Using Region Testing to Evaluate PAC Bounds

Thomas Walker

Supervised by Professor Alessio Lomuscio

Summer 2023

Abstract

The theory of Probably Approximately Correct (PAC) generalization bounds for neural networks has been around since the work of [1]. Initially, they were a theoretical construct that elucidated the details of the learning process and gave probabilistic guarantees on the performance of machine learning models on unseen data. However, their utility in practice was only first realised by [7] who managed to contextualize the bounds practically for neural networks in a non-vacuous manner. Since then there have been other successful implementations [9][10][11]. Each of these optimizes different components of the bounds to ensure their tightness. They each are evaluated using finite training sets of discrete points. Although we cannot train networks on regions of points, we can verify the network's performance in these regions. In this work, we want to understand how knowing the network's performance on a region can be used to condition PAC bounds.

1 Introduction

Generalization of neural networks refers to the ability to perform well on data that lies outside the training set. For a neural network, this means that it can learn a function from the training set that captures information which extends to the underlying data distribution. This is observed as good network performance in training translating to good network performance on unseen data. During training the network is only exposed to a finite set of samples. These samples are a proxy for the true distribution that the network is intending to learn. The network's quality is estimated using a loss function, which tries to quantify the discrepancy between the current network outputs and the intended output. The loss function attempts this quantification by observing the network's performance on the training set. Modern neural networks typically have a far greater number of parameters than samples in the training set which means that there are many possible ways that the network can be tuned to obtain a low loss on the training set. As the training set is simply a proxy it is not clear that optimizing for good performance on this set will encode the intended behaviour into the network. The network has the capacity to memorize the training set, however, we do not observe networks learning such interpolatory functions. In fact, neural networks portray a remarkable capacity to learn representations that extend reasonably well beyond the training set. There have been many efforts to try and understand this phenomenon either from a theoretical or empirical perspective.

Probably Approximately Correct (PAC) bounds are a theoretical tool that has been developed to provide quantitative guarantees on the generalization property of neural networks. With high probability, they bound the difference in network performance on training data and unseen data. There are different types of PAC bounds that appeal to different components of the learning process to develop their bounds. For example, PAC-Bayes bounds are those developed under the Bayesian machine learning framework. It was a PAC-Bayes bound that [7] was able to implement non-vacuously. Then, [8] introduced compression bounds that are derived from compression algorithms. Where they motivated the construction of algorithms designed to reduce the number of parameters needed to represent a given neural network whilst guaranteeing a certain level of performance. Using these algorithms they were able to derive bounds on the generalization capacity, however, these particular bounds were not meaningful in practice. Although, they did motivate the subsequent work of [9] that combines this paradigm with the Bayesian framework. They utilize the notion of a compression scheme to develop priors that tightened PAC-Bayes bounds sufficiently for practical implementation. Then, [11] extends this line of reasoning to further improve the tightness of the bounds. With this work, we intend to introduce a different strategy for evaluating these bounds. With the current strategy only considering single samples, our intention is to update these bounds using regions of points. The intuition is that operating with a region allows us to infer more information on the potential behaviour of the network to unseen data. However, we can only train neural networks on discrete finite sets, these updates will have to occur after training which will introduce some added constraints to when our updated bounds will hold.

2 Problem Formalization

2.1 Notation

In this work, we will only focus on the PAC generalization bounds that apply to neural networks. To do this we need a data space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ can be thought of as a feature space and $\mathcal{Y}$ as the output space and we will suppose that there is an unknown distribution $\mathcal{D}$ defined on this space. The aim of the training process is to learn a network $h : \mathcal{X} \to \mathcal{Y}$ that produces outputs in accordance with the distribution $\mathcal{D}$. This will involve employing a learning algorithm on a training set $S = \{z_i\}_{i=1}^m = \{(x_i, y_i)\}_{i=1}^m$ which we assume consists of m $\mathrm{i.i.d}$ samples from $\mathcal{D}$. Our neural network will be parameterized by a weight vector $\mathbf{w} \in \mathcal{W}$ with the corresponding network denoted $h_{\mathbf{w}} \in \mathcal{H}$. The sets $\mathcal{W}$ and $\mathcal{H}$ will be referred to as the parameter space and the hypothesis set respectively. To assess the quality of a particular network we use a loss function $f : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ which we will require to be bounded, say by a constant C. We will use the loss function to quantify the difference between a particular network and the true output. That is for $z = (x, y) \in \mathcal{Z}$ we identify the quantity $l(h_{\mathbf{w}}(x), y) =: l_z(\mathbf{w})$ as the error of this particular example. Using this we can define the risk of our network to be

