Progress in Self-Certified Neural Networks

María Pérez-Ortiz
AI Centre, University College London, UK
maria.perez@ucl.ac.uk

Omar Rivasplata
AI Centre, University College London, UK
o.rivasplata@ucl.ac.uk

Emilio Parrado-Hernández
Dpt. of Signal Processing and Communications
Universidad Carlos III de Madrid, Spain
eparrado@ing.uc3m.es

Benjamin Guedj
AI Centre, University College London, UK
Inria, Lille Nord-Europe Research Centre, France
b.guedj@ucl.ac.uk

John Shawe-Taylor
AI Centre, University College London, UK
j.shawe-taylor@ucl.ac.uk

Abstract

A learning method is *self-certified* if it uses all available data to simultaneously learn a predictor and certify its quality with a statistical certificate that is valid on unseen data. Recent work has shown that neural network models trained by optimising PAC-Bayes bounds lead not only to accurate predictors, but also to tight risk certificates, bearing promise towards achieving self-certified learning. In this context, learning and certification strategies based on PAC-Bayes bounds are especially attractive due to their ability to leverage all data to learn a posterior and simultaneously certify its risk. In this paper, we assess the progress towards self-certification in probabilistic neural networks learnt by PAC-Bayes inspired objectives. We empirically compare (on 4 classification datasets) classical test set bounds for deterministic predictors and a PAC-Bayes bound for randomised self-certified predictors. We first show that both of these generalisation bounds are not too far from out-of-sample test set errors. We then show that in data starvation regimes, holding out data for the test set bounds adversely affects generalisation performance, while self-certified strategies based on PAC-Bayes bounds do not suffer from this drawback, proving that they might be a suitable choice for the small data regime. We also find that probabilistic neural networks learnt by PAC-Bayes inspired objectives lead to certificates that can be surprisingly competitive with commonly used test set bounds.

1 Introduction

A crucial question arising in machine learning is how to *certify* the generalisation ability of a predictor. In statistical learning theory, this generalisation ability is assessed by its risk, also known as out-of-sample error, which is a measure of how accurately it performs on random data from the same distribution that generated the training data. Commonly, the quality of a predictor is certified on a finite sample, namely a held-out test set, making the estimation sensitive to sampling error. Generalisation bounds, however, provide a statistically sound certificate of how the model may perform on unseen data. Intuitively speaking, if the upper bound is small, then this ensures that the quantity being upper-bounded by it—i.e. the error/loss at population level—must also be small. Among the tightest

NeurIPS 2021 Workshop - Bayesian Deep Learning.

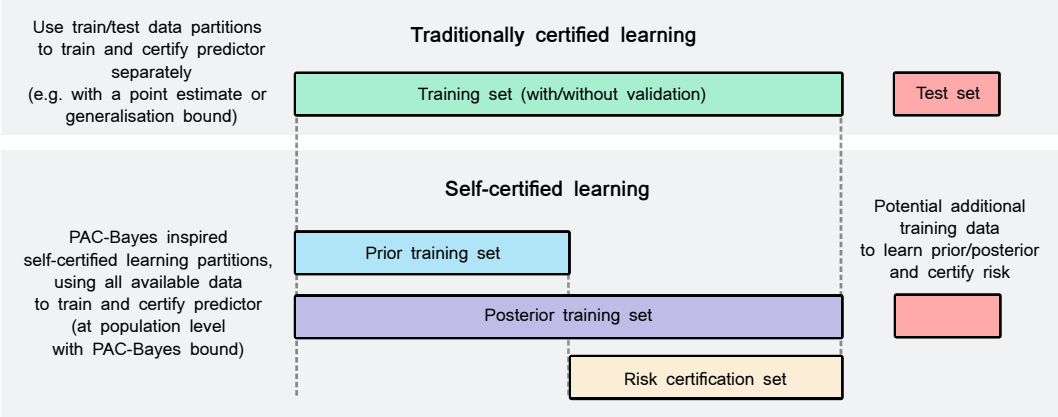

Figure 1: Comparison of data partitions in traditional machine learning vs. self-certified learning through PAC-Bayes inspired learning and certification strategies [9].

