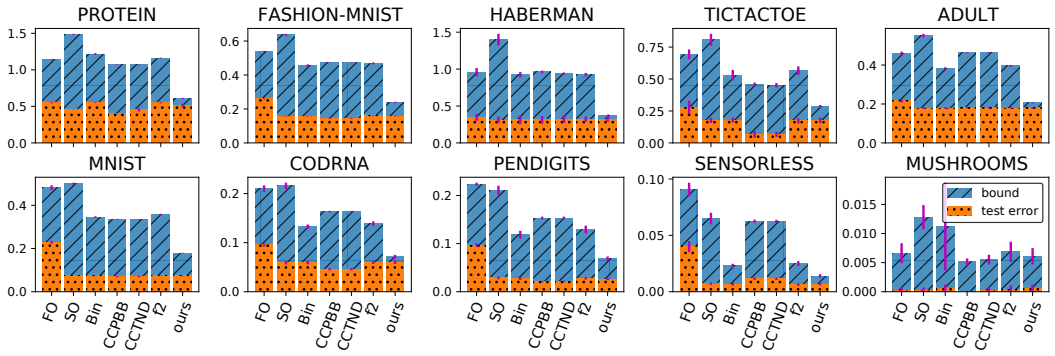

Figure 7.7: Theorem 7.2 (**ours**) as optimisation objective compared to other PAC-Bayes results (*FO*, *SO*, *Bin*, *CCPBB*, *CCTND* and *f2*) as objectives, with a Random Forest as set of voters.

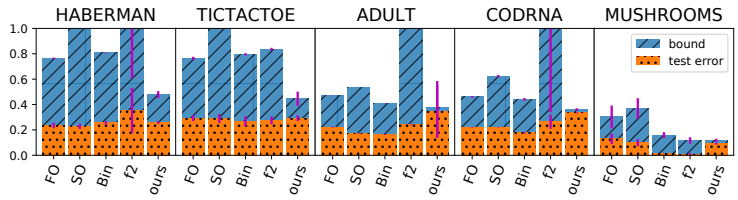

Figure 7.8: Theorem 7.2 (**ours**) as optimisation objective compared to other PAC-Bayes results (*FO*, *SO*, *Bin*, *f2*) as objectives, with decision stumps as voters.

Chapter 8

Tighter PAC-Bayes Generalisation Bounds by Leveraging Example Difficulty

We introduce a modified version of the excess risk, which can be used to obtain empirically tighter, faster-rate PAC-Bayesian generalisation bounds. This modified excess risk leverages information about the relative hardness of data examples to reduce the variance of its empirical counterpart, tightening the bound. We combine this with a new bound for $[-1, 1]$-valued (and potentially non-independent) signed losses, which is more favourable when they empirically have low variance around 0. The primary new technical tool is a novel result for sequences of interdependent random vectors which may be of independent interest. We empirically evaluate these new bounds on a number of real-world datasets.

8.1 Introduction and overview of contributions

Generalisation bounds are of paramount importance in machine learning, both for understanding generalisation, and for obtaining guarantees for predictors. Obtaining the tightest possible bounds shines light on the former and leads to numerically better guarantees for the latter.

Consider a parameterised learning problem where we are interested in training a predictor h_w depending on weights w (*e.g.*, a neural network). In PAC-Bayes, predictions are typically made by drawing randomised weights $W \sim Q$ where Q is a so-called posterior distribution, then predicting $h_W(x)$ for some input x. Thus the learning is moved from the parameter w to a distribution Q over W.

PAC-Bayesian generalisation bounds (Catoni, 2007; McAllester, 1998, 1999; Shawe-Taylor and Williamson, 1997) allow for quantifying the generalisation performance of predictors of

the form h_W with high probability. They can also be used as a stepping stone to proving bounds where w is not random, for example for majority votes (Biggs et al., 2022; Masegosa et al., 2020; Zantedeschi et al., 2021). The recent surge in attention given to the PAC-Bayesian approach partially derives from a number of works establishing numerically non-vacuous bounds for neural networks with randomised (Biggs and Guedj, 2021; Dziugaite and Roy, 2017, 2018; Dziugaite et al., 2021; Letarte et al., 2019; Pérez-Ortiz et al., 2021c; Zhou et al., 2019) or non-randomised (Biggs and Guedj, 2022b) weights on real-world datasets. We refer to Guedj (2019) and Alquier (2021) and the many references therein for a broad introduction to PAC-Bayes.

Two terms commonly appear in PAC-Bayes bounds: $\mathrm{KL} := \mathrm{KL}(Q, P)$, which defines the complexity of Q as a Kullback-Leibler divergence from some sample-independent reference measure (usually referred to as "prior") P; and LG, a term logarithmic in the probability δ. If the number of examples is m, then at worst $\mathrm{LG} \leq \mathcal{O}(\log(m/\delta))$. The simplest such bound for bounded losses (McAllester, 1998) takes the form

$$\text{generalisation gap of } Q \leq \mathcal{O}\left(\sqrt{\frac{\mathrm{KL} + \mathrm{LG}}{m}}\right),$$

holding with probability at least $1-\delta$ over the sample. The above is rarely tight, and was greatly improved by the bound of Maurer (2004), which we discuss further in Section 8.1.3. Maurer's bound has the advantage that it can (when the empirical loss of Q is small) achieve a faster rate of convergence, where the dependence $\mathcal{O}(\sqrt{\mathrm{KL}/m})$ is improved to the "fast-rate" $\mathcal{O}(\mathrm{KL}/m)$. Since commonly $\mathrm{KL} \gg \mathrm{LG}$, this can lead to numerically tighter bounds.

A major question in (PAC-Bayesian) learning theory is *under what conditions such rates can be possible.*

As in VC theory, such fast-rates are possible when the empirical risk of Q is zero, but it is also possible to get close to this fast regime under more general conditions. Getting such faster rates is a primary motivation for "Bernstein" and "Bennett"-type bounds (which leverage low variance to get faster rates) in classical learning theory, as well as for the introduction of the excess loss, which combines nicely with the former.

8.1.1 Notation

In order to further discuss existing approaches, we define our terms more thoroughly. In the following, we examine different PAC-Bayesian generalisation bounds for bounded losses $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$ (where the specific range $[0,1]$ is w.l.o.g. due to the possibility of rescaling). We let $\mathcal{W}$ denote the weight space and $\mathcal{Z}$ is the sample space.

A generalisation bound is an upper bound on the risk $\mathcal{L}(w) := \mathbb{E}_{Z\sim\mathcal{D}}\,\ell(w, Z)$, that holds for some data-dependent hypothesis w. We extend this by abuse of notation in a PAC-Bayesian setting to also write $\mathcal{L}(Q) := \mathbb{E}_{W\sim Q}\,\mathcal{L}(W)$ for PAC-Bayesian posterior distribution

$Q \in \mathscr{P}(\mathcal{W})$.

We also introduce notation for a sequence of examples, $z_{1:i} = (z_1, \ldots, z_i) \in \mathcal{Z}^\star$, where $\mathcal{A}^\star := \emptyset \cup \bigcup_{i=1}^\infty \mathcal{A}^i$ is the set of sequences of elements in set $\mathcal{A}$ and we notate $z_{1:0} = \emptyset$. Learning takes place based on a i.i.d. sample of size m, $S = Z_{1:m} \sim \mathcal{D}^m$, and we define the empirical (in-sample) risk using it as $\widehat{\mathcal{L}}(w) = \mathbb{E}_{Z' \sim \mathrm{Uniform}(S)} \ell(w, Z')$.

8.1.2 Fast rates and excess losses

The simplest PAC-Bayesian bound which can achieve fast rates (and therefore tighter bound values) is the following:

$$\mathcal{L} - \widehat{\mathcal{L}} \leq \sqrt{\frac{\mathrm{KL} + \mathrm{LG}}{m} \cdot 2\widehat{\mathcal{L}}} + 2\frac{\mathrm{KL} + \mathrm{LG}}{m}. \tag{8.1}$$

This bound[1] (which is a relaxation of Maurer's bound, see Section 8.1.3) has a well-studied form common in classical learning theory where the KL term is replaced by a different complexity term. When $\widehat{\mathcal{L}} \to 0$ it achieves the fast rate on $\mathcal{L} - \widehat{\mathcal{L}}$ of $\mathcal{O}(\mathrm{KL}/m)$ and will be numerically tighter, but otherwise (for example, on a difficult dataset where $\widehat{\mathcal{L}}$ is large) the square root term typically dominates.

A common question in learning theory has therefore been on whether empirical risk under the square root can be replaced by something faster-decaying, like a variance (Tolstikhin and Seldin, 2013) or an *excess* risk. The excess risk (which we later generalise from this definition) is introduced by comparing the loss of our hypothesis w to a fixed "good" hypothesis $w^\star$ (we leave aside for now the question of choosing $w^\star$) in a modified loss function, $\tilde{\ell}(w, z) = \ell(w, z) - \ell(w^\star, z)$. This has the population and sample counterparts

$$\mathcal{E}(w) := \mathop{\mathbb{E}}_{Z \sim \mathcal{D}} \tilde{\ell}(w, Z) = \mathcal{L}(w) - \mathcal{L}(w^\star)$$

and

$$\widehat{\mathcal{E}}(w) := \mathop{\mathbb{E}}_{Z \sim \mathrm{Uniform}(S)} \tilde{\ell}(w, Z).$$

For example, Mhammedi et al. (2019) prove the "Unexpected Bernstein" PAC-Bayes bound

$$\mathcal{E} \leq \widehat{\mathcal{E}} + \mathcal{O}\left(\sqrt{\frac{\mathrm{KL} + \mathrm{LG}}{m} \cdot \widehat{V}} + \frac{\mathrm{KL} + \mathrm{LG}}{m}\right),$$

where $\widehat{V}(Q) = \mathbb{E}_{W \sim Q}[\frac{1}{m}\sum_{i=1}^m |\ell(W, Z_i) - \ell(w^\star, Z_i)|^2]$. The idea is that the second loss term in the excess risk "de-biases" and reduces the variance term in $\widehat{V}(Q)$, so that if the predictors err on a similar set of examples, $\widehat{\mathcal{E}}(w)$ will be small, giving a faster rate. Such bounds on $\mathcal{E}$ can be converted back into generalisation bounds, by using that $\mathcal{L}(w^\star) - \widehat{\mathcal{L}}(w^\star) \leq \mathcal{O}(\sqrt{\mathrm{LG}/m})$ (since $w^\star$ is independent of the dataset) to get a bound like

$$\mathcal{L} \leq \widehat{\mathcal{L}} + \mathcal{O}\left(\sqrt{\frac{\mathrm{KL} + \mathrm{LG}}{m} \cdot \widehat{V}} + \frac{\mathrm{KL} + \mathrm{LG}}{m} + \sqrt{\frac{\mathrm{LG}}{m}}\right). \tag{8.2}$$

[1] Note that for the sake of clarity we will make the slight notational abuse of omitting the argument of $\mathcal{L}$, $\widehat{\mathcal{L}}$, $\mathcal{E}$ and $\widehat{\mathcal{E}}$ when the context is clear.

Since in most cases LG $\ll$ KL, the final term is usually an insignificant price compared with the reduction from $\widehat{\mathcal{L}}$ to $\widehat{V}$. The rate of the final term can also be improved even further using assumptions about the noise (as examined at length in Mhammedi et al., 2019), or using dataset evaluations of the loss of $w^\star$.

A problem with this approach is the fact that $w^\star$ must be independent of the data. This means we must split the dataset as with PAC-Bayes data-dependent priors (as in, *e.g.*, Mhammedi et al., 2019; Parrado-Hernández et al., 2012; Pérez-Ortiz et al., 2021a; Rivasplata et al., 2018), into parts used to produce $w^\star$ (and potentially learn a prior), and to actually apply the bound to. This reduces the effective sample size in the bound (*e.g.*, from m to $m/2$ when a 50-50 split is used). This issue can be partially circumvented through the use of forwards-backwards "informed" priors (introduced in Mhammedi et al., 2019 and named such by Wu and Seldin, 2022), but in expectation over different splits of the data this approach is actually weaker than the naive splitting procedure, since it uses Jensen's inequality to combine two bounds.

8.1.3 KL-based bounds

The most well known (and often tightest) PAC-Bayesian bound for bounded losses $\in [0,1]$ is Maurer's bound (Maurer, 2004):

$$\mathrm{kl}\left(\widehat{\mathcal{L}}(Q) \| \mathcal{L}(Q)\right) \leq \frac{\mathrm{KL}\left(Q,P\right) + \log \frac{2\sqrt{m}}{\delta}}{m}$$

where $\mathrm{kl}(q\|p) = q\log\frac{q}{p} + (1-q)\log\frac{1-q}{1-p}$ is the KL divergence between Bernoulli distributions of biases q, p. This bound can be inverted to obtain an upper bound directly on $\mathcal{L}$ by defining the inverse

$$\mathrm{kl}^{-1}(u\|b) := \sup\{r : \mathrm{kl}(u\|r) \leq b\}.$$

The bound in Eq. (8.1) is obtained through the relaxation $\mathrm{kl}^{-1}(u\|b) \leq u + \sqrt{2bu} + 2b$ (McAllester, 2003). However, note that this relaxed bound can be considerably weaker, as it does not leverage the combinatorial power of the small-kl. It has been shown (Foong et al., 2021) that no bound on the naive loss (but not necessarily when we leverage the excess loss) can improve Maurer's bound (aside from the open question of whether it is possible to remove the logarithmic factor).

We note also that although the small-kl bound can be re-scaled to use the excess loss, this leads to a bound like Eq. (8.2) with $\widehat{V} = \frac{\widehat{\mathcal{E}}+1}{2} \geq \frac{1}{2}\widehat{\mathcal{L}}$ (by applying the bound to $\frac{\mathcal{E}+1}{2}$ and relaxing as above). This does not lead to fast rates under different conditions to usual, since it is still necessary that $\widehat{\mathcal{L}} \to 0$.