$$R(\mathbf{w}) = \mathbb{E}_{z \sim \mathcal{D}}(l_z(\mathbf{w}))$$

which is dependent on the unknown distribution $\mathcal{D}$ and hence is also unknown. So instead we work with the empirical risk of the network

$$\hat{R}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} l_{z_i}$$

which is implicitly dependent on a training set and is such that $\mathbb{E}_{S \sim \mathcal{D}^m} \left(\hat{R}(\mathbf{w})\right)$. It will be useful to introduce the notation $[k] = 1, \ldots, k$ and $[[k]]_m = k, \ldots, m$ for $k, m \in \mathbb{N}$.

2.2 Problem Statement

We will formulate the problem by considering a relatively simple PAC bound.

Theorem 2.1 ([12]). *Let $|\mathcal{W}| = M < \infty$, $\delta \in (0, 1)$ and $\mathbf{w} \in \mathcal{W}$ then it follows that*

$$\mathbb{P}_{S \sim \mathcal{D}^m} \left(R(\mathbf{w}) \leq \hat{R}(\mathbf{w}) + C \sqrt{\frac{\log\left(\frac{M}{\delta}\right)}{2m}} \right) \geq 1 - \delta.$$

As is the case with most machine learning problems, we can obtain zero training error on the training set, that is $\hat{R}(\mathbf{w}) = 0$. Assuming this we see that Theorem 2.1 tells us a $1 - \delta$ confident bound for the true error of the network. More importantly, it tells us how this bound changes as we add points to the training set. The bound gets tighter to the true error of the network as we increase the number of points on which we train the network. Adding an extra data point and training the network to zero training error reduces the bound of Theorem 2.1 by

$$C \sqrt{\frac{\log\left(\frac{M}{\delta}\right)}{2}} \left(\frac{1}{\sqrt{m}} - \frac{1}{\sqrt{m+1}} \right).$$

Or we can say that the original bound holds with confidence

$$1-\delta' = 1 - \frac{M}{\left(\frac{M}{\delta}\right)^{\frac{m+1}{m}}} \geq 1-\delta.$$

Note that the bound of Theorem 2.1 holds for all $\mathbf{w} \in \mathcal{W}$, which is why the factor of M appears. It is necessary to do this because we cannot guarantee that the added data point has a zero training error under the current parameter value. It may be possible to ensure that the updated parameter value is within some subset of the parameter space, perhaps through a stochastic gradient descent argument. However, this will not be discussed in this work. We are going to update our bounds in a slightly different way. Suppose that we have trained to a network $h_{\mathbf{w}}$ for which we can guarantee a zero training error on some region $\Delta \subset \mathcal{Z}$, we will denote this assumption by $l_{\Delta}(\mathbf{w}) = 0$. Note how this is different from the previous case in two different ways. The main difference is that we are now considering a region of points rather than a single point. The more subtle point is that there is no change in parameter value, we are assuming that for the network trained on the initial m points, there exists such a region Δ. In the previous scenario, we retrained the network and so the parameter value is different. If we did not retrain the network and instead supposed that it achieved zero training error on a point, then we cannot update our bounds in the same way as knowing the performance at a (deterministic) point provides no information (when working on a continuous data space). Henceforth, we only discuss the update of bounds in the scenario where we can guarantee performance on a region of the data space. Therefore, when we update our bounds we are going to lose the property that it holds over the entire parameter space. Despite this, we want to understand how this affects bounds such as those given in Theorem 2.1. One of the major implications of our assumptions is that they provide information for the distribution $\mathcal{D}$. From the assumptions, we obtain partial information on this unknown distribution as we can explicitly calculate,

$$p_{\Delta} = \mathbb{P}_{z\sim\mathcal{D}}(z \in \Delta) = \int_{z\in\Delta} \mathcal{D}(z)dz.$$

It is important to understand that this will not affect the quantity $R(\mathbf{w})$ as this value is calculated under the assumption that we know the distribution $\mathcal{D}$.

3 Improving PAC Bounds

3.1 Improving the Tightness of Bounds

As we saw, there are two ways in which we could improve bounds. We can either make the bound smaller or ensure that the existing bound holds with greater confidence. Following the steps of the proof of Theorem 2.1 we can determine an updated bound under our assumptions.