bounds are test set bounds [1] and PAC-Bayes bounds [2–6]. Test set bounds are evaluated on a test set, while PAC-Bayes bounds may be evaluated on training data exclusively, meaning they could potentially avoid the need to hold out data and instead use all data for learning. However, PAC-Bayes bounds were still, until very recently, notoriously loose when compared to test set errors. In the last few years, strategies that optimise PAC-Bayes bounds to learn a Probabilistic Neural Network (PNN), which is realised as a probability distribution over the neural network weight space [7–9], have shown promise by delivering tight risk certificates that are competitive compared to test set error rates obtained by standard empirical risk minimisation (ERM) [9]. PNNs themselves come with many advantages, such as a principled approach for uncertainty quantification [7]. However, most importantly, PNNs coupled with PAC-Bayes bounds could bring us closer to the concept of self-certified learning [9, 10], where one uses all the available data for (i) learning a predictor and (ii) certifying the predictor's performance at population level. The certification strategy would then not require a held-out test set, which may allow a more efficient use of the available data (in contrast to test set bounds and most traditional certification strategies used in the machine learning community, which require held-out data). This could radically change not only how we estimate generalisation ability but also how we approach model selection in machine learning. Our methods also provide statistically sound risk certificates that might be useful for machine learning algorithm governance.

To claim self-certified learning has been achieved we first require self-certification strategies that deliver tight risk certificates [9], so that certificates are informative of the out-of-sample error. The tightness would mean that the computed certificates closely match these out-of-sample errors, which are estimated by the error rates evaluated on a test set. In this paper, we evaluate the progress towards self-certification by comparing PAC-Bayes inspired PNNs to standard neural networks learnt by ERM and certified with test set bounds and test set errors. Fig. 1 shows the data partitions used at the different stages of learning and certification for (i) traditional strategies and (ii) our proposed strategy leading to self-certified predictors.

2 Elements of Statistical Learning

Supervised classification algorithms receive training data $S = ((X_1, Y_1), \ldots, (X_n, Y_n))$ consisting of pairs that encode inputs $X_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and their labels $Y_i \in \mathcal{Y}$. Classifiers $h_w : \mathcal{X} \to \mathcal{Y}$ are mappings from input space $\mathcal{X}$ to label space $\mathcal{Y}$, and we assume they are parametrised by 'weight vectors' $w \in \mathcal{W} \subseteq \mathbb{R}^p$. The quality of h_w is given by its risk $L(w)$, which by definition is the expected classification error on a randomly chosen pair (X, Y). However, $L(w)$ is an inaccessible measure of quality, since the distribution that generates the data is unknown. An accessible measure of quality is given by the empirical risk functional $\hat{L}_S(w) = n^{-1} \sum_{i=1}^{n} \ell(h_w(X_i), Y_i)$, defined in terms of a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to [0, \infty)$ which may be the zero-one loss or a surrogate loss function. Indeed, the empirical risk minimisation (ERM) paradigm aims to find $w \in \mathcal{W}$ that minimises this

empirical functional, typically for some choice of surrogate loss that is amenable to gradient-based optimisation, such as the squared loss or the cross-entropy loss for classification.

The outcome of training a PNN is a distribution Q over weight space and this distribution depends on the sample S. Then, given a fresh input X, the randomised classifier predicts its label by drawing a weight vector W at random from Q and applying the predictor h_W to X. For the sake of simplicity, we identify the randomised predictor with the distribution Q that defines it. The quality of this randomised predictor is measured by the expected loss notions under the random draws of weights. Thus, the loss of Q is given by $L(Q) = \int_{\mathcal{W}} L(w)Q(dw)$; and the empirical loss of Q is given by $\hat{L}_S(Q) = \int_{\mathcal{W}} \hat{L}_S(w)Q(dw)$.