Recently, Adams et al. (2022) proved a generalisation of this bound which holds for *vector*-valued losses, $\boldsymbol{\ell} : \mathcal{W} \times \mathcal{Z} \to \Delta_{M-1}$ (with

$simplex M$ the M-dimensional simplex),

$$\mathrm{kl}\left(\mathop{\mathbb{E}}_{W\sim Q}\widehat{\boldsymbol{U}}(W)\middle\|\mathop{\mathbb{E}}_{W\sim Q}\boldsymbol{\mu}(W)\right)\leq\frac{\mathrm{KL}\left(Q,P\right)+\log\frac{\xi(M,m)}{\delta}}{m}.$$

where $\widehat{U}_i(w)=\frac{1}{m}\sum_{i=1}^m\ell_i(w,z)$ and $\boldsymbol{\mu}=\mathbb{E}_S\widehat{\boldsymbol{U}}$, and $\xi(M,m)$ is a polynomial function of m. We also note a closely related earlier bound in Seldin and Tishby (2010, Theorem 2). Inverting such a bound is somewhat more complicated, but we can use it to obtain an upper bound on $\sum_i\alpha_iU_i$ for some set of coefficients α_i. This is the tool we will use to obtain our bounds.

8.1.4 Our contributions

From the above starting points, we pursue two parallel and complimentary directions of improvement. First we provide a generalisation of the excess risk which allows $w^\star$ to be learned from the stream of data as we receive it. In this way, we are able to more effectively "de-bias" our bounds, reducing the effective variance term. This is intended to demonstrate how PAC-Bayes bounds can be made tighter by using information from the training set (in a sense, the relative difficulty of different examples) more efficiently.

Secondly, we observe that Eq. (8.1) is a *relaxation* of Maurer's bound, and this weakening leads to a loss of some of the tightness of the original "kl" formulation. Thus we give a new bound which leverages the tightness of kl-based bounds like Maurer's, but also relaxes to a form like that in Eq. (8.2). Specifically, it reduces to the form given in Eq. (8.2) with $\widehat{V}(Q)=\mathbb{E}_{W\sim Q}\frac{1}{m}|\ell(W,Z_i)-\ell(w^\star,Z_i)|$. This is very similar (if slightly larger) to the term given in Mhammedi et al. (2019), although both are equivalent when $\ell\in\{0,1\}$, as for example with the misclassification loss, which ensures faster rates under similar conditions. However this form of our bound is only a relaxation, and the kl-type formulation that we give for it is considerably tighter empirically, and we show it is strictly tighter under a $\{0,1\}$-valued loss. In combination, these lead to a new bound on the out-of-sample risk that is empirically tighter than Maurer's bound in some cases, a feat very rare in the literature.

This new bound is related to the "split-kl" bound of Wu and Seldin (2022), which decomposes the excess loss into positive and negative parts, and applies Maurer's bound to each. Our result instead derives from a starting point of vector-based and ternary $\in\{-1,0,1\}$ losses, which can potentially lead to tighter bounds[2], but combines with their decomposition when extending to non-ternary bounded losses.

In order to prove these results, we make several technical contributions, including new results for simplex-valued Martingales within PAC-Bayes and a new method for relaxing our kl-type bound.

[2]In the standard regime of $\mathrm{KL}\gg\log m$, basic empirical comparisons show our bound to be $5-10\%$ tighter, but when $\log m\gg\mathrm{KL}$ our bound can be weaker.

8.2 Warm up

In this section we give a simplified version of our main results, discussing only classification. In this setting, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ for $\mathcal{Y} = \{1, \ldots, d_{\text{out}}\}$, and $S = \{(X_i, Y_i)\}_{i=1}^m$. Our predictions are given by $h_w(x)$ for $w \in \mathcal{W}$ and we consider the misclassification loss, $\ell(w, (x, y)) = \mathbf{1}_{h_w(x) \neq y}$.

We consider positive and negative deviations of the excess loss, $\widehat{\mathcal{E}}_+$ and $\widehat{\mathcal{E}}_-$, which we will define for general losses in the next section. For intuition, in the *specific case* of misclassification loss they are equal to the following

$$\widehat{\mathcal{E}}_+^{\text{mc}} = \mathop{\mathbb{E}}_{W \sim Q}\left[\frac{|\{(x,y) \in S : h_W(x) \neq y, h_{w^\star}(x) = y\}|}{m}\right],$$
$$\widehat{\mathcal{E}}_-^{\text{mc}} = \mathop{\mathbb{E}}_{W \sim Q}\left[\frac{|\{(x,y) \in S : h_W(x) = y, h_{w^\star}(x) \neq y\}|}{m}\right].$$

Thus in this case we are merely counting the numbers of two different types of loss: an error using w but not using $w^\star$, and the converse. If neither predictor or both predictors err, this incurs no loss. These two error types have simple interpretations as counts, which is similar to the work of Adams et al. (2022), and indeed the preliminary results we show here could follow fairly straightforwardly from theirs.

We note that $\mathcal{E} = \mathcal{E}_+ - \mathcal{E}_-$, and

$$\widehat{\mathcal{E}}_+^{\text{mc}} + \widehat{\mathcal{E}}_-^{\text{mc}} = \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{P}}_{W \sim Q}(h_W(X_i) \neq h_{w^\star}(X_i)),$$

a form which commonly appears in learning theory, and can also be controlled by Mammen-Tsybakov or Massart noise conditions.

Our main result implies the following (weakened) relaxed bound:

$$\mathcal{E} \leq \widehat{\mathcal{E}} + 2\sqrt{\frac{\text{KL} + \text{LG}}{m} \cdot (\widehat{\mathcal{E}}_+ + \widehat{\mathcal{E}}_-)} + 2\frac{\text{KL} + \text{LG}}{m} \tag{8.3}$$

with $\text{LG} = \log(2m/\delta)$.

We then can easily go from Eq. (8.3) to a bound on the population risk (like Eq. (8.2)), since $w^\star$ is independent of the data, and can be estimated empirically with no complexity penalty. The (simplest) Hoeffding bound gives $\mathcal{L}(w^\star) - \widehat{\mathcal{L}}(w^\star) \leq \sqrt{\log(1/\delta)/2m}$. Overall, with cancellations (and a union bound, so that the following holds with probability $1 - 2\delta$) this gives a bound of

$$\mathcal{L} \leq \widehat{\mathcal{L}} + 2\sqrt{\frac{\text{KL} + \text{LG}}{m} \cdot (\widehat{\mathcal{E}}_+ + \widehat{\mathcal{E}}_-)} + 2\frac{\text{KL} + \text{LG}}{m} + \sqrt{\frac{\log \delta^{-1}}{2m}}. \tag{8.4}$$

The last term is complexity-free and hence will generally be dominated by the other terms.

8.2.1 KL-type formulation

Here we show that the above can be considerably tightened into a "kl"-type formulation. This can be relaxed back into the previous form, similarly to the way in which Maurer's

bound implies the much weaker result in Eq. (8.1). Collecting the error type counts into the vector

$$\widehat{\mathcal{E}}(Q) = [\widehat{\mathcal{E}}_+(Q), \widehat{\mathcal{E}}_-(Q), 1 - \widehat{\mathcal{E}}_+(Q) - \widehat{\mathcal{E}}_-(Q)]^T,$$

we will prove the bound

$$\mathcal{E}(Q) \leq \phi\left(\widehat{\mathcal{E}}(Q), \frac{\mathrm{KL}\,(Q,P) + \log\frac{2m}{\delta}}{m}\right)$$

where $\phi(\boldsymbol{u}, b) := \sup\{r_1 - r_2 : \boldsymbol{r} \in \Delta_{3-1}, \mathrm{kl}(\boldsymbol{u}\|\boldsymbol{r}) \leq b\}$.

This function (and its gradients) can be calculated by a simple procedure outlined in Section 8.B.2 This form leverages the tightness of the kl bound and is empirically much tighter than Eq. (8.3). In the next section we give a more detailed analysis of this function and show that it implies several relaxations, including Eq. (8.3), the unexpected Bernstein bound (for certain losses), and a split-kl type inequality similar to that from Wu and Seldin (2022).

For intuition, we point out that this basic result (though not the relaxations which require further proofs, or the general bounded loss forms, since their result applies only for count-based losses) can be proved by application of results from Adams et al. (2022) to a vector loss of counts, $\boldsymbol{\ell} : \mathcal{W} \times \mathcal{Z} \to \Delta_{3-1}$, with

$$\begin{aligned}
\ell_1 &= \mathbf{1}_{h_w \neq y, h_{w^\star} = y}, \\
\ell_2 &= \mathbf{1}_{h_w = y, h_{w^\star} \neq y}, \\
\ell_3 &= 1 - \ell_1 - \ell_2.
\end{aligned}$$

Just as above, this result can be combined with a (test set) bound on $\mathcal{L}(w^\star)$ using $\widehat{\mathcal{L}}(w^\star)$ to provide a generalisation bound for Q, as in Eqs. (8.2) and (8.4). Again, to leverage the tightness of the kl formulation, we could replace the weaker Hoeffding bound used before with a tighter (kl-based, inverted) Chernoff bound (as given in *e.g.* Foong et al., 2022):

$$\mathcal{L}(w^\star) \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}(w^\star) \middle\| \frac{\log\frac{1}{\delta}}{m}\right).$$

Thus we can get an overall result with two inverse kl-type terms, which is much tighter in practice than the formulations given in the previous section. In Section 8.3 we give formal statements of the above and adapt the bound to work for any bounded loss function.

8.2.2 Generalising the excess loss

A problem with formulations using the excess risk is that the optimal choice of $w^\star$ is essentially impossible to know in advance. In order to optimally de-bias bounds, $w^\star$ ought to perform similarly to our overall posterior, essentially identifying difficult examples, but this is very difficult when it must be chosen in a data-free way. One solution to this problem

is data-splitting, but this is sub-optimal because it reduces the amount of data available to the bound.

Here instead, drawing on PAC-Bayesian tools for non-independent, martingale based losses, we propose an alternative solution through a generalisation of the excess risk. We first recall the notation $z_{1:i}$ for a sequence of examples $z_1, \ldots, z_i$. Given a sequence of i.i.d. variables Z_i, it turns out we can derive empirically-calculable bounds by considering a sequence of losses like

$$\ell(w, Z_i) - \ell(w_i^\star(Z_{1:i-1}), Z_i). \tag{8.5}$$

Here, instead of being fixed in a data independent way, $w_i^\star(Z_{1:i-1})$ is an online sequence of weights learned using the first $i-1$ examples. If the procedure used to learn the $w_i^\star$ is similar to that used to learn $W \sim Q$, and is relatively stable to changes in dataset size, the errors of Q and the $w_i^\star$ will be highly correlated. The terms in Eq. (8.5) will mostly cancel, reducing the excess risk and tightening the overall bound. This approach can be easily generalised to stochastic online algorithms, a procedure mentioned by Mhammedi et al. (2019) as "online estimators", but not used empirically. We note that, unlike with data-dependent priors in PAC-Bayes bounds, no data-splitting is necessary for this procedure, the bound still uses all of the training data with $m = |S|$.

To clarify further, this technique can tighten any bound that leverages excess losses to achieve faster rates, such as ours or the unexpected Bernstein, but *not* Maurer's bound. Ordering examples by difficulty is not necessary; what is useful is that a learner on the first i points makes similar errors to our (whole-dataset) posterior, leveraging the notion that not all examples have the same difficulty. This is what we mean by "leveraging example difficulty".

8.3 Main results

In the following, $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$ is a bounded loss and $S = Z_{1:m}$ is an i.i.d. sample from unknown distribution $\mathcal{D}$. We choose a sequence of online estimators $Q_i^\star \in \mathscr{P}(\mathcal{W})$ for $i \in \{1, \ldots, m\}$, each depending on only the previous $i-1$ examples $Z_{1:i-1}$. This might be done for example through choosing $Q_i^\star = \mathbb{A}(Z_{1:i-1})$, where $\mathbb{A} : \mathcal{Z}^\star \to \mathscr{P}(\mathcal{W})$ is a predetermined algorithm.

Our de-biased (generalised, in that it depends on i examples) loss for the ith example takes the form

$$\tilde{\ell}(w, z_{1:i}) := \ell(w, z_i) - \mathop{\mathbb{E}}_{W' \sim Q_i^\star}[\ell(W', z_i)].$$

As in Wu and Seldin (2022), we distinguish two different types of sample errors (corresponding to positive and negative parts $\tilde{\ell} > 0$ and $\tilde{\ell} < 0$) with corresponding generalised empirical risk

values:

$$\widehat{\mathcal{E}}_+(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m}\max(\tilde{\ell}(W,Z_{1:i}),0)\right],$$
$$\widehat{\mathcal{E}}_-(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m}\max(-\tilde{\ell}(W,Z_{1:i}),0)\right].$$

For notational convenience we collect these in the vector

$$\widehat{\mathcal{E}}(Q) = [\widehat{\mathcal{E}}_+(Q), \widehat{\mathcal{E}}_-(Q), 1-\widehat{\mathcal{E}}_+(Q)-\widehat{\mathcal{E}}_-(Q)]^T.$$

Given a PAC-Bayesian posterior $Q \in \mathscr{P}(\mathcal{W})$, we define the generalised excess risk as

$$\mathcal{E}(Q) := \mathcal{L}(Q) - \mathcal{L}^\star$$

with

$$\mathcal{L}^\star := \frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{Z}\mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z)].$$

We next present a result in two parts which can be used to control $\mathcal{E}$ and $\mathcal{L}^\star$, and hence obtain a bound on $\mathcal{L}(Q)$ directly.