Theorem 3.1. *For $\mathbf{w} \in \mathcal{W}$ and $\delta \in (0,1)$ we have that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) \leq \hat{R}(\mathbf{w}) + CB(m, p_{\Delta}, \delta) \middle| l_{\Delta} = 0\right) \geq 1-\delta$$

for

$$B(m, p_{\Delta}, \delta) = \sqrt{\frac{\log\left(\frac{(1-p_{\Delta})+\sqrt{(1-p_{\Delta})^2+4\delta^{\frac{1}{m}}p_{\Delta}}}{2\delta^{\frac{1}{m}}}\right)}{2}}.$$

Note that by letting $p_{\Delta} = 0$ we just get the statement of the Theorem 2.1 without the factor of M. That is because we have not included a union bound argument as we are only interested in a single parameter value. The union bound argument is required in the setting of Theorem 2.1 as the result holds for all parameter values. The reason such bounds are derived to hold across all parameter values is that no one parameter value has information that can be leveraged, whereas, in our case, that is exactly what we have. With $p_{\Delta} = 1$ we see that $B(m, p_{\Delta}, \delta) > 0$ which is not ideal as in such a scenario we would know that $R(\mathbf{w}) = 0$. The issue arises due to a step in the proof of the theorem.

3.2 Improving the Confidence of Bounds

We can also look at improving the confidence with which bounds hold by conditioning on the event that $l_\triangle(\mathbf{w}) = 0$, which is essentially equivalent to improving the tightness of a bound. It is potentially more desirable to improve the tightness of bounds as this improvement manifests more readily in practical applications. However, we proceed with improving the confidence of bounds as often the explicit results are easier to derive. Furthermore, it provides a standardised metric to quantify the improvement imposed by conditioning on the event $l_\triangle(\mathbf{w}) = 0$.

Theorem 3.2. *For $\mathbf{w} \in \mathcal{W}$ and $\delta \in (0,1)$ we have that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) \leq \hat{R}(\mathbf{w}) + C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\triangle(\mathbf{w}) = 0\right) \geq 1 - \left(\sum_{k=1}^{m}\binom{m}{k}\delta_k p_\triangle^{m-k}(1-p_\triangle)^k\right)$$

where

$$\delta_k = \frac{1}{\left(\frac{1}{\delta}\right)^{\frac{m^2}{k^2}}}.$$

We note that $\delta_k \leq \delta$ so that we do get an improvement in the confidence of the bound. Again we see that with $p_\triangle = 0$ we recover the bounds and the confidence of Theorem 2.1. However, what we notice now is that when we let $p_\triangle = 1$ we get full confidence in our bound as we expect, which we do not get with the result of Theorem 3.1.

We will now work with a PAC bound that was derived to hold for all parameters in a countable parameter space on which a prior distribution $\pi(\mathbf{w})$ is defined. The requirement that the parameter space is countable is not as restrictive as it may seem, as computers necessarily need to work with floating point numbers and so the values of the parameters will be from a countable set even if we set out the problem over an uncountable parameter space. Furthermore, the requirement of a prior is also not restrictive as networks are often randomly initialized and so one can simply take the prior to be the distribution of the initialization on the parameter space.

Theorem 3.3 ([5])**.** *Simultaneously for all $\mathbf{w} \in \mathcal{W}$ and $\delta \in (0,1)$ the following holds,*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) \leq \inf_{\lambda > \frac{1}{2}} \frac{1}{1-\frac{1}{2\lambda}}\left(\hat{R}(\mathbf{w}) + \frac{\lambda C}{m}\left(\log\left(\frac{1}{\pi(\mathbf{w})}\right) + \log\left(\frac{1}{\delta}\right)\right)\right)\right) \geq 1 - \delta.$$

However, at the moment we are only interested in bounds that hold for a single parameter value. We can re-derive this bound to hold for a single parameter value, which will allow us to drop the assumption that the parameter space is countable and the prior distribution will no longer influence the bound.

Theorem 3.4. *For $\mathbf{w} \in \mathcal{W}$ and $\delta \in (0,1)$ we have that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) \leq \inf_{\lambda > \frac{1}{2}} \frac{1}{1-\frac{1}{2\lambda}}\left(\hat{R}(\mathbf{w}) + \frac{\lambda C}{m}\left(\log\left(\frac{1}{\delta}\right)\right)\right)\right) \geq 1 - \delta.$$

In a similar way to before we can condition this probability on the event that $l_\triangle(\mathbf{w}) = 0$ to improve the confidence with which the bound holds.