The PAC-Bayes-quadratic bound [9] says that for any $\delta \in (0, 1)$, with probability of at least $1 - \delta$ over size-n i.i.d. random samples S, simultaneously for all distributions Q over $\mathcal{W}$ we have:

$$L(Q) \leq \left(\sqrt{\hat{L}_S(Q) + \frac{\mathrm{KL}(Q\|Q^0) + \log(\frac{2\sqrt{n}}{\delta})}{2n}} + \sqrt{\frac{\mathrm{KL}(Q\|Q^0) + \log(\frac{2\sqrt{n}}{\delta})}{2n}} \right)^2 .$$

In this case the prior Q^0 must be chosen without any dependence on the data S on which the empirical term $\hat{L}_S$ is evaluated. In this work, we use a partitioning scheme for the training data $S = S_{\mathrm{pri}} \cup S_{\mathrm{cert}}$ such that the prior is trained on S_{pri}, the posterior is trained on the whole set S and the risk certificate is evaluated on S_{cert}. See Fig. 1. The 'prior' and 'posterior' distributions that appear in PAC-Bayes bounds should not be confused with their Bayesian counterparts. In PAC-Bayes bounds, what is called 'prior' is a reference distribution, and what is called 'posterior' is an unrestricted distribution, in the sense that there is no likelihood factor connecting them (we refer the reader to [11, 12]).

3 Learning and Certification Strategy

In a nutshell, the learning and certification strategy used (we refer to [9, 13]) has three components: (1) choose/learn a prior; (2) learn a posterior; and (3) evaluate the risk certificate for the posterior.

3.1 Data-dependent PAC-Bayes priors

We experiment with Gaussian PAC-Bayes priors Q^0 with a diagonal covariance matrix centered at (i) random weights (uninformed data-free priors) and (ii) learnt weights (data-dependent priors) based on a subset of the dataset which is does not overlap with the subset used to compute the risk certificate (see Fig. 1). In all cases, the posterior is initialised to the prior. Similar approaches have been considered before in the PAC-Bayesian literature (we refer to [14, 9, 15]). To learn the prior mean we use ERM with dropout. The prior scale is set as a hyperparameter as done in [9].

3.2 Posterior Optimisation & Certification

We now present the essential idea of training PNNs by minimising a PAC-Bayes upper bound on the risk. We use a recently proposed PAC-Bayes inspired training objective [9], derived from Eq. (1) in the context of neural network classifiers:

$$f_{\mathrm{quad}}(Q) = \left(\sqrt{\hat{L}_S^{\text{x-e}}(Q) + \frac{\mathrm{KL}(Q\|Q^0) + \log(\frac{2\sqrt{n}}{\delta})}{2n}} + \sqrt{\frac{\mathrm{KL}(Q\|Q^0) + \log(\frac{2\sqrt{n}}{\delta})}{2n}} \right)^2 .$$

This objective is implemented using the cross-entropy loss, which is the standard surrogate loss commonly used in neural network classification. Since the PAC-Bayes bounds of Eq. (1) require the loss within [0,1], we construct a 'bounded cross-entropy' loss by lower-bounding the network probabilities by a value $p_{\min} > 0$ (cf. [16, 9]) and re-scaling the resulting bounded loss to [0,1]. The empirical risk term $\hat{L}_S^{\text{x-e}}(w)$ is then calculated with this bounded version of the cross-entropy loss.

Optimisation of the objective in Eq. (1) entails minimising over Q. By choosing Q in a parametric family of distributions, we can use the pathwise gradient estimator (see e.g. [17]) as done by [7]. The details of the reparameterisation strategy are outlined in [9]. Following [7], the reparameterisation we use is $W = \mu + \sigma \odot V$ with Gaussian distributions for each coordinate of V. The optimisation uses $\sigma = \log(1 + \exp(\rho))$, thus gradient updates are with respect to μ and ρ.