Theorem 8.1 (Generalisation Loss Bound)**.** *Fix data distribution $\mathcal{D} \in \mathscr{P}(\mathcal{Z})$, prior $P \in \mathscr{P}(\mathcal{W})$, $m \geq 3, \delta \in (0,1)$, and bounded loss $\ell : \mathcal{Z}\times\mathcal{W} \to [0,1]$. Let $S = Z_{1:m} \sim \mathcal{D}^m$ be a sample and $Q_i^\star \in \mathscr{P}(\mathcal{W})$ be any sequence of online estimators with $i = 1,\dots,m$, each depending only on the $i-1$ examples $Z_{1:i-1}$, and let $\mathcal{E}$ and $\widehat{\mathcal{E}}$ be as defined in this section.*

Then, with probability at least $1-\delta$ over S for arbitrary posterior $Q \in \mathscr{P}(\mathcal{W})$,

$$\mathcal{L}(Q) \leq \phi\left(\widehat{\mathcal{E}}(Q), \frac{\mathrm{KL}\,(Q,P) + \log\frac{2m}{\delta}}{m}\right) + \mathcal{L}^\star,$$

where $\phi(\boldsymbol{u}, b) := \sup\{r_1 - r_2 : \boldsymbol{r} \in \Delta_{3-1}, \mathrm{kl}(\boldsymbol{u}\|\boldsymbol{r}) \leq b\}$.

Further, with probability at least $1-\delta$ over S,

$$\mathcal{L}^\star \leq \mathrm{kl}^{-1}\left(\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z_i)]\middle\|\frac{\log\frac{1}{\delta}}{m}\right).$$

With probability at least $1-2\delta$, the above hold simultaneously, and $\mathcal{L}(Q) = \mathcal{E}(Q) + \mathcal{L}^\star$ is bounded by the sum of the right hand sides.

This is a complex result, so to begin we note that a deterministic and data-free choice of $Q_i^\star$ as a distribution fixed to $w^\star$ recovers the results from Section 8.2.1. The relaxed form given by Eq. (8.3) is still also valid with the modified forms of the excess risk, and gives intuition about the bound; in this more general (bounded, not misclassification) case,

$$\widehat{\mathcal{E}}_+ + \widehat{\mathcal{E}}_- = \frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{W\sim Q}|\ell(W,Z_i) - \mathop{\mathbb{E}}_{W'\sim Q_i^\star}\ell(W',Z_i)|.$$

In the next sub-sections, we examine some corollaries and relaxations of this bound. We also note here that it is possible to directly calculate the value and gradients of ϕ with respect to both of its arguments using a procedure outlined by Adams et al. (2022), which could be very useful in optimising the bound directly as an objective.

8.3.1 Relaxation to Maurer

As a basic sanity check to show that the new bound leverages the tightness of a small-kl-based bound, we show that it can be used to recover Maurer's bound (up to a factor 2 in front of the logarithmic term). Assume the realisable case[3], where there exists $w^\star$ such that $\ell(w^\star, z) = 0$ for all z in the support of $\mathcal{D}$. This ensures that $\widehat{\mathcal{E}}_- = 0$, $\mathcal{L}(Q) = \mathcal{E}(Q)$, and

$$\widehat{\mathcal{E}}_+(Q) = \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m}\ell(W, Z_i)\right] = \widehat{\mathcal{L}}(Q).$$

We also have the following.

Proposition 8.1. *For any $u \in [0,1]$ and $b > 0$*

$$\phi([u, 0, 1-u], b) = \mathrm{kl}^{-1}(u\|b).$$

Combining Proposition 8.1 with our Theorem 8.1 in this setting implies that

$$\mathcal{L}(Q) \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{L}}(Q)\middle\|\frac{\mathrm{KL}\left(Q,P\right) + \log\frac{2m}{\delta}}{m}\right),$$

which is Maurer's bound (up to a constant factor for the log term). In the realisable case, we would expect $\widehat{\mathcal{L}}(Q) \to 0$ and de-biasing is not necessary to achieve faster rates, so this shows that we can recover the tightest known bound for that scenario.

8.3.2 Relaxation to unexpected Bernstein and noise conditions

The following result can be used to obtain a number of relaxations of our result that enable more direct comparison to Mhammedi et al. (2019).

Proposition 8.2. *For any $\boldsymbol{u} \in \Delta_{3-1}, b > 0$,*

$$\begin{aligned}\phi(\boldsymbol{u}, b) &\leq \inf_{\eta\in(0,1)} \frac{1}{\eta}\left(1 - e^{u_1\log(1-\eta) + u_2\log(1+\eta) - b}\right)\\ &\leq \inf_{\eta\in(0,1)} f_\eta(u_1 - u_2 + c_\eta(u_1+u_2) + b/\eta)\end{aligned}$$

where $f_\eta(t) = \eta^{-1}(1-e^{-\eta t}) \leq t$, and $c_\eta = -1 - \log(1-\eta)/\eta$ is a term also appearing in the unexpected Bernstein bound.

[3]Surprisingly, we can assume this w.l.o.g. by a mathematical trick, extending $\mathcal{W}$ to contain a special, non-learnable point $w^\dagger$ such that $\ell(w^\dagger, z) = 0$ for all z. However, this deprives us of the use of excess losses, which can give tighter bounds when we do not learn a low-risk solution as in the "true" realisable case.

This enables us to relax our main result to find that simultaneously for any $\eta \in (0,1)$

$$\mathcal{E} \leq f_\eta \left(\widehat{\mathcal{E}}_+ - \widehat{\mathcal{E}}_- + c_\eta(\widehat{\mathcal{E}}_+ + \widehat{\mathcal{E}}_-) + \frac{\mathrm{KL} + \mathrm{LG}}{m\eta} \right). \tag{8.6}$$

Firstly we note that setting $\eta = 1 - e^{-C}$ for $C > 0$ gives a bound similar to Catoni's Catoni (2007) with a free choice of C and an extra $\log(2m)$ term, and recovers it exactly when $\widehat{\mathcal{E}}_- = 0$.

We also compare this relaxation with the strongest form of the unexpected Bernstein (given by Mhammedi et al. (2019) and Theorem 8.6 our appendix), which takes the form

$$\mathcal{E} \leq \widehat{\mathcal{E}}_+ - \widehat{\mathcal{E}}_- + c_\eta \hat{V} + \frac{\mathrm{KL} + \mathrm{LG}}{m\eta}$$

for a relatively free (it must be chosen from within some covering grid) choice of η. In the specific case of the misclassification loss (and non-randomised $Q_i^\star$ each supported on a single point), the term $\widehat{\mathcal{E}}_+ + \widehat{\mathcal{E}}_- = \hat{V}$, is equal to that from the unexpected Bernstein. We therefore recover a bound (Eq. (8.6)) taking the same form but with an extra $f_\eta(t) \leq t$ around it, so in the case of misclassification loss our bound is always at least as strong as the results of Mhammedi et al. (2019). For more general losses, since the squaring of the loss difference terms in $\hat{V}$ makes them smaller, the unexpected Bernstein might be more able to leverage small but non-zero per-example excess losses.

A corollary of this misclassification equivalence is that our bound can achieve faster rates under a (c,β)-Mammen-Tsybakov noise condition. These noise conditions for the misclassification loss imply a β-Bernstein condition. Under the analysis of Mhammedi et al. (2019, Section 5), with the same learning algorithm choices, our bound therefore achieves a rate of at least $\widetilde{\mathcal{O}}(m^{-1/(2-\beta)})$.

8.3.3 Relaxation to split-kl

A relaxation of our bound that is very similar in form to the recent PAC-Bayes split-kl inequality from Wu and Seldin (2022) is obtained through the following proposition.

Proposition 8.3. *For any $\boldsymbol{u} \in \Delta_{3-1}$, $b > 0$,*

$$\begin{aligned} \phi(\boldsymbol{u}, b) &\leq \mathrm{kl}^{-1}(u_1 \| b) - \mathrm{kl}_{\mathrm{LB}}^{-1}(u_2 \| b) \\ &\leq u_1 - u_2 + 2\sqrt{b \cdot (u_1 + u_2)} + 2b, \end{aligned}$$

where $\mathrm{kl}_{\mathrm{LB}}^{-1}(u\|b) := \inf\{r \in [0,1] : \mathrm{kl}(u\|r) \leq b\}$ is the lower *tail small-kl inversion.*

This gives the bound

$$\mathcal{E} \leq \mathrm{kl}^{-1}\left(\widehat{\mathcal{E}}_+ \middle\| \frac{\mathrm{KL} + \mathrm{LG}}{m} \right) - \mathrm{kl}_{\mathrm{LB}}^{-1}\left(\widehat{\mathcal{E}}_- \middle\| \frac{\mathrm{KL} + \mathrm{LG}}{m} \right). \tag{8.7}$$

This is essentially the same as the split-kl inequality (Wu and Seldin, 2022), except that we have $\mathrm{LG} = \log \frac{2m}{\delta}$ while in their bound $\mathrm{LG} = \log \frac{4\sqrt{m}}{\delta}$, so the constants in ours are slightly

worse, as in Section 8.3.1. Their main bound is not limited to excess losses, but is primarily aimed at losses where there are three different special values to be focused on (in the simple case, $\{-1,0,1\}$, as here). We note that the techniques they use to do this (which involve re-scaling and translating loss functions) could also be combined with our (non-split) kl formulation.

8.3.4 Aside: slight generalisations

We here give a slight generalisation of our main result. This suggests two different possible directions for our main technical results.

Theorem 8.2. *For any measurable $\tilde{\ell} : \mathcal{W} \times \mathcal{Z}^\star \to [-1,1]$ as defined above and any $\mathcal{D} \in \mathscr{P}(\mathcal{Z}), m \geq 3$, with probability at least $1-\delta$ over $S \sim \mathcal{D}^m$ simultaneously for all $Q \in \mathscr{P}(\mathcal{W})$,*

$$\mathfrak{L}(Q) \leq \phi\left(\widehat{\mathfrak{L}}(Q), \frac{\mathrm{KL}\,(Q,P) + \log\frac{2m}{\delta}}{m}\right),$$

where we have defined

$$\begin{aligned}
\widehat{\mathfrak{L}}_+(Q) &:= \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^m \max(\tilde{\ell}(W,Z_{1:i}),0)\right],\\
\widehat{\mathfrak{L}}_-(Q) &:= \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^m \max(-\tilde{\ell}(W,Z_{1:i}),0)\right],\\
\widehat{\mathfrak{L}}(Q) &:= [\widehat{\mathfrak{L}}_+(Q), \widehat{\mathfrak{L}}_-(Q), 1-\widehat{\mathfrak{L}}_+(Q)-\widehat{\mathfrak{L}}_-(Q)]^T,\\
\mathfrak{L}(Q) &:= \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{Z_i}[\tilde{\ell}(Z_{1:i})|Z_{1:i-1}],
\end{aligned}$$

and $\phi(\boldsymbol{u},b) := \sup\{r_1 - r_2 : \boldsymbol{r} \in \Delta_{3-1}, \mathrm{kl}(\boldsymbol{u}\|\boldsymbol{r}) \leq b\}$.

Firstly, this shows that we can straightforwardly consider general "signed" loss in $[-1,1]$, rather than an excess loss. This takes our work closer to that of Wu and Seldin (2022).

We can also consider a more general de-biasing term, setting $\tilde{\ell} = \ell(w,z_i) - \ell^\star(z_{1:i})$, where $\ell^\star : \mathcal{Z}^\star \to [0,1]$. However, to obtain bounds on $\mathcal{L}(Q)$ with for arbitrary choices of this function, we must also bound

$$\mathcal{L}(Q) - \mathfrak{L}(Q) = \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{Z_i}[\ell^\star(Z_{1:i})|Z_{1:i-1}].$$

This is difficult in general, but in some cases the following result is useful: One way to do this is suggested by the following:

Theorem 8.3 (Martingale Chernoff-Hoeffding Inversion)**.** *Let $U_i \in [0,1], i = 1,\ldots,m$ have conditional expectations $\mathbb{E}[U_i|U_{1:i-1}] = \mu_i$, and averages $\bar{U} := \frac{1}{m}\sum_{i=1}^m U_i$, $\bar{\mu} = \frac{1}{m}\sum_{i=1}^m \mu_i$. Then with probability at least $1-\delta$,*

$$\bar{\mu} \leq \mathrm{kl}^{-1}\left(\bar{U}\middle\|\frac{\log\frac{1}{\delta}}{m}\right).$$

This result shows that any fixed method of choosing the de-biasing term can work. For example, we could train some model on the first $i-1$ points to predict the loss (which is in $[0,1]$) of the next data point. As long as this model is not trained on the i-th point, its evaluation on this point is akin to a real test-set evaluation, and so we can get a relatively sharp bound on the term $\mathcal{L}(Q) - \mathfrak{L}(Q)$.

8.4 Proofs and corollaries

Firstly, we prove two theorems which generalise theorems of Adams et al. (2022) to random variables with a dependence structure, using ideas from Seldin et al. (2012). These results may be of interest in their own right.