Theorem 3.5. *For $\mathbf{w} \in \mathcal{W}$ and $\delta \in (0,1)$ we have that*

$$\begin{aligned}\mathbb{P}_{S\sim\mathcal{D}^m}&\left(R(\mathbf{w}) \leq \inf_{\lambda > \frac{1}{2}} \frac{1}{1-\frac{1}{2\lambda}}\left(\hat{R}(\mathbf{w}) + \frac{\lambda C}{m}\left(\log\left(\frac{1}{\delta}\right)\right)\right)\middle| l_\triangle(\mathbf{w}) = 0\right)\\ &\geq 1 - \left(\sum_{k=1}^{m}\binom{m}{k}\exp\left(-\frac{m^2\epsilon(\mathbf{w})^2}{2kR(\mathbf{w})}\right)p_\triangle^{m-k}(1-p_\triangle)^k\right),\end{aligned}$$

where

$$\epsilon(\mathbf{w}) = \sqrt{\frac{2R(\mathbf{w})\log\left(\frac{1}{\delta}\right)}{m}}.$$

This is indeed an improvement in confidence as

$$\exp\left(-\frac{m^2\epsilon(\mathbf{w})^2}{2kR(\mathbf{w})}\right) \leq \exp\left(-\frac{m\epsilon(\mathbf{w})^2}{2R(\mathbf{w})}\right) = \delta.$$

Again with $p_\Delta = 0$ and $p_\Delta = 1$ we get the same conclusions we made from our investigation of improving the confidence of Theorem 2.1.

4 Impact on PAC-Bayes Bounds

4.1 Bounding Expected Empirical Error

We now want to operate in the Bayesian machine learning paradigm and investigate PAC Bayes bounds. We will first look at making a slightly stronger, but reasonable, assumption. Suppose that there is a subset $\Omega \subset \mathcal{W}$ in which the weights correspond to networks that achieve zero training error on the region $\Delta \subset \mathcal{Z}$. This seems reasonable if one considers linear classifiers. If a dataset has a non-zero margin between the clusters of different classes then there exists a set of parameters that would achieve zero training error. These sets of parameters would correspond to the subset Ω. Furthermore, if we recall that the research on this topic is due to the fact that neural networks are over-parameterized and can therefore learn functions that overfit to training data. The overfitting arises as there are multiple representations of the data, of which some are more desirable. Hence, it is justified to presume that a subset of our parameter space is capable of achieving zero training error on a particular region of the data space. It is important that we make it clear that we are not considering parameters that achieve zero error on some region, all the parameters in Ω must achieve zero error on the same region. This added assumption will allow us to work with expected errors, as the parameters for which $l_\Delta(\mathbf{w}) = 0$ need to have a non-zero probability mass. In particular, we will work with Theorem 4.1 and see how we can improve the confidence with which it holds using our assumptions.

Theorem 4.1. *[2] For all $\rho \in \mathcal{M}(\mathcal{W})$ and $\delta \in (0,1)$ we have that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\rho) \leq \hat{R}(\rho) + \sqrt{\frac{\mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\delta}\right) + \frac{5}{2}\log(m) + 8}{2m-1}}\right) \geq 1-\delta.$$

In the following let

$$p_\Omega = \int_\Omega \rho(\mathbf{w})d\mathbf{w}$$

and $l_\Delta(\Omega) = 0$ be the event that for all $\mathbf{w} \in \Omega$ we have $l_\Delta(\mathbf{w}) = 0$.

Theorem 4.2. *For all $\rho \in \mathcal{M}(\mathcal{W})$ and $\delta \in (0,1)$ we have that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\rho) \leq \hat{R}(\rho) + \sqrt{\frac{\mathrm{KL}(\rho,\pi)\log\left(\frac{1}{\delta}\right) + \frac{5}{2}\log(m) + 8}{2m-1}}\right)$$
$$\geq 1 - \left(\sum_{k=1}^{m}\binom{m}{k}\left(\delta_k p_\Omega + \delta(1-p_\Omega)\right)p_\Delta^{m-k}(1-p_\Delta)^k\right),$$

where δ_k is such that

$$\frac{m}{k}\sqrt{\frac{\mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\delta}\right) + \frac{5}{2}\log(m) + 8}{2m-1}} = \sqrt{\frac{\mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\delta_k}\right) + \frac{5}{2}\log(m) + 8}{2m-1}}.$$