3.3 Evaluation of the Risk Certificates

After optimising the posterior distribution over network weights through the previously presented training objective, we compute a risk certificate on the error of the stochastic predictor. To do so, we follow the procedure outlined in [9], which was used before by [8] and goes back to the work of [18]. This certification procedure uses the PAC-Bayes-kl bound. In particular, the procedure is based on numerical inversion of the binary KL divergence, as done by [8, 9].

4 Experiments

Our work aims at empirically investigating two questions: *(i) Could PAC-Bayes-inspired self-certified learning algorithms, making use of all available data, provide tighter risk certificates than test set bounds? (ii) How far are these bounds from out-of-sample test set errors?* To answer these questions, we compare: (a) PNNs learnt by optimising the PAC-Bayes-quadratic bound following [9] (both in a self-certified and traditional fashion) and (b) standard neural networks learnt by empirical risk minimisation (traditionally certified, *i.e.* using a held-out test set). For the former we compute the PAC-Bayes-kl bound from [18], while the latter is evaluated with a test set bound (we used the Chernoff and binomial test set bounds, see [1] and the recent [19]).

4.1 Experimental setup

In all experiments the models are compared under the same experimental conditions, *i.e.* architecture, weight initialisation and optimiser (SGD with momentum), as well as data partitions and confidence for the bounds. The mean parameters μ_0 of the prior are initialised randomly from a truncated centered Gaussian distribution with standard deviation set to $1/\sqrt{n_{\text{in}}}$, where n_{in} is the dimension of the inputs to a particular layer, truncating at ± 2 standard deviations. All risk certificates are computed using the the PAC-Bayes-kl inequality, as explained in Section 6 of [9], with $\delta = 0.025$ and $\delta' = 0.01$ and $m = 150000$ Monte Carlo model samples. We also report the average 0-1 error of the stochastic predictor, where we randomly sample fresh model weights for each test example 100 times and compute the average 01 error. Input data was standardised for all datasets. Test set bounds are evaluated with $\delta = 0.035$ (to match the total confidence level $0.025 + 0.01$ used for the PAC-Bayes-kl bound).

We experiment with fully connected neural networks (FCN) with 3 layers (excluding the 'input layer') and 100 units per hidden layer. ReLU activations are used in each hidden layer. For learning the prior we ran the training for 500 epochs. Posterior training was run for 100 epochs. We use a training batch size of 250. ERM was run for 600 epochs. In all experiments we reserve 1% of the training data (or prior training data in the case of PAC-Bayes inspired learning) to validate the prior, as done in [13].

For all experiments we use the same hyper-parameters, which were found to work well in previous work and architectures for these datasets [13]. The prior distribution scale hyper-parameter (i.e. standard deviation σ_0) is set to 0.005. For SGD with momentum the learning rate is set to 1e^{-3} and momentum to 0.95. The same values are used for learning the prior. The dropout rate used for learning the prior was 0.01 and applied to all layers. For PAC-Bayes inspired learning, we test multiple splits of data for learning the prior and certifying the posterior from 0.5 to 0.8 and choose the one that provides the best risk certificates, as it has been shown that the optimal percentage may be dataset dependent [13].

Dataset	n	$\#f$	$\#c$
Spambase	4601	58	2
Bioresponse	3751	1777	2
Har	10299	562	6
Mammography	11183	7	2

Table 1: Datasets used: n is the total number of data points, $\#f$ the number of features and $\#c$ the number of classes.

We experiment with the four datasets described in Table 1, which are publicly available (`OpenML.org`) and were selected so as to represent a wide range of characteristics (dataset size, data dimensionality, and number of classes). Moreover, we create datasets of different sizes by removing data at random.

Our experiments span $[0.00, 0.25, 0.50, 0.75, 0.90, 0.95, 0.97, 0.98]\%$ of data removed at random for each dataset. For all datasets we selected 10% of the data as test set (stratified with class label).