Theorem 8.4 (Generalisation of Lemma 5 in Adams et al., 2022 and Lemma 1 in Seldin et al., 2012)**.** *Let $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_m$ be a sequence of random vectors, each in Δ_{M-1}, such that*

$$\mathbb{E}[\boldsymbol{U}_i | \boldsymbol{U}_1, \ldots, \boldsymbol{U}_{i-1}] = \boldsymbol{\mu}_i$$

for $i = 1, \ldots, m$. Let $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_m$ be independent Multinomial$(1, M, \boldsymbol{\mu}_i)$[4]*. random vectors such that $\mathbb{E}\boldsymbol{V}_i = \boldsymbol{\mu}_i$. Then for any convex function $f : \Delta_{M-1}^m \to \mathbb{R}$:*

$$\mathbb{E}[f(\boldsymbol{U}_1, \ldots, \boldsymbol{U}_m)] \leq \mathbb{E}[f(\boldsymbol{V}_1, \ldots, \boldsymbol{V}_m)].$$

Proof. Let E_M denote the set of canonical (axis-aligned) M-dimensional basis vectors, for example $E_3 = \{[1,0,0], [0,1,0], [0,0,1]\}$. We will denote typical members of this set by $\boldsymbol{\eta}_i$, and tuples $\boldsymbol{\eta}_{1:m} = (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_m) \in \Delta_{M-1}^m$. Firstly we show that the definitions in the theorem lead to a Martingale-type result:

$$\begin{aligned}
&\mathbb{E}\left[\prod_{i=1}^{m} \boldsymbol{U}_i \cdot \boldsymbol{\eta}_i\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{U}_{1:m-1}}\left[\left(\prod_{i=1}^{m-1} \boldsymbol{U}_i \cdot \boldsymbol{\eta}_i\right) \mathop{\mathbb{E}}_{\boldsymbol{U}_m}[\boldsymbol{U}_m | \boldsymbol{U}_{1:m-1}] \cdot \boldsymbol{\eta}_m\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{U}_{1:m-1}}\left[\left(\prod_{i=1}^{m-1} \boldsymbol{U}_i \cdot \boldsymbol{\eta}_i\right) \boldsymbol{\mu}_m \cdot \boldsymbol{\eta}_m\right] \\
&= \prod_{i=1}^{m} \boldsymbol{\mu}_i \cdot \boldsymbol{\eta}_i.
\end{aligned}$$

In Adams et al. (2022, proof of Lemma 5), it is shown that for any convex function $f : \Delta_{M-1}^m \to \mathbb{R}$ and $\boldsymbol{u}_{1:m} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m) \in \Delta_{M-1}^m$,

$$f(\boldsymbol{u}_{1:m}) \leq \sum_{\boldsymbol{\eta}_{1:m} \in E_M^m} \left(\prod_{i=1}^{m} \boldsymbol{u}_i \cdot \boldsymbol{\eta}_i\right) f(\boldsymbol{\eta}_{1:m}).$$

[4] A multinomial distribution Multinomial$(m, M, \boldsymbol{p})$ is supported on non-negative integers $x_1 \ldots x_M$ summing to m, with probability mass function $\frac{m!}{x_1! \ldots x_M!} p_1^{x_1} \ldots p_M^{x_M}$.

Applying this result to the random variables $\boldsymbol{U}_{1:m}$ and combining with the Martingale-type result leads to the following:

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{U}_{1:m})] \\
&\leq \mathbb{E}\left[\sum_{\boldsymbol{\eta}_{1:m}\in E_M^m}\left(\prod_{i=1}^m \boldsymbol{U}_i\cdot\boldsymbol{\eta}_i\right) f(\boldsymbol{\eta}_{1:m})\right] \\
&= \sum_{\boldsymbol{\eta}_{1:m}\in E_M^m}\mathbb{E}\left[\prod_{i=1}^m \boldsymbol{U}_i\cdot\boldsymbol{\eta}_i\right] f(\boldsymbol{\eta}_{1:m}) \\
&= \sum_{\boldsymbol{\eta}_{1:m}\in E_M^m}\left(\prod_{i=1}^m \boldsymbol{\mu}_i\cdot\boldsymbol{\eta}_i\right) f(\boldsymbol{\eta}_{1:m}) \\
&= \sum_{\boldsymbol{\eta}_{1:m}\in E_M^m}\left(\prod_{i=1}^m \mathbb{P}(\boldsymbol{V}_i=\boldsymbol{\eta}_i)\right) f(\boldsymbol{\eta}_{1:m}) \\
&= \mathbb{E}\, f(\boldsymbol{V}_{1:m}).
\end{aligned}$$

The final step in the proof followed via the definition of expectation w.r.t. the $\boldsymbol{V}_i$. □

Theorem 8.5 (Martingale PAC-Bayes for Vector KL.)**.** *Let $\boldsymbol{U}_1(w),\ldots,\boldsymbol{U}_m(w)$ be a sequence of random vector valued functions, each in Δ_{M-1}, such that*

$$\mathbb{E}[\boldsymbol{U}_i(w)|\boldsymbol{U}_1(w),\ldots,\boldsymbol{U}_{i-1}(w)] = \boldsymbol{\mu}_i(w)$$

for $i=1,\ldots,m$ and all $w\in\mathcal{W}$. Define

$$\widehat{\boldsymbol{U}}(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^m \boldsymbol{U}_i(W)\right]$$

and

$$\bar{\boldsymbol{\mu}}(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^m \boldsymbol{\mu}_i(W)\right].$$

Then for fixed $P\in\mathscr{P}(\mathcal{W}), \delta\in(0,1)$, with probability at least $1-\delta$ (over $\{\boldsymbol{U}_i(w): i\in 1,\ldots,m : w\in\mathcal{W}\}$), simultaneously for all $Q\in\mathscr{P}(\mathcal{W})$,

$$\mathrm{kl}\left(\widehat{\boldsymbol{U}}(Q)\middle\|\bar{\boldsymbol{\mu}}(Q)\right) \leq \frac{\mathrm{KL}(Q,P)+\log\frac{\xi(M,m)}{\delta}}{m}$$

where $\xi(M,m)$ is defined for $m\geq M$ by

$$\sqrt{\pi}e^{1/12m}\left(\frac{m}{2}\right)^{\frac{M-1}{2}}\sum_{k=0}^{M-1}\binom{M}{k}\frac{1}{(m\pi)^{k/2}\Gamma\left(\frac{M-k}{2}\right)}.$$

Proof. The proof begins by a common pattern in PAC-Bayesian proofs (as first pointed out by Germain et al., 2009). By Jensen's inequality, the Donsker-Varadhan change-of-measure theorem, Markov's inequality and the independence of P from $\{\boldsymbol{U}_i,\boldsymbol{\mu}_i\}$, the following holds with at least $1-\delta$ for any Q:

$$m\,\mathrm{kl}\left(\widehat{\boldsymbol{U}}(Q)\middle\|\bar{\boldsymbol{\mu}}(Q)\right) - \mathrm{KL}(Q,P)$$

$$
\begin{aligned}
&\leq m \mathop{\mathbb{E}}_{W\sim Q} \mathrm{kl}\left(\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{U}_i(w) \middle\| \bar{\boldsymbol{\mu}}(w)\right) - \mathrm{KL}\left(Q, P\right) \\
&\leq \log \mathop{\mathbb{E}}_{W\sim P}\left[e^{m\,\mathrm{kl}\left(\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{U}_i(w) \middle\| \bar{\boldsymbol{\mu}}(w)\right)}\right] \\
&\leq \log \frac{1}{\delta} \mathop{\mathbb{E}}_{\boldsymbol{U}_i} \mathop{\mathbb{E}}_{W\sim P}\left[e^{m\,\mathrm{kl}\left(\frac{1}{m}\sum_{i} \boldsymbol{U}_i(w) \middle\| \bar{\boldsymbol{\mu}}(w)\right)}\right] \\
&\leq \log \frac{1}{\delta} \mathop{\mathbb{E}}_{W\sim P} \mathop{\mathbb{E}}_{\boldsymbol{U}_i}\left[e^{m\,\mathrm{kl}\left(\frac{1}{m}\sum_{i} \boldsymbol{U}_i(w) \middle\| \bar{\boldsymbol{\mu}}(w)\right)}\right].
\end{aligned}
$$

By applying Theorem 8.4 to the inner term we find that

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{U}_i}\left[e^{m\,\mathrm{kl}\left(\frac{1}{m}\sum_{i} \boldsymbol{U}_i(w) \middle\| \bar{\boldsymbol{\mu}}(w)\right)}\right] \\
&\quad\leq \mathop{\mathbb{E}}_{\boldsymbol{V}_i}\left[e^{m\,\mathrm{kl}\left(\frac{1}{m}\sum_{i} \boldsymbol{V}_i(w) \middle\| \bar{\boldsymbol{\mu}}(w)\right)}\right] \\
&\quad\leq \mathop{\mathbb{E}}_{\bar{\boldsymbol{V}}}\left[e^{m\,\mathrm{kl}\left(\bar{\boldsymbol{V}} \middle\| \bar{\boldsymbol{\mu}}(w)\right)}\right],
\end{aligned}
$$

where $\bar{\boldsymbol{V}} \sim \mathrm{Multinomial}(m, M, \bar{\boldsymbol{\mu}}(w))$. The latter step follows as the expectation of a convex sum of Multinomial variables is maximised by variables having the same constants, $\boldsymbol{\mu}_i = \bar{\boldsymbol{\mu}}$ (Hoeffding, 1956). This final term is shown in Corollary 7 of Adams et al. (2022) to be upper bounded by $\xi(M, m)$ uniformly for all w. We divide both sides by m to obtain the theorem statement. □

In showing the simpler form of our bound also we use the following.

Proposition 8.4. *For any $m \geq 3$, $\xi(3, m) \leq 2m$.*

Proof. For $M = 3$ the upper bound in Theorem 8.5 evaluates to

$$
\frac{1}{2} e^{\frac{1}{12m}}\left(1 + \frac{3}{\sqrt{m}} + \frac{6}{\pi m}\right) \cdot m.
$$

The right hand part of this is a decreasing function of m and less than 2 for $m \geq M = 3$. □

Proof of Theorem 8.1. Set $M = 3$ and

$$
\boldsymbol{U}_i(w) = \begin{bmatrix} \max(\tilde{\ell}(w, Z_{1:i}), 0) \\ \max(-\tilde{\ell}(w, Z_{1:i}), 0) \\ 1 - |\tilde{\ell}(w, Z_{1:i})| \end{bmatrix}
$$

in Theorem 8.5, and bound $\xi(3, m)$ with Proposition 8.4. This gives the first part of the statement. For the second, we set $U_i = \mathbb{E}_{W\sim Q_i^*}[\ell(W, Z_i)]$ in Theorem 8.3 and note that $\mu_i = \mathbb{E}_{W\sim Q_i^*}\mathbb{E}_{Z\sim\mathcal{D}}[\ell(W, Z)]$ since $Q_i^\star$ is independent of Z_i. The final part is obtained through a union bound of the two events. □

Proof of Theorem 8.2. The proof is the same as the first part of the proof of Theorem 8.1. □

Proof of Proposition 8.1. We know that $\mathrm{kl}([u,0,1-u]\|\boldsymbol{r}) \geq \mathrm{kl}(u\|r_1)$ by Adams et al. (2022, Proposition 9), with equality when $r_2 = 0$. Therefore we can set $r_2 = 0$ without making it any more difficult to satisfy the constraint $\mathrm{kl}([u,0,1-u]\|\boldsymbol{r}) \leq b$. In this case

$$\phi([u,0,1-u],b) = \sup\{r_1 : \mathrm{kl}(u\|r_1) \leq b\}$$

which is the definition of kl^{-1}. □

Proof of Proposition 8.3. Firstly, we recall (Adams et al., 2022, Proposition 9) that $\mathrm{kl}(\boldsymbol{u}\|\boldsymbol{v}) \geq \mathrm{kl}(u_i\|v_i)$ for any i. This immediately gives the first inequality upon inversion, if we note that $v_1 - v_2 \leq c$ for all $\mathrm{kl}(\boldsymbol{u}\|\boldsymbol{v}) \leq b$ implies that $\phi(\boldsymbol{u},b) \leq c$. The first term is bounded with $v_1 \leq \mathrm{kl}^{-1}(u_1\|b) \leq u_1 + \sqrt{2bu} + 2b$ as in the relaxation of Maurer's bound. Next we know that by Taylor's theorem, for any lower bound $0 \leq p < q \leq 1$, there exists $s \in [p,q]$ such that

$$\mathrm{kl}(q\|p) = \frac{(p-q)^2}{2s(1-s)} \geq \frac{(p-q)^2}{2q}.$$

Therefore

$$-v_2 \leq -u_2 + \sqrt{2bv_2}.$$

The proof is completed by summing these and applying the bound $\sqrt{a} + \sqrt{b} \leq \sqrt{2(a+b)}$ for $a,b \geq 0$ (square both sides and subtract $a+b$ to reduce this to Young's inequality). □

8.5 Experiments

In this section we empirically compare our bound to that of Maurer (2004) and the Unexpected Bernstein (Mhammedi et al., 2019), with a particular focus on the tightening arising through de-biasing by online estimators. For completeness, we give the exact statements of the bounds used in Section 8.B, along with the procedure for calculating our bound.

We replicate the experimental setup of Mhammedi et al. (2019), looking which looks at classification with the 0-1 loss by logistic regression of UCI datasets.

The data space is $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \{0,1\}$. Our hypotheses take the form $h_w(x) = \mathbf{1}_{\psi(w\cdot x) > \frac{1}{2}}$, where $\mathbf{1}$ is the indicator function and $\psi(t) = 1/(1+e^{-t})$ is the standard logistic function. The 0-1 loss can be written as

$$\ell(w,(x,y)) = |y - \mathbf{1}_{\psi(w\cdot x) > \frac{1}{2}}|.$$

Specifically, we look at learning w by regularised logistic regression, which (for sample S and regularisation constant λ) outputs

$$\mathrm{LGR}_\lambda(S) = \operatorname*{argmin}_{w \in \mathcal{W}} \frac{\lambda\|w\|^2}{2} + \frac{1}{|S|}\sum_{(x,y)\in S} \sigma_w(x,y)$$

with

$$\sigma_w(x,y) = -y\log\psi(w\cdot x) - (1-y)\log(1-\psi(w\cdot x)).$$

Dataset	Test	Maurer	UB	Ours
Haberman	0.273	**0.415**	0.583	0.501
Breast-C	0.037	**0.139**	0.208	0.164
Tictactoe	0.043	**0.214**	0.369	0.245
Banknote	0.050	**0.129**	0.192	0.136
kr-vs-kp	0.045	0.167	0.247	**0.164**
Spambase	0.169	0.324	0.501	**0.306**
Mushroom	0.003	**0.055**	0.082	0.056
Adult	0.170	0.234	0.384	**0.211**

Table 8.1: Test error of $\mathrm{LGR}_\lambda(S)$, and bounds for $Q(S)$ with optimised σ as obtained on the UCI datasets listed. The datasets are ranked in order of size, from least examples to most. The bounds evaluated are Maurer's small-kl bound, the unexpected Bernstein bound and our Theorem 8.1, with the latter two using online estimators for de-biasing as described.