Again we observe that is indeed an improvement in confidence. Notice that

$$\sqrt{\frac{\mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\delta}\right) + \frac{5}{2}\log(m) + 8}{2m-1}}$$

is a decreasing function in δ. Therefore, as the statement of Theorem 4.2 implies that

$$\sqrt{\frac{\mathrm{KL}(\rho,\pi)+\log\left(\frac{1}{\delta}\right)+\frac{5}{2}\log(m)+8}{2m-1}} \leq \sqrt{\frac{\mathrm{KL}(\rho,\pi)+\log\left(\frac{1}{\delta_k}\right)+\frac{5}{2}\log(m)+8}{2m-1}}$$

we must have that $\delta_k \leq \delta$.

4.2 Data-Dependent Probability Measure

In machine learning, we form the posterior distribution by the application of a learning algorithm on our training sample. For the case of neural networks we often choose our network parameter by drawing a realisation from the posterior distribution defined by stochastic gradient descent (SGD). As SGD is a random process, it really does define some probabilistic distribution on the parameter space.

Definition 4.3 ([12]). *Let $\mathcal{M}(\mathcal{W})$ be a set of probability distributions defined over $\mathcal{W}$. A data-dependent probability measure is a function*

$$\tilde{\rho} : \bigcup_{m=1}^{\infty} (\mathcal{X}\times\mathcal{Y})^m \to \mathcal{M}(\mathcal{W}).$$

Theorem 4.4 ([12]). *For all $\lambda > 0$, and data-dependent probability measure $\tilde{\rho}$, we have that*

$$\mathbb{E}_{S\sim\mathcal{D}^m}(R(\tilde{\rho})) \leq \mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\rho}) + \frac{\lambda C^2}{8m} + \frac{\mathrm{KL}(\tilde{\rho},\pi)}{\lambda}\right).$$

Suppose that we find that the network $h_{\tilde{\mathbf{w}}}$ is such that $l_\Delta(\tilde{\mathbf{w}}) = 0$. Then we could simply let our data-dependent probability measure be $\tilde{\rho}(\mathbf{w}) = \mathbb{I}(\mathbf{w})_{\{\mathbf{w}=\tilde{\mathbf{w}}\}}$. So that

$$\mathrm{KL}(\tilde{\rho},\pi) = \log\left(\frac{1}{\pi(\tilde{\mathbf{w}})}\right)$$

and

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\rho})\middle| l_\Delta(\tilde{\mathbf{w}}) = 0\right) &= \sum_{k=0}^{m}\binom{m}{k}\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\middle| z_{[k]} \notin \Delta, z_{[[k]]_m} \in \Delta\right) p_\Delta^{m-k}(1-p_\Delta)^k \\
&= \sum_{k=0}^{m}\binom{m}{k}\frac{k}{m}\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\right) p_\Delta^{m-k}(1-p_\Delta)^k \\
&= (1-p_\Delta)\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\right).
\end{aligned}$$

Corollary 4.5. *For all $\lambda > 0$, with the data-dependent probability measure $\tilde{\rho}(\mathbf{w}) = \mathbb{I}(\mathbf{w})_{\{\mathbf{w}=\tilde{\mathbf{w}}\}}$, we have that*

$$\mathbb{E}_{S\sim\mathcal{D}^m}(R(\tilde{\rho})) \leq (1-p_\Delta)\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\right) + \frac{\lambda C^2}{8m} + \frac{1}{\lambda}\log\left(\frac{1}{\pi(\tilde{\mathbf{w}})}\right).$$

On the other hand, we could assume that we have optimized to some posterior distribution $\rho \in \mathcal{W}$ which we can augment using the information that $l_\Delta(\tilde{\mathbf{w}}) = 0$ by defining the data-dependent probability measure

$$\tilde{\rho} = \gamma\mathbb{I}(\mathbf{w})_{\{\mathbf{w}=\tilde{\mathbf{w}}\}} + (1-\gamma)\rho(\mathbf{w})\mathbb{I}(\mathbf{w})_{\{\mathbf{w}\neq\tilde{\mathbf{w}}\}},$$

for $\gamma \in (0,1)$.