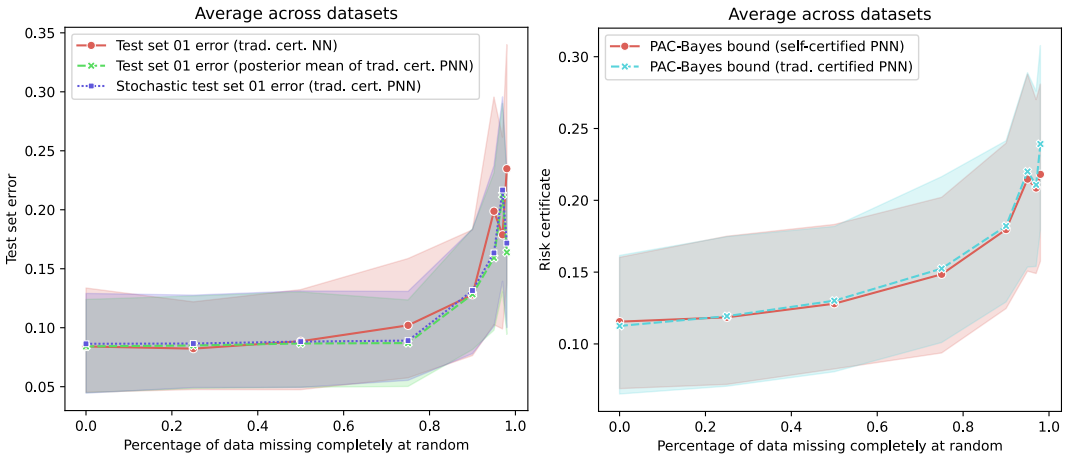

Figure 2: Left panel: we compare test set errors for standard neural networks learnt by ERM and PNNs learnt by PAC-Bayes inspired objectives. Right panel: we compare values of the PAC-Bayes-kl bound for the self-certified PNN and the traditionally certified PNN. The latter uses a held out test set.

4.2 Results

We experiment with the four mentioned datasets (see Table 1) and study the case of data starvation (*i.e.* we remove data at random from each dataset). In all cases, we reserve 10% of the available data as test (except for self-certified PNNs, which use all the available data for training). Note that our results and conclusions may be susceptible to the percentage of data that is reserved for testing purposes, as this will impact the gains achieved by self-certified learning, as well as the predictors in traditionally certified learning and the test set bounds. In our experience, similar results were obtained when holding 20% of data as test. Note that when we remove data we do so from the whole dataset, which effectively impacts the size of the test set. This is to mimic more realistically what would happen in the small data regime in which both training and test sets are reduced. The comparison between PAC-Bayes and test set bounds is particularly interesting in this data starvation regime as one cannot discard data to compute a test set bound without significantly harming performance [19].

The left part of Fig. 2 shows a comparison of test set errors for deterministic neural networks and PNNs. First, we note that these models have comparable test set performance. Additionally, the stochastic test set error (when sampling networks from the posterior distribution) does not deviate significantly from the test set error achieved by the posterior mean. This has been shown in the literature before [9, 16], with the intuition that these training objectives may promote flatter minima. The right part of the plot shows a comparison of the PAC-Bayes certificate for self-certified PNNs and traditionally certified PNNs. As expected, risk certificates seem to be improved by a self-certified setting, but especially so in the small data regime. Both plots show the average across 4 datasets (which are subsets of those described in Table 1) and 5 runs per dataset.

Fig. 3 shows a comparison of self-certified learning with PAC-Bayes bounds and traditionally certified learning with test set bounds for 4 datasets and different amounts of data missing at random. The results show that the PAC-Bayes bound of the self-certified version is competitive with the test set bounds for the traditionally certified setting, specially with the commonly used Chernoff bound. In the small data regime (specifically when removing at least 75% of training data), PAC-Bayes inspired self-certified learning shows a clear advantage, demonstrating significantly tighter bounds than both of the test set bounds considered for traditionally certified learning. See for example the case of Spambase and Bioresponse (the smallest datasets), where test set bounds on the zero-one error achieve a risk certificate between 0.7 and 0.8, while PAC-Bayes bounds stay below 0.2 and 0.5 respectively. We hypothesize that we can not see such a difference for Har and Mammography because these datasets are initially larger, so removing 98% of the data at random would still give a