This is solved empirically using the L-BFGS algorithm (Liu and Nocedal, 1989).

We set δ=0.05 and λ=0.01 on all datasets. In our bounds we choose posterior $Q(S) = \mathcal{N}(\mathrm{LGR}_\lambda(S), \sigma^2 I)$, with $\sigma^2 \in \{1/2, \ldots, 1/2^J : J = \lceil \log_2 m \rceil\}$ chosen to minimise the bound being considered. For our prior we fix $P = \mathcal{N}(0, \sigma_0^2 I)$. Note that we are not using data-dependent priors as originally studied in Mhammedi et al. (2019), in order to isolate the effect of de-biasing; data-dependent PAC-Bayes priors are a rich topic in their own right.

The sequence of online estimators for our bound and the Unexpected Bernstein are chosen as the deterministic predictors outputted by $\mathrm{LGR}_\lambda(Z_{1:i-1})$; for computational reasons we update these only after every 150 examples, so that each new online estimator predicts the next 150 points. For the first 150 data points the online estimators are not yet effective so we simply choose them to have zero error, which does not change the bound on the loss of the online estimators.

The experiments use several UCI datasets, encoded and pre-processed using the same methods as Mhammedi et al. (2019). Specifically, we encode categorical variables in $0-1$ vectors (increasing the effective dimension of the feature space), remove any instances with missing features, and scale each feature to have values in $[-1, 1]$. Experiments are repeated 20 times with different data shuffling and test-train allocation, and expectation with respect to Gaussian variables are evaluated using Monte Carlo estimates.

Discussion. Empirically we observe that our bound more effectively leverages the de-biasing of online estimators than the unexpected Bernstein, providing a tighter numerical guarantee

in every case. On the smaller datasets it is somewhat weaker than Maurer's bound, but it is close to or better than it on the larger datasets. This arises because when the number of examples is very small, the online estimators are poor surrogates for the final posterior, and the de-biasing term is only weakly correlated with the loss. On the larger datasets, the de-biasing process is more effective and our bound is the tightest.

8.6 Summary

In Theorem 8.1 we have provided a new PAC-Bayesian bound which can be used alongside an extension of excess losses. In particular, this extension of the excess loss is able to use information about the difficulty of examples in a pseudo-online fashion, as the learning algorithm passes over the dataset. This minimises the variance of our generalised excess loss. Our new bound is able to leverage this reduced variance to obtain tighter overall generalisation bounds and fast rates under broader settings.

By harnessing the power of online estimators and small-kl-based bounds in a new way, we have provided a new direction for numerical and theoretical improvements in PAC-Bayes bounds. Information about the difficulty of examples is most easily used for stable algorithms in our framework, which links nicely to further ideas like the complimentary use of data-dependent or distribution-dependent priors. In future work these same ideas could also be explored in an information-theoretic generalisation setting, such as the CMI setting of Steinke and Zakynthinou (2020); and extended to time-uniform bounds Haddouche and Guedj (2022).

8.A Additional proofs and theorems

8.A.1 Proof of proposition 8.2

Proof. We begin with the Donsker-Varadhan variational definition of the KL divergence for probability measures μ, ν on $\mathcal{A}$:

$$\mathrm{KL}(\mu,\nu) = \sup_{g:\mathcal{A}\to\mathbb{R}} \mathop{\mathbb{E}}_{A\sim\mu}[g(A)] - \log \mathop{\mathbb{E}}_{A\sim\nu}[\exp\circ g(A)].$$

Adapting to the case of the excess loss, $|\mathcal{A}| = 3$ and the probabilities of the possible events are measured by $\boldsymbol{q}, \boldsymbol{p} \in \Delta_{3-1}$. The set of real-valued functions on this space can be parameterised by a vector of values for each event, $\boldsymbol{r} \in \mathbb{R}^3$, so that the above equation is expressible as

$$\begin{aligned}
\mathrm{kl}(\boldsymbol{q},\boldsymbol{p}) &= \sup_{\boldsymbol{r}\in\mathbb{R}^3} \sum_{i=1}^{3} q_i r_i - \log\left(\sum_{i=1}^{3} p_i e^{r_i}\right) \\
&= \sup_{\boldsymbol{r}\in\mathbb{R}^3} q_1(r_1-r_3) + q_2(r_2-r_3) + r_3 - \log\left(e^{r_3}\left(p_1(e^{r_1-r_3}-1) + p_2(e^{r_2-r_3}-1)+1\right)\right) \\
&= \sup_{\boldsymbol{r}\in\mathbb{R}^3} q_1(r_1-r_3) + q_2(r_2-r_3) - \log\left(p_1(e^{r_1-r_3}-1) + p_2(e^{r_2-r_3}-1)+1\right).
\end{aligned}$$

Now we introduce the following reparamterisation of $r_1 - r_3 = \log(1-\eta), r_2 - r_3 = \log(1+\eta')$ so that

$$\mathrm{kl}(\boldsymbol{q},\boldsymbol{p}) = \sup_{\eta<1,\eta'>-1} -\log\left((-\eta p_1 + \eta' p_2 + 1)e^{-q_1\log(1-\eta) - q_2\log(1+\eta')}\right)$$

which can be re-arranged to give the following form very close to our final result

$$\sup_{\eta<1,\eta'>-1} \eta p_1 - \eta' p_2 - \left(1 - e^{q_1\log(1-\eta) + q_2\log(1+\eta') - \mathrm{kl}(\boldsymbol{q},\boldsymbol{p})}\right) = 0.$$

If we then relax the result by insisting that $\eta' = \eta \in (0,1)$, we find that

$$p_1 - p_2 \leq \inf_{\eta\in(0,1)} \frac{1}{\eta}\left(1 - e^{q_1\log(1-\eta) + q_2\log(1+\eta) - \mathrm{kl}(\boldsymbol{q},\boldsymbol{p})}\right)$$

which gives the first part of our result.

Rearranging terms,

$$\begin{aligned}
q_1\log(1-\eta) + q_2\log(1+\eta) &= q_1 - q_2 + q_1\left(\frac{-\log(1-\eta)-\eta}{\eta}\right) + q_2\left(\frac{-\log(1+\eta)+\eta}{\eta}\right) \\
&\leq q_1 - q_2 + (q_1+q_2)\left(\frac{-\log(1-\eta)-\eta}{\eta}\right) \\
&= q_1 - q_2 + c_\eta(q_1+q_2)
\end{aligned}$$

where it is easily verified that $-\log(1+\eta)+\eta \leq -\log(1-\eta)-\eta$ for $\eta > 0$.

Substitution of the definitions gives our proposition. □

8.A.2 Proof of theorem 8.3

Proof. We show that

$$\mathbb{P}\left(\bar{U} \geq \bar{\mu} + t\right) \leq e^{-m \cdot \mathrm{kl}(\bar{\mu}+t \| \bar{\mu})}. \tag{8.8}$$

From this, the proof of Theorem 4 in Biggs (2022) implies our theorem statement. By Markov's inequality and the convexity of $t \mapsto e^{ct}$,

$$\begin{aligned}
\mathbb{P}\left(\bar{U} \geq \bar{\mu} + t\right) &\leq \mathbb{P}\left(e^{m\lambda\bar{U}} \geq e^{m\lambda(\bar{\mu}+t)}\right) \\
&\leq e^{-m\lambda(\bar{\mu}+t)}\, \mathbb{E}\left[e^{m\lambda\bar{U}}\right] \\
&\leq e^{-m\lambda(\bar{\mu}+t)}\, \mathbb{E}\left[\prod_{i=1}^{m} e^{\lambda U_i}\right] \\
&\leq e^{-m\lambda(\bar{\mu}+t)}\, \mathbb{E}\left[\prod_{i=1}^{m} (1 - U_i + U_i e^{\lambda})\right] \\
&\leq e^{-m\lambda(\bar{\mu}+t)}\, \mathbb{E}\left[\prod_{i=1}^{m} (1 - \mu_i + \mu_i e^{\lambda})\right]
\end{aligned}$$

where in the final step we have used the same telescoping property of conditional expectations as in the proof of Theorem 8.4. By the arithmetic-geometric mean inequality, the product term is upper bounded by

$$\left(\frac{1}{m}\sum_{i=1}^{m}(1 - \mu_i + \mu_i e^{\lambda})\right)^m = (1 - \bar{\mu} + \bar{\mu} e^{\lambda})^m.$$

Substitution shows that the probability above is upper bounded by

$$\left(\frac{1 - \bar{\mu} + \bar{\mu} e^{\lambda}}{e^{\lambda(\bar{\mu}+t)}}\right).$$

Optimising this bound w.r.t. λ gives the form on Eq. (8.8). □

8.A.3 PAC-Bayes unexpected bernstein with generalised excess loss

In this section we reproduce the following central result of Mhammedi et al. (2019) in the form used by our empirical comparison (which uses de-biasing but not informed priors).

Theorem 8.6 (PAC-Bayes unexpected Bernstein Excess Loss)**.** *For loss $\ell \in [0,1]$, for any fixed $\eta \in (0,1)$ and prior $P \in \mathscr{P}(\mathcal{W})$, with probability at least $1-\delta$ over the sample S simultaneously for any $Q \in \mathscr{P}(\mathcal{W})$*

$$\mathcal{E}(Q) - \widehat{\mathcal{E}}(Q) \leq c_\eta \widehat{V}(Q) + \frac{\mathrm{KL}(Q,P) + \log\frac{1}{\delta}}{m\eta}.$$

Here $c_\eta = \frac{\eta + \log(1-\eta)}{\eta}$,

$$\mathcal{E}(Q) := \mathcal{L}(Q) - \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{Z} \mathop{\mathbb{E}}_{W \sim Q_i^\star}[\ell(W,Z)],$$

$$\widehat{\mathcal{E}}(Q) := \widehat{\mathcal{L}}(Q) - \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z)],$$
$$\widehat{V}(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m} |\ell(W,Z_i) - \mathop{\mathbb{E}}_{W'\sim Q_i^\star}\ell(W',Z_i)|^2\right].$$

We note that if the online estimators are fixed these quantities reduce to the standard excess risk terms. In order to prove this result, we first state and prove some intermediate results.

Proposition 8.5 (Unexpected Bernstein Lemma; Mhammedi et al., 2019, Lemma 13)**.** *Let $U \le 1$ a.s.; then for any $\eta \in (0,1)$*

$$\mathbb{E}\, e^{\eta(\mathbb{E}[U]-U-c_\eta U^2)} \le 1$$

where $c_\eta = \frac{\eta+\log(1-\eta)}{\eta}$.

Proof. For $t<1$, define the decreasing function

$$f(t) = \frac{\log(1-t)+t}{t^2}.$$

Let $u \le 1$ and $\eta \in (0,1)$, so that $u\eta \le \eta < 1$. Since f is decreasing,

$$f(\eta) \le f(u\eta) \implies \frac{\log(1-\eta)+\eta}{\eta^2} \le \frac{\log(1-u\eta)+u\eta}{(u\eta)^2} \implies \exp\big(\eta c_\eta u^2 - \eta u\big) \le 1-u\eta.$$

Setting $u = U$ and taking the expectation, and using $1-t \le e^{-t}$,

$$\mathbb{E}\exp\big(\eta c_\eta U^2 - \eta U\big) \le 1 - \mathbb{E}[U]\eta \le e^{-\mathbb{E}[U]},$$

dividing through by the right hand side gives the result. □

We also give the following unexpected Bernstein counterpart of Theorem 8.5 which can be used to trivially prove the main result.

Theorem 8.7. *Let $U_1(w),\dots,U_m(w)$ be a sequence of random bounded functions, valued in $[0,1]$, such that*

$$\mathbb{E}[U_i(w)|U_1(w),\dots,U_{i-1}(w)] = \mu_i(w)$$

for $i=1,\dots,m$ and all $w\in\mathcal{W}$. Define

$$\widehat{U}(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m} U_i(W)\right] \qquad \text{and} \qquad \bar{\mu}(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m}\mu_i(W)\right].$$

For any fixed $P\in\mathscr{P}(\mathcal{W}), \delta\in(0,1), \eta\in(0,1)$, with probability at least $1-\delta$ (over all U_i), simultaneously for all $Q\in\mathscr{P}(\mathcal{W})$,

$$\bar{\mu}(Q) - \widehat{U}(Q) \le \frac{c_\eta}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{W\sim Q}|U_i(W)|^2 + \frac{\mathrm{KL}\,(Q,P)+\log\frac{1}{\delta}}{m}$$

where $c_\eta = \frac{\eta+\log(1-\eta)}{\eta}$.