Corollary 4.6. *For all $\lambda > 0$, with the data dependent probability measure*

$$\tilde{\rho} = \gamma\mathbb{I}(\mathbf{w})_{\{\mathbf{w}=\tilde{\mathbf{w}}\}} + (1-\gamma)\rho(\mathbf{w})\mathbb{I}(\mathbf{w})_{\{\mathbf{w}\neq\tilde{\mathbf{w}}\}},$$

for $\gamma \in (0,1)$ *we have that*

$$\begin{aligned}\mathbb{E}_{S\sim\mathcal{D}^m}\left(R(\tilde{\rho})\right) \leq & (1-\gamma)\left(\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\rho})\right) + \frac{\mathrm{KL}(\tilde{\rho},\pi)}{\lambda} + \frac{\lambda C^2}{8m}\right) \\ & + \gamma\left(\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\right) + \frac{1}{\lambda}\log\left(\frac{1}{\pi(\tilde{\mathbf{w}})}\right) + \frac{\lambda C^2}{8m}\right) - \gamma p_\Delta \mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\right) + \frac{1-\gamma}{\lambda}\log\left(1-\gamma\right).\end{aligned}$$

Reassuringly, we see that for $\gamma = 0$ we recover the original bound of Theorem 4.4 and with $\gamma = 1$ we get the bounded we deduced previously. The general expression is minimized by,

$$\gamma = 1 - \exp\left(\lambda(1-p_\Delta)\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\right) - \lambda\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\rho)\right) - \mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\pi(\tilde{\mathbf{w}})}\right) - 1\right).$$

The reason we do not minimize the bound for $\gamma = 1$ is because we have no control on the behaviour of $h_{\tilde{\mathbf{w}}}$ on $\mathcal{Z}\setminus\Delta$. It may be the case that another parameter exists that achieves zero error on Δ and performs better on $\mathcal{Z}\setminus\Delta$ than $h_{\tilde{\mathbf{w}}}$. Similarly, if p_Δ is small then optimizing for performance on this region will probably not be a good proxy for optimizing performance on the $\mathcal{Z}$ which is what the bound is aiming to do. We can also work with the probabilistic version of Theorem 4.4 which takes the form of Theorem 4.7.

Theorem 4.7 ([4])**.** *For all* $\lambda > 0$*, for all* $\rho \in \mathcal{M}(\mathcal{W})$*, and* $\delta \in (0,1)$ *it follows that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\rho) \leq \hat{R}(\rho) + \frac{\lambda C^2}{8m} + \frac{\mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\delta}\right)}{\lambda}\right) \geq 1-\delta.$$

For our next results, we resume the assumption that $l_\Delta(\Omega) = 0$ for some $\Omega \subset \mathcal{W}$.

Theorem 4.8. *For all* $\lambda > 0$*, for all* $\rho \in \mathcal{M}(\mathcal{W})$ *and* $\delta \in (0,1)$ *it follows that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\rho) \leq \hat{R}(\rho) + \frac{\log\left(B(\lambda,m,p_\Delta,p_\Omega)\right) + \mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\delta}\right)}{\lambda}\middle| l_\Delta(\Omega) = 0\right) \geq 1-\delta,$$

where

$$B(\lambda,m,p_\Delta,p_\Omega) = p_\Omega\left(p_\Delta^m + \sum_{k=1}^m \binom{m}{k}\exp\left(\frac{\lambda^2C^2}{8k}\right)p_\Delta^{m-k}(1-p_\Delta)^k\right) + (1-p_\Omega)\exp\left(\frac{\lambda^2C^2}{8m}\right).$$

Corollary 4.9. *If we let* ρ *be the point mass at* $\tilde{\mathbf{w}} \in \Omega$ *then for all* $\lambda > 0$ *and* $\delta \in (0,1)$ *we have that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) \leq \hat{R}(\tilde{\mathbf{w}}) + \frac{\log\left(p_\Delta^m + \sum_{k=1}^m\binom{m}{k}\exp\left(\frac{\lambda^2C^2}{8k}\right)p_\Delta^{m-k}(1-p_\Delta)^k\right) + \log\left(\frac{1}{\pi(\tilde{\mathbf{w}})}\right) + \log\left(\frac{1}{\delta}\right)}{\lambda}\middle| l_\Delta(\tilde{\mathbf{w}}) = 0\right) \geq 1-\delta.$$

5 Experimentation

5.1 Simple Setup

We can implement these bounds on neural networks to empirical quantities for these bounds and assess their properties. We first consider the bound of Theorem 3.1 on a simple classification problem. Suppose that $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \{0,1\}$, so that our problem is just a binary classification of a set of one-dimensional features. We will train a ReLU network with cross-entropy loss which although is not bounded we can calculate the maximum loss observed and