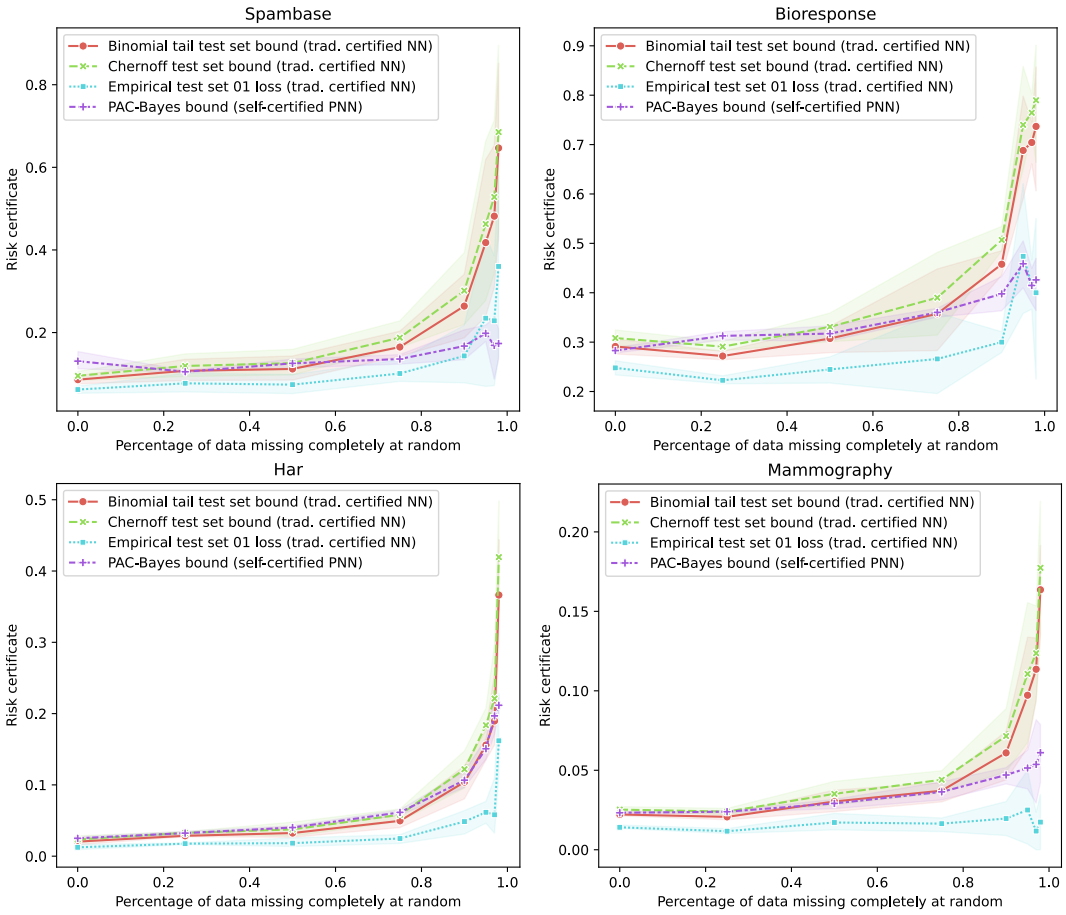

Figure 3: Comparison of results obtained with (i) self-certified PNNs with PAC-Bayes bounds and (ii) traditionally certified standard neural networks with test set bounds. Note that in both cases the compared strategies make use the same amount of data, however the self-certified learner uses all available data for training, instead of holding out part of the data for testing, as done in the traditional certification cases. The plot shows mean and confidence intervals computed over 5 runs.

dataset of around 206 and 224 points respectively, whereas for Spambase and Bioresponse we would have a total dataset size of 92 and 75 data points.