Proof. Firstly, we combine Proposition 8.5 with recursion of conditional expectations to find that

$$\begin{aligned}
&\mathbb{E}\exp\left(\eta\sum_{i=1}^{m}(\mu_i - U_i - c_\eta U_i^2)\right) \\
&\quad\leq \mathbb{E}\left[\prod_{i=1}^{m}\exp\left(\eta(\mu_i - U_i - c_\eta U_i^2)\right)\right] \\
&\quad\leq \mathop{\mathbb{E}}_{U_{1:m-1}}\left[\prod_{i=1}^{m-1}\exp\left(\eta(\mu_i - U_i - c_\eta U_i^2)\right)\cdot \mathop{\mathbb{E}}_{U_m}\left[\exp\left(\eta(\mu_m - U_m - c_\eta U_m^2)\right)|U_{1:m-1}\right]\right] \\
&\quad\leq \mathop{\mathbb{E}}_{U_{1:m-1}}\left[\prod_{i=1}^{m-1}\exp\left(\eta(\mu_i - U_i - c_\eta U_i^2)\right)\right] \\
&\quad\leq 1.
\end{aligned}$$

Next, as in the proof of Theorem 8.5, we combine this with Donsker-Varadhan, Markov's inequality and the independence of P from $U_{1:m}$ to find the following holds with probability at least $1-\delta$ for all Q:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{W\sim Q}\left[\eta\sum_{i=1}^{m}(\mu_i(W) - U_i(W) - c_\eta U_i(W)^2)\right] - \mathrm{KL}\left(Q, P\right) \\
&\quad\leq \log \mathop{\mathbb{E}}_{W\sim P}\left[\eta\sum_{i=1}^{m}(\mu_i(W) - U_i(W) - c_\eta U_i(W)^2)\right] \\
&\quad\leq \log \mathop{\mathbb{E}}_{U_{1:m}}\mathop{\mathbb{E}}_{W\sim P}\left[\eta\sum_{i=1}^{m}(\mu_i(W) - U_i(W) - c_\eta U_i(W)^2)\right] \\
&\quad\leq \log \frac{1}{\delta}\mathop{\mathbb{E}}_{W\sim P}\mathop{\mathbb{E}}_{U_{1:m}}\left[\eta\sum_{i=1}^{m}(\mu_i(W) - U_i(W) - c_\eta U_i(W)^2)\right] \\
&\quad\leq \log \frac{1}{\delta}.
\end{aligned}$$

Dividing both sides through by $m\eta$ gives the result. □

Proof of Theorem 8.6. In Theorem 8.7, set

$$U_i(w) = \ell(w, Z_i) - \mathop{\mathbb{E}}_{W\sim Q_i^\star}\ell(W, Z_i)$$

for each i, which is bounded above by 1. □

8.A.4 Relaxation of small kl

In the following, we prove a relaxation of the inverse small kl which leads to a form much more similar to the unexpected Bernstein, and is used later to motivate our experimental setup.

Proposition 8.6. *For $0 \leq q < 1$ and $b > 0$,*

$$\mathrm{kl}^{-1}\left(q\|b\right) < \inf_{\eta\in(0,1)}\left[(1+c_\eta)q + \frac{b}{\eta}\right]$$

where $c_\eta := -\frac{\eta+\log(1-\eta)}{\eta}$.

In Germain et al. (2009, Proposition 2.1) it is proved that

Proposition 8.7. *For any* $0 \leq q \leq p < 1$,

$$\sup_{C>0} \left[C\Phi_C(p) - Cq\right] = \mathrm{kl}(q\|p)$$

where

$$\Phi_C(p) = -\frac{1}{C}\log(1-p+pe^{-C}).$$

Proof of Proposition 8.6. For any $C > 0$, $C\Phi_C(p) - Cq \leq \mathrm{kl}(q\|p)$, and thus

$$\mathrm{kl}^{-1}(q\|b) = \sup\{p \in (0,1) : \mathrm{kl}(q\|p) \leq b\} \leq \sup\{p \in (0,1) : C\Phi_C(p) - Cq \leq b\} = \Phi_C^{-1}(q + b/c)$$

with the latter step following by the invertibility of Φ_C. Since $1 - e^{-t} \leq t$ with equality only at $t = 0$,

$$\Phi_C^{-1}(t) = \frac{1-e^{-Ct}}{1-e^{-C}} \leq \frac{Ct}{1-e^{-C}}.$$

As $b > 0$ and $q \geq 0$, we have $q + b/c \neq 0$ and therefore

$$\Phi_C^{-1}(q + b/c) < \frac{Cq+b}{1-e^{-C}}.$$

Introducing $\eta = 1 - e^{-C} \in (0,1)$ (so that $C = -\log(1-\eta)$),

$$\frac{Cq+b}{1-e^{-C}} = \frac{-\log(1-\eta)}{\eta}q + \frac{b}{\eta} = (1+c_\eta)q + \frac{b}{\eta}.$$

Chaining these results, and taking an infimum of both sides over the free variable $\eta \in (0,1)$ completes the proof. □

8.B Full bounds used in experiments

In our experiments, we use the following bounds, obtained through Theorem 8.1 or (a slight refinement of) Theorem 8.6 with online estimators. In the unexpected Bernstein case, we combine the result with a grid over possible values of η, in the same way as the original paper. over possible values of η.

Theorem 8.8 (Generalisation Loss Bound)**.** *Fix* $\mathcal{D} \in \mathscr{P}(\mathcal{Z}), P \in \mathscr{P}(\mathcal{W}), \delta \in (0,1)$, *and* $\ell : \mathcal{Z} \times \mathcal{W} \to [0,1]$. *With probability at least* $1-\delta$ *over* $Z_{1:m} = S \sim \mathcal{D}^m$, *for a sequence of online estimators* $Q_i^\star \in \mathscr{P}(\mathcal{W})$ *with* $i = 1, \ldots, m$, *where each depenends only on the* $i-1$ *examples* $Z_{1:i-1}$, *and for any posterior* $Q \in \mathscr{P}(\mathcal{W})$, *the following holds*

$$\mathcal{L}(Q) \leq \phi\left(\begin{bmatrix} \widehat{\mathcal{E}}_+(Q) \\ \widehat{\mathcal{E}}_-(Q) \\ 1 - \widehat{\mathcal{E}}_+(Q) - \widehat{\mathcal{E}}_-(Q) \end{bmatrix}, \frac{\mathrm{KL}(Q,P) + \log\frac{4m}{\delta}}{m}\right) + \mathrm{kl}^{-1}\left(\frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z_i)] \middle\| \frac{\log\frac{2}{\delta}}{m}\right)$$

with

$$\widehat{\mathcal{E}}_+(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^m \max(\ell(W,Z_i) - \mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z_i)], 0)\right],$$

$$\widehat{\mathcal{E}}_-(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m}\max(\mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z_i)]-\ell(W,Z_i),0)\right].$$

Theorem 8.9 (Unexpected Bernstein for Generalisation Loss)**.** *Fix $\mathcal{D}\in\mathscr{P}(\mathcal{Z}), P\in\mathscr{P}(\mathcal{W}), \delta\in(0,1)$, and $\ell:\mathcal{Z}\times\mathcal{W}\to[0,1]$. With probability at least $1-\delta$ over $Z_{1:m}=S\sim\mathcal{D}^m$, for a sequence of online estimators $Q_i^\star\in\mathscr{P}(\mathcal{W})$ with $i=1,\dots,m$, where each depenends only on the $i-1$ examples $Z_{1:i-1}$, and for any posterior $Q\in\mathscr{P}(\mathcal{W})$, the following holds*

$$\mathcal{L}(Q)\le\widehat{\mathcal{L}}(Q)+\inf_{\eta\in\mathcal{G}}\left[c_\eta\widehat{V}(Q)+\frac{\mathrm{KL}(Q,P)+\log\frac{2K}{\delta}}{m\eta}\right]+\mathrm{kl}^{-1}\left(\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z_i)]\middle\|\frac{\log\frac{2}{\delta}}{m}\right)$$

where

$$\widehat{V}(Q) := \mathop{\mathbb{E}}_{W\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m}|\ell(W,Z_i)-\mathop{\mathbb{E}}_{W'\sim Q_i^\star}\ell(W',Z_i)|^2\right]$$

and

$$\mathcal{G} := \left\{\frac{1}{2},\dots,\frac{1}{2^K} : K=\left\lceil\log_2\left(\frac{1}{2}\sqrt{\frac{n}{\log(1/\delta)}}\right)\right\rceil\right\}.$$

Proof. Combine Theorem 8.6 with the bound on $\mathcal{L}^\star$ in Theorem 8.1 and take a union bound over the grid $\mathcal{G}$. □

8.B.1 Bounding the online estimator loss

We note here that Mhammedi et al. (2019) originally used an alternative bound to go from the excess loss to the Generalisation risk. Instead of using the bound on $\mathcal{L}^\star$ in Theorem 8.1 as we do above, they used the bound

$$\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{Z}\mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z)]\le\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z_i)]+\inf_{\eta\in\mathcal{G}}\left[\frac{c_\eta}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{W\sim Q_i^\star}|\ell(W,Z_i)|^2+\frac{\mathrm{KL}(Q,P)+\log\frac{|\mathcal{G}|}{\delta}}{m\eta}\right].$$

In the case of the 0-1 misclassification loss used in our experimental setup, where $\ell^2=\ell$, this simplifies to the following:

$$\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{Z}\mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z)]\le\inf_{\eta\in\mathcal{G}}\left[(1+c_\eta)\left(\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{W\sim Q_i^\star}\ell(W,Z_i)\right)+\frac{\mathrm{KL}(Q,P)+\log\frac{|\mathcal{G}|}{\delta}}{m\eta}\right].$$

As we find through Proposition 8.6, our Theorem 8.1 implies that

$$\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{Z}\mathop{\mathbb{E}}_{W\sim Q_i^\star}[\ell(W,Z)]<\inf_{\eta\in(0,1)}\left[(1+c_\eta)\left(\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{W\sim Q_i^\star}\ell(W,Z_i)\right)+\frac{\mathrm{KL}(Q,P)+\log\frac{1}{\delta}}{m\eta}\right]$$

which is strictly stronger (for example, our result holds simultaneously over all $\eta\in(0,1)$ with no grid size penalty, and even for the optimal η this bound is slacker). Therefore, the second part of our result Theorem 8.1 represents a significant contribution, that can be leveraged in combination with the original unexpected Bernstein bound to tighten it in the case of 0-1 losses (and it may also give tighter numerical bounds with some other loss functions also).

We note that Theorem 8.1 can also easily be combined with the backwards-forwards dataset split used by Mhammedi et al. (2019).

In order to make the fairest empirical comparison between the effects of de-biasing on our bound versus the unexpected Bernstein, we therefore use our bound in the comparison.

8.B.2 Calculation of inverse kl ϕ

Based directly on Proposition 11 in Adams et al. (2022), we give the following proposition, which can be used to calculate $\phi(\boldsymbol{u}, b)$.

Proposition 8.8. *Fix $\boldsymbol{u} \in \Delta_{3-1}$ and $b > 0$. Define the increasing function*

$$f_{\boldsymbol{u}}(s) := \log\left(u_1 + \frac{u_2}{1+2s} + \frac{u_3}{1+s}\right) + u_2 \log(1+2s) + u_3 \log(1+s)$$

and its inverse $s^\star := f_{\boldsymbol{u}}^{-1}(b)$. If $t := -\exp(-s^\star) - 1$, then

$$\phi(\boldsymbol{u}, b) = \frac{\frac{u_1}{t+1} - \frac{u_2}{t-1}}{\frac{u_1}{t+1} + \frac{u_2}{t-1} + \frac{u_3}{t}}.$$

Computationally, we can find the inverse $s^\star$ by a simple bisection-search or Newton's method. Our slight re-parameterisation (where we write f in terms of s instead of $t = -e^{-s} - 1$ as used by Adams et al., 2022) of the original result makes this calculation considerably more numerically stable.

We note as an aside that once we have calculated $s^\star$, we can also use it to find the gradients $\frac{\partial}{\partial u_i}\phi(\boldsymbol{u}, b)$ and $\frac{\partial}{\partial b}\phi(\boldsymbol{u}, b)$, which may be useful when directly optimising the bound as an objective.

8.C Further experimental details

Below we provide additional information about the datasets used and tabulated empirical results. Our code is available at https://github.com/biggs/tighter-pac-bayes-difficulty.

Dataset	Size	Test	Maurer	UB	Ours
Haberman	306	0.2726 ± 0.0388	0.4140 ± 0.0114	0.5829 ± 0.0176	0.5020 ± 0.0113
Breast-C	699	0.0371 ± 0.0133	0.1387 ± 0.0049	0.2079 ± 0.0070	0.1635 ± 0.0068
Tictactoe	958	0.0427 ± 0.0151	0.2148 ± 0.0056	0.3683 ± 0.0215	0.2456 ± 0.0069
Banknote	1372	0.0498 ± 0.0113	0.1292 ± 0.0033	0.1926 ± 0.0075	0.1359 ± 0.0038
kr-vs-kp	3196	0.0449 ± 0.0084	0.1670 ± 0.0023	0.2466 ± 0.0039	0.1633 ± 0.0029
Spambase	4601	0.1694 ± 0.0132	0.3238 ± 0.0027	0.5015 ± 0.0082	0.3054 ± 0.0032
Mushroom	8124	0.0026 ± 0.0013	0.0551 ± 0.0007	0.0820 ± 0.0015	0.0565 ± 0.0009
Adult	32561	0.1696 ± 0.0045	0.2341 ± 0.0013	0.3842 ± 0.0024	0.2108 ± 0.0014

Table 8.2: Test error of $\mathrm{LGR}_\lambda(S)$, and bounds for $Q(S)$ with optimised σ as obtained on the UCI datasets listed. The datasets are ranked in order of size, from least examples to most. This size (listed) is the dataset size before the 20% test set is removed. The bounds evaluated are Maurer's small-kl bound, the unexpected Bernstein bound and our Theorem 8.1 as described in Section 8.B, with the latter two using online estimators for de-biasing as in Section 8.5. Results are an average of 20 runs with standard errors provided.

Chapter 9

Conclusion

This thesis has examined generalisation within machine learning from a variety of different angles, attempting to address the shortfalls of classical learning theory in settings such as deep learning, where it fails to make useful predictions. Our research aims to improve this by developing new and meaningful generalisation bounds.