5 Discussion

This work is a preliminary empirical analysis of the progress towards self-certified neural networks, where we experiment with predictors trained on all the available data and certified with PAC-Bayes generalisation bounds. Our results show that self-certified PNNs trained optimising PAC-Bayes inspired objectives reach competitive risk certificates compared to commonly used test set bounds. At the same time, the bounds are shown to be relatively close to test set errors. These conclusions are especially true for the small data regime, where PAC-Bayes bounds with self-certified learning are significantly tighter than Chernoff and binomial tail test set bounds.

We believe that as new generalisation bounds are developed and used to inspire learning algorithms, we will get closer to the ambitious but promising objective of self-certified learning, where (i) all data can be used to learn and certify a predictor (without needing to hold-out test set data for measuring generalisation ability and model selection purposes) and (ii) we have statistically sound risk certificates of predictors' performance which do not suffer from sampling bias and hence could be used for setting performance standards when governing machine learning algorithms.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge support and funding from the U.S. Army Research Laboratory and the U. S. Army Research Office, and by the U.K. Ministry of Defence and the U.K. Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/R013616/1. This work is also partially supported by the European Commission funded project "Humane AI: Toward AI Systems That Augment and Empower Humans by Understanding Us, our Society and the World Around Us" (grant 820437).

Omar Rivasplata gratefully acknowledges funding from the U.K. Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/W522636/1.

Emilio Parrado-Hernández acknowledges support from the Spanish State Research Agency (AEI) through project PID2020-115363RB-I00.

References

[1] John Langford. Tutorial on practical prediction theory for classification. Technical report, IBM Research, 2002.

[2] David A. McAllester. Some PAC-Bayesian theorems. In *Computational Learning Theory [COLT]*, pages 230–234. ACM, 1998. Also one year later in *Machine Learning* 37(3), pages 355–363.

[3] David A. McAllester. PAC-Bayesian model averaging. In *Computational Learning Theory [COLT]*, pages 164–170. ACM, 1999.

[4] 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.

[5] Olivier Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *IMS Lecture Notes-Monograph Series*. Institute of Mathematical Statistics, 2007.

[6] Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. arXiv preprint arXiv:2110.11216, 2021.

[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning [ICML]*, pages 1613–1622. PMLR, 2015.

[8] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Conference on Uncertainty in Artificial Intelligence [UAI]*. AUAI Press, 2017.

[9] María Pérez-Ortiz, Omar Risvaplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227):1–40, 2021.

[10] Yoav Freund. Self bounding learning algorithms. In *Computational Learning Theory [COLT]*, pages 247–258. ACM, 1998.

[11] Benjamin Guedj. A primer on PAC-Bayesian learning. In Emmanuel Breuillard, editor, *Congrès de la Société Mathématique de France, Collection SMF*, volume 33, 2019.

[12] Omar Rivasplata, Ilja Kuzborskij, Csaba Szepesvári, and John Shawe-Taylor. PAC-Bayes analysis beyond the usual bounds. In *Advances in Neural Information Processing Systems [NeurIPS]*, pages 16833–16845, 2020.

[13] 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. *CoRR*, abs/2109.10304, 2021.

[14] Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. PAC-Bayes bounds with data dependent priors. *Journal of Machine Learning Research*, 13(112):3507–3531, 2012.

[15] Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel M. Roy. On the role of data in PAC-Bayes bounds. In *International Conference on Artificial Intelligence and Statistics [AISTATS]*, pages 604–612. PMLR, 2021.

[16] Gintare Karolina Dziugaite and Daniel M. Roy. Data-dependent PAC-Bayes priors via differential privacy. In *Advances in Neural Information Processing Systems [NeurIPS]*, pages 8440–8450, 2018.

[17] Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. In *International Conference on Machine Learning [ICML]*, pages 2240–2249. PMLR, 2018.

[18] John Langford and Rich Caruana. (Not) bounding the true error. In *Advances in Neural Information Processing Systems [NIPS]*, pages 809–816, 2001.

[19] 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? arXiv 2106.03542, 2021. To appear in NeurIPS 2021.