Our diverse research contributions are united by this goal, the use of PAC-Bayesian methods, and a number of recurring themes. To begin, all of the works focused on providing empirically tighter bounds, whether for specific methods or in general (as in Chapter 8). Chapters 5 to 7 focused on deriving non-randomised bounds (using margins or majority voting) via a specially-chosen randomised proxy. These approaches yielded the first non-vacuous bounds for deterministic shallow neural networks on real-world data and provided state-of-the-art guarantees for majority voting on finite ensembles of classifiers. Chapters 4 and 7 looked at improving PAC-Bayesian methods as objectives, advancing theoretical understanding and providing practical tools for improved model training.

These developments have broad and fundamental implications. By improving our understanding of generalisation, particularly in settings where traditional bounds suggest overfitting, we build a foundation for more robust and reliable machine learning systems. This is crucial for fostering trust and safeguarding their adoption in sectors such as healthcare and finance, where reliability and trustworthiness is paramount. Public policy around these systems cannot be effectively developed without the transparency of genuinely understanding how they operate.

The publications forming this thesis have already influenced subsequent work, as evidenced by citations and follow-on research. A few other potential future directions—in addition to those already suggested in the chapters themselves—that could follow include:

- Further development of aggregation as a training method for learning stochastic neural networks, building on the results in Chapter 4 and closely related follow on work in Clerico et al. (2022).

- By finding ways to express deeper neural networks as ensembles, there is potential to extend the non-vacuous results in Chapter 6 to deeper networks or different architectures. In particular, the proof method for shallow GELU networks in Chapter 6 is related to the gated linear units (Shazeer, 2020) with layer normalisation (Ba et al., 2016) used by modern transformer architechures (Vaswani et al., 2017).
- The concentration-based method for obtaining margin bounds formalised in Chapter 5 and used to great effect in Chapter 7 could be extended to other settings, such as mutual information bounds (Russo and Zou, 2016; Xu and Raginsky, 2017, and many others), or applied to different function classes.
- The new general bound for excess risk in Chapter 8 could be adapted to other frameworks such as mutual information, or with other PAC-Bayes derived bounds (such as the margin bounds elsewhere in this thesis) to obtain even tighter bounds; as a technique it has the potential to make many existing bounds tighter.

In conclusion, this thesis represents a significant step towards a deeper understanding of generalisation in machine learning. By developing new theoretical tools and demonstrating their practical relevance, we contribute to the ongoing effort to build more robust, trustworthy, and transparent machine learning systems. This work underscores the importance of rigorous theoretical foundations in advancing the field and ensuring the responsible deployment of AI technologies.

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. URL https://www.tensorflow.org/.

Reuben Adams, John Shawe-Taylor, and Benjamin Guedj. Controlling confusion via generalisation bounds. *CoRR*, abs/2202.05560, 2022.

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. 2021. URL https://www.arxiv.org/abs/2110.11216.

Pierre Alquier and Gérard Biau. Sparse single-index model. *Journal of Machine Learning Research*, 14:243–280, 2013a.

Pierre Alquier and Gérard Biau. Sparse single-index model. *J. Mach. Learn. Res.*, 14 (1):243–280, 2013b. doi: 10.5555/2567709.2502589. URL https://dl.acm.org/doi/10.5555/2567709.2502589.

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of gibbs posteriors. *J. Mach. Learn. Res.*, 17:239:1–239:41, 2016. URL http://jmlr.org/papers/v17/15-290.html.

Amiran Ambroladze, Emilio Parrado-Hernández, and John Shawe-Taylor. Tighter pac-bayes bounds. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 9–16. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Martin Anthony and John Shawe-Taylor. A result of vapnik with applications. *Discret. Appl. Math.*, 47(3):207–217, 1993. doi: 10.1016/0166-218X(93)90126-9. URL https://doi.org/10.1016/0166-218X(93)90126-9.

Martin Anthony, Peter L Bartlett, Peter L Bartlett, et al. *Neural network learning: Theoretical foundations*, volume 9. cambridge university press Cambridge, 1999.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 254–263. PMLR, 2018. URL http://proceedings.mlr.press/v80/arora18b.html.

Lei Jimmy Ba and Brendan J. Frey. Adaptive dropout for training deep neural networks. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3084–3092, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/7b5b23f4aadf9513306bcd59afb6e4c9-Abstract.html.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Arindam Banerjee, Tiancong Chen, and Yingxue Zhou. De-randomized pac-bayes margin bounds: Applications to non-convex and non-smooth predictors. *CoRR*, abs/2002.09956, 2020.

Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In Bernard Schölkopf, Christopher J C Burges, and Alexander J Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, USA, 1998.

Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*,

pages 6240–6249, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/b22b257ad0519d4500539da3c8bcf4dd-Abstract.html.

Robert M. Bell and Yehuda Koren. Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79, dec 2007. ISSN 1931-0145. doi: 10.1145/1345448.1345465.

Shai Ben-David and Hans Ulrich Simon. Efficient learning of linear perceptrons. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 189–195. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/2000/hash/39027dfad5138c9ca0c474d71db915c3-Abstract.html.

Felix Biggs. A note on the efficient evaluation of PAC-Bayes bounds. *CoRR*, abs/2209.05188, 2022. doi: 10.48550/arXiv.2209.05188.

Felix Biggs and Benjamin Guedj. Differentiable PAC–Bayes objectives with partially aggregated neural networks. *Entropy*, 23(10), 2021. ISSN 1099-4300. doi: 10.3390/e23101280. URL https://www.mdpi.com/1099-4300/23/10/1280.

Felix Biggs and Benjamin Guedj. On margins and derandomisation in pac-bayes. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 3709–3731. PMLR, 2022a. URL https://proceedings.mlr.press/v151/biggs22a.html.

Felix Biggs and Benjamin Guedj. Non-vacuous generalisation bounds for shallow neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 1963–1981. PMLR, 2022b. URL https://proceedings.mlr.press/v162/biggs22a.html.

Felix Biggs and Benjamin Guedj. Tighter pac-bayes generalisation bounds by leveraging example difficulty. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent, editors, *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 8165–8182. PMLR, 2023. URL https://proceedings.mlr.press/v206/biggs23a.html.

Felix Biggs, Valentina Zantedeschi, and Benjamin Guedj. On margins and generalisation for voting classifiers. In *NeurIPS*, 2022. doi: 10.48550/arXiv.2206.04607.

Felix Biggs, Antonin Schrab, and Arthur Gretton. MMD-FUSE: Learning and combining kernels for two-sample testing without data splitting. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, December 10-16, 2023, New Orleans*, 2023.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France, 2015. PMLR. URL http://proceedings.mlr.press/v37/blundell15.html.

Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375, 2005.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. ISBN 978-0-19-953525-5. doi: 10.1093/acprof:oso/9780199535255.001.0001.

Leo Breiman. Prediction Games and Arcing Algorithms. *Neural Computation*, 11(7): 1493–1517, 10 1999. ISSN 0899-7667. doi: 10.1162/089976699300016106.

Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Olivier Catoni. A pac-bayesian approach to adaptive classification. *Preprint*, 840:2, 2003.

Olivier Catoni. *Statistical Learning Theory and Stochastic Optimization: Ecole d'Eté de Probabilités de Saint-Flour XXXI-2001*, volume 1851 of *Lecture Notes in Mathematics*. Springer, 2004.

Olivier Catoni. Pac-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning. *IMS Lecture Notes Monogr. Ser.*, 56:1–163, 2007. ISSN 0749-2170. doi: 10.1214/074921707000000391.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM, 2016. doi: 10.1145/2939672.2939785.

Eugenio Clerico, George Deligiannidis, and Arnaud Doucet. Conditionally gaussian pac-bayes. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022,*

Virtual Event, volume 151 of *Proceedings of Machine Learning Research*, pages 2311–2329. PMLR, 2022. URL https://proceedings.mlr.press/v151/clerico22a.html.

Eugenio Clerico, George Deligiannidis, and Arnaud Doucet. Wide stochastic networks: Gaussian limit and pac-bayesian training. In Shipra Agrawal and Francesco Orabona, editors, *International Conference on Algorithmic Learning Theory, February 20-23, 2023, Singapore*, volume 201 of *Proceedings of Machine Learning Research*, pages 447–470. PMLR, 2023. URL https://proceedings.mlr.press/v201/clerico23a.html.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995. doi: 10.1007/BF00994018.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006. ISBN 0471241954.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158, 1975.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. In Sham M. Kakade and Ulrike von Luxburg, editors, *COLT 2011 - The 24th Annual Conference on Learning Theory, June 9-11, 2011, Budapest, Hungary*, volume 19 of *JMLR Proceedings*, pages 207–232. JMLR.org, 2011. URL http://proceedings.mlr.press/v19/daniely11a/daniely11a.pdf.

Erik A. Daxberger, Eric T. Nalisnick, James Urquhart Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Bayesian deep learning via subnetwork inference. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2510–2521. PMLR, 2021. URL http://proceedings.mlr.press/v139/daxberger21a.html.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

Monroe D Donsker and SR Srinivasa Varadhan. On a variational formula for the principal eigenvalue for operators with maximum principle. *Proceedings of the National Academy of Sciences*, 72(3):780–783, 1975.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Conference on Uncertainty in Artificial Intelligence 33.*, 2017.

Gintare Karolina Dziugaite and Daniel M. Roy. Data-dependent PAC-Bayes priors via differential privacy. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8440–8450, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/9a0ee0a9e7a42d2d69b8f86b3a0756b1-Abstract.html.

Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M. Roy. In search of robust measures of generalization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/86d7c8a08b4aaa1bc7c599473f5dddda-Abstract.html.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in pac-bayes. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 604–612. PMLR, 2021. URL http://proceedings.mlr.press/v130/karolina-dziugaite21a.html.

Andrew Y. K. Foong, Wessel P. Bruinsma, David R. Burt, and Richard E. Turner. How tight can PAC-Bayes be in the small data regime? In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4093–4105, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/214cfbe603b7f9f9bc005d5f53f7a1d3-Abstract.html.

Andrew Y. K. Foong, Wessel P. Bruinsma, and David R. Burt. A note on the Chernoff bound for random variables in the unit interval. *CoRR*, abs/2205.07880, 2022. doi: 10.48550/arXiv.2205.07880.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *9th International Conference on*

Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Louis Fortier-Dubois, Benjamin Leblanc, Gaël Letarte, François Laviolette, and Pascal Germain. Pac-bayesian learning of aggregated binary activated neural networks with probabilities over representations. In Amílcar Soares, Farhana H. Zulkernine, Renata Dividino, Reihaneh Rabbany, Qiang Ye, David Beach, and Karim Ali, editors, *36th Canadian Conference on Artificial Intelligence, Canadian AI, CANAI 2023, Montreal, Canada, June 5-9, 2023, Proceedings*. Canadian Artificial Intelligence Association, 2023. doi: 10.21428/594757db.4cec07db.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504. URL https://doi.org/10.1006/jcss.1997.1504.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

Wei Gao and Zhi-Hua Zhou. On the doubt about margin explanation of boosting. *Artif. Intell.*, 203:1–18, 2013. doi: 10.1016/j.artint.2013.07.002.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, Montreal, Quebec, Canada, 2009. ACM Press. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553419.

Pascal Germain, Alexandre Lacasse, François Laviolette, Mario Marchand, and Jean-Francis Roy. Risk bounds for the majority vote: from a PAC-Bayesian analysis to a learning algorithm. *J. Mach. Learn. Res.*, 16:787–860, 2015. doi: 10.5555/2789272.2831140.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. PAC-Bayesian theory meets bayesian inference. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1884–1892. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6569-pac-bayesian-theory-meets-bayesian-inference.pdf.

Allan Grønlund, Lior Kamma, and Kasper Green Larsen. Near-tight margin-based generalization bounds for support vector machines. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3779–3788. PMLR, 2020. URL http://proceedings.mlr.press/v119/gronlund20a.html.

Allan Grønlund, Lior Kamma, and Kasper Green Larsen. Margins are insufficient for explaining gradient boosting. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/146f7dd4c91bc9d80cf4458ad6d6cd1b-Abstract.html.

Peter Grünwald, Thomas Steinke, and Lydia Zakynthinou. Pac-bayes, mac-bayes and conditional mutual information: Fast rate bounds that handle general VC classes. In Mikhail Belkin and Samory Kpotufe, editors, *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pages 2217–2247. PMLR, 2021. URL http://proceedings.mlr.press/v134/grunwald21a.html.

Benjamin Guedj. A primer on PAC-Bayesian learning. In *Proceedings of the second congress of the French Mathematical Society*, volume 33, 2019.

Benjamin Guedj and Pierre Alquier. PAC-Bayesian estimation and prediction in sparse additive models. *Electron. J. Statist.*, 7:264–291, 2013a. doi: 10.1214/13-EJS771.

Benjamin Guedj and Pierre Alquier. Pac-bayesian estimation and prediction in sparse additive models. 2013b.

Maxime Haddouche and Benjamin Guedj. Pac-bayes with unbounded losses through supermartingales. *CoRR*, abs/2210.00928, 2022. doi: 10.48550/arXiv.2210.00928.

Steve Hanneke and Aryeh Kontorovich. Stable sample compression schemes: New applications and an optimal SVM margin bound. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Algorithmic Learning Theory, 16-19 March 2021, Virtual Conference, Worldwide*, volume 132 of *Proceedings of Machine Learning Research*, pages 697–721. PMLR, 2021. URL http://proceedings.mlr.press/v132/hanneke21a.html.

Fredrik Hellström and Giuseppe Durisi. Fast-rate loss bounds via conditional information measures with applications to neural networks. In *IEEE International Symposium on Information Theory, ISIT 2021, Melbourne, Australia, July 12-20, 2021*, pages 952–957. IEEE, 2021. doi: 10.1109/ISIT45174.2021.9517731.

Fredrik Hellström, Giuseppe Durisi, Benjamin Guedj, and Maxim Raginsky. Generalization bounds: Perspectives from information theory and pac-bayes. *arXiv preprint arXiv:2309.04381*, 2023.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with Gaussian error linear units. *CoRR*, abs/1606.08415, 2016.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In Lenny Pitt, editor, *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory, COLT 1993, Santa Cruz, CA, USA, July 26-28, 1993*, pages 5–13. ACM, 1993. doi: 10.1145/168304.168306.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Comput.*, 9(1):1–42, 1997. doi: 10.1162/neco.1997.9.1.1.

Wassily Hoeffding. On the distribution of the number of successes in independent trials. *The Annals of Mathematical Statistics*, pages 713–721, 1956.

Yiding Jiang, Pierre Foret, Scott Yak, Daniel M. Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. NeurIPS 2020 competition: Predicting generalization in deep learning. *CoRR*, abs/2012.07976, 2020a.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b. URL https://openreview.net/forum?id=SJgIPJBFvH.

Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 793–800. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/hash/5b69b9cb83065d403869739ae7f0995e-Abstract.html.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. To appear in Mathematics of Deep Learning, Cambridge University Press, 2020. URL https://www.arxiv.org/abs/1710.05468.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2575–2583. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5666-variational-dropout-and-the-local-reparameterization-trick.pdf.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized Variational Inference: Three arguments for deriving new Posteriors. *arXiv preprint arXiv:1904.02063*, December 2019.

Vladimir Koltchinskii and Dmitriy Panchenko. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer, 2000.

Aryeh Kontorovich and Iosif Pinelis. Exact lower bounds for the agnostic probably-approximately-correct (PAC) machine learning model. *CoRR*, abs/1606.08920, 2016. URL http://arxiv.org/abs/1606.08920.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. PMLR, 2020. URL http://proceedings.mlr.press/v119/kristiadi20a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alexandre Lacasse, François Laviolette, Mario Marchand, Pascal Germain, and Nicolas Usunier. PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 769–776. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/779efbd24d5a7e37ce8dc93e7c04d572-Abstract.html.

Alexandre Lacasse, François Laviolette, Mario Marchand, and Francis Turgeon-Boutin. Learning with randomized majority votes. In José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part II*, volume 6322 of *Lecture Notes in Computer Science*, pages 162–177. Springer, 2010. doi: 10.1007/978-3-642-15883-4_11. URL https://doi.org/10.1007/978-3-642-15883-4_11.

John Langford. Quantitatively tight sample complexity bounds. In *Carnegie Mellon Thesis*, page 130. 2002.

John Langford. Tutorial on practical prediction theory for classification. *J. Mach. Learn. Res.*, 6:273–306, 2005. URL http://jmlr.org/papers/v6/langford05a.html.

John Langford and Rich Caruana. (Not) Bounding the True Error. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 809–816. MIT Press, 2002. URL http://papers.nips.cc/paper/1968-not-bounding-the-true-error.pdf.

John Langford and Matthias Seeger. Bounds for averaging classifiers. 2001. URL http://www.cs.cmu.edu/~jcl/papers/averaging/averaging_tech.pdf.

John Langford and John Shawe-Taylor. PAC-Bayes & margins. In *Advances in Neural Information Processing Systems*, pages 439–446, 2003.

François Laviolette, Emilie Morvant, Liva Ralaivola, and Jean-Francis Roy. Risk upper bounds for general ensemble methods with an application to multiclass classification. *Neurocomputing*, 2017.

Jean-Samuel Leboeuf, Frédéric Leblanc, and Mario Marchand. Improving generalization bounds for VC classes using the hypergeometric tail inversion. *CoRR*, abs/2111.00062, 2021. URL https://arxiv.org/abs/2111.00062.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Gaël Letarte, Pascal Germain, Benjamin Guedj, and Francois Laviolette. Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 6872–6882. Curran Associates, Inc., 2019.

Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. 1986.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(1-3):503–528, 1989. doi: 10.1007/BF01589116.

Stephan Sloth Lorenzen, Christian Igel, and Yevgeny Seldin. On PAC-Bayesian bounds for random forests. *Mach. Learn.*, 108(8-9):1503–1522, 2019. doi: 10.1007/s10994-019-05803-4.

Gábor Lugosi. Pattern classification and learning theory. In *Principles of nonparametric learning*, pages 1–56. Springer, 2002.

Enno Mammen and Alexandre B Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

Olivier Marchal and Julyan Arbel. On the sub-Gaussianity of the Beta and Dirichlet distributions. *Electronic Communications in Probability*, 22:1–14, 2017.

Andrés R. Masegosa, Stephan Sloth Lorenzen, Christian Igel, and Yevgeny Seldin. Second order PAC-Bayesian bounds for the weighted majority vote. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/386854131f58a556343e056f03626e00-Abstract.html.

Andreas Maurer. A note on the PAC-Bayesian theorem. *CoRR*, cs.LG/0411099, 2004.

David A McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational Learning Theory*, pages 230–234. ACM, 1998.

David A McAllester. PAC-Bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational Learning Theory*, pages 164–170. ACM, 1999.

David A. McAllester. Simplified PAC-Bayesian margin bounds. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, volume 2777 of *Lecture Notes in Computer Science*, pages 203–215. Springer, 2003. doi: 10.1007/978-3-540-45167-9_16.

Zakaria Mhammedi, Peter Grünwald, and Benjamin Guedj. PAC-Bayes Un-Expected Bernstein Inequality. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, [NeurIPS 2019], 8-14 December 2019, Vancouver, BC, Canada*, pages 12180–12191, 2019. URL http://papers.nips.cc/paper/9387-pac-bayes-un-expected-bernstein-inequality.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *arXiv preprint arXiv:1906.10652*, 2019.

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in*

Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 11611–11622, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/05e97c207235d63ceb1db43c60db7bbb-Abstract.html.

Balas K Natarajan. On learning sets and functions. *Machine Learning*, 4:67–97, 1989.

Radford M. Neal. *Priors for Infinite Networks*, pages 29–53. Springer New York, New York, NY, 1996. ISBN 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0_2.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 1376–1401. JMLR.org, 2015. URL http://proceedings.mlr.press/v40/Neyshabur15.html.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5947–5956, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/10ce03a1ed01077e3e289f3e53c72813-Abstract.html.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Skz_WfbCZ.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BygfghAcYX.

Didrik Nielsen. Tree boosting with XGBoost: why does XGBoost win "every" machine learning competition? Master's thesis, NTNU, 2016.

A. B. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, New York, NY, USA, 1962. Polytechnic Institute of Brooklyn.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. PAC-Bayes bounds with data dependent priors. *J. Mach. Learn. Res.*, 13:3507–3531, 2012. URL http://dl.acm.org/citation.cfm?id=2503353.

María Pérez-Ortiz, Omar Rivasplata, Benjamin Guedj, Matthew Gleeson, Jingyu Zhang, John Shawe-Taylor, Miroslaw Bober, and Josef Kittler. Learning PAC-Bayes priors for probabilistic neural networks. 2021a.

María Pérez-Ortiz, Omar Rivasplata, Emilio Parrado-Hernández, Benjamin Guedj, and John Shawe-Taylor. Progress in self-certified neural networks. *CoRR*, abs/2111.07737, 2021b.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvari. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227):1–40, 2021c. URL http://jmlr.org/papers/v22/20-879.html.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hkuq2EkPf.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, 11(2):416–431, 1983.

Omar Rivasplata, Csaba Szepesvári, John Shawe-Taylor, Emilio Parrado-Hernández, and Shiliang Sun. PAC-Bayes bounds for stable algorithms with instance-dependent priors. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9234–9244, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/386854131f58a556343e056f03626e00-Abstract.html.

William H Rogers and Terry J Wagner. A finite sample distribution-free performance bound for local discrimination rules. *The Annals of Statistics*, pages 506–514, 1978.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Jean-Francis Roy, François Laviolette, and Mario Marchand. From PAC-Bayes bounds to quadratic programs for majority votes. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011,*

Bellevue, Washington, USA, June 28 - July 2, 2011, pages 649–656. Omnipress, 2011. URL https://icml.cc/2011/papers/379_icmlpaper.pdf.

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985.

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 1232–1240. JMLR.org, 2016. URL http://proceedings.mlr.press/v51/russo16.html.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5:197–227, 1990.

Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5): 1651–1686, October 1998. doi: 10.1214/aos/1024691352.

Matthias Seeger, John Langford, and Nimrod Megiddo. An improved predictive accuracy bound for averaging classifiers. In *Proceedings of the 18th International Conference on Machine Learning*, number CONF, pages 290–297, 2001.

Matthias W. Seeger. Pac-bayesian generalisation error bounds for gaussian process classification. *J. Mach. Learn. Res.*, 3:233–269, 2002. URL http://jmlr.org/papers/v3/seeger02a.html.

Yevgeny Seldin and Naftali Tishby. Pac-bayesian analysis of co-clustering and beyond. *J. Mach. Learn. Res.*, 11:3595–3646, 2010. doi: 10.5555/1756006.1953046. URL https://dl.acm.org/doi/10.5555/1756006.1953046.

Yevgeny Seldin, François Laviolette, Nicolò Cesa-Bianchi, John Shawe-Taylor, and Peter Auer. PAC-Bayesian inequalities for martingales. In Nando de Freitas and Kevin P. Murphy, editors, *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, page 12. AUAI Press, 2012. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2341&proceeding_id=28.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge university press, 2014.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *J. Mach. Learn. Res.*, 11:2635–2670, 2010. doi: 10.5555/1756006.1953019. URL https://dl.acm.org/doi/10.5555/1756006.1953019.

J. Shawe-Taylor and R. C. Williamson. A PAC analysis of a Bayes estimator. In *Proceedings of the 10th annual conference on Computational Learning Theory*, pages 2–9. ACM, 1997.

John Shawe-Taylor and David R. Hardoon. PAC-Bayes analysis of maximum entropy classification. In *AISTATS*, 2009.

Noam Shazeer. GLU variants improve transformer. *CoRR*, abs/2002.05202, 2020. URL https://arxiv.org/abs/2002.05202.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Inf. Fusion*, 81:84–90, 2022. doi: 10.1016/j.inffus.2021.11.011.

Thomas Steinke and Lydia Zakynthinou. Reasoning about generalization via conditional mutual information. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 3437–3452. PMLR, 2020. URL http://proceedings.mlr.press/v125/steinke20a.html.

Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A strongly quasiconvex PAC-Bayesian bound. In Steve Hanneke and Lev Reyzin, editors, *International Conference on Algorithmic Learning Theory, ALT 2017, 15-17 October 2017, Kyoto University, Kyoto, Japan*, volume 76 of *Proceedings of Machine Learning Research*, pages 466–492. PMLR, 2017. URL http://proceedings.mlr.press/v76/thiemann17a.html.

Andrey Nikolayevich Tikhonov et al. On the stability of inverse problems. In *Dokl. akad. nauk sssr*, volume 39, pages 195–198, 1943.

Laura Tinsi and Arnak S. Dalalyan. Risk bounds for aggregated shallow neural networks using gaussian priors. 2021.

Ilya O. Tolstikhin and Yevgeny Seldin. PAC-Bayes-empirical-Bernstein inequality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 109–117, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/a97da629b098b75c294dffdc3e463904-Abstract.html.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, 12(4):389–434, 2012. doi: 10.1007/s10208-011-9099-z.

Thomas Uriot, Dario Izzo, Luís F Simões, Rasit Abay, Nils Einecke, Sven Rebhan, Jose Martinez-Heras, Francesca Letizia, Jan Siminski, and Klaus Merz. Spacecraft collision

avoidance challenge: design and results of a machine learning competition. *Astrodynamics*, pages 1–20, 2021.

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

Tim van Erven. PAC-Bayes Mini-tutorial: A Continuous Union Bound. May 2014.

V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. doi: 10.1137/1116025.

Vladimir Vapnik. Principles of risk minimization for learning theory. In John E. Moody, Stephen Jose Hanson, and Richard Lippmann, editors, *Advances in Neural Information Processing Systems 4, [NIPS Conference, Denver, Colorado, USA, December 2-5, 1991]*, pages 831–838. Morgan Kaufmann, 1991. URL http://papers.nips.cc/paper/506-principles-of-risk-minimization-for-learning-theory.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

Vladimir Vapnik and A Ya Chervonenkis. The method of ordered risk minimization, i. *Avtomatika i Telemekhanika*, 8:21–30, 1974.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Paul Viallard, Pascal Germain, Amaury Habrard, and Emilie Morvant. Self-bounding majority vote learning algorithms by the direct minimization of a tight PAC-Bayesian C-bound. In *ECML-PKDD 2021*, pages 167–183, 2021.

Liwei Wang, Masashi Sugiyama, Cheng Yang, Zhi-Hua Zhou, and Jufu Feng. On the margin explanation of boosting algorithms. In Rocco A. Servedio and Tong Zhang, editors, *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 479–490. Omnipress, 2008. URL http://colt2008.cs.helsinki.fi/papers/08-Wang.pdf.

Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJNpifWAb.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Yi-Shan Wu and Yevgeny Seldin. Split-kl and PAC-Bayes-split-kl inequalities. In *NeurIPS*, volume abs/2206.00706, 2022. doi: 10.48550/arXiv.2206.00706.

Yi-Shan Wu, Andrés R. Masegosa, Stephan Sloth Lorenzen, Christian Igel, and Yevgeny Seldin. Chebyshev-Cantelli PAC-Bayes-Bennett inequality for the weighted majority vote. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12625–12636, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2524–2533, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/ad71c82b22f4f65b9398f76d8be4c615-Abstract.html.

Valentina Zantedeschi, Paul Viallard, Emilie Morvant, Rémi Emonet, Amaury Habrard, Pascal Germain, and Benjamin Guedj. Learning stochastic majority votes by minimizing a PAC-Bayes generalization bound. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS*, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3): 107–115, 2021. doi: 10.1145/3446776.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. Non-vacuous generalization bounds at the ImageNet scale: a PAC-Bayesian compression approach. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BJgqqsAct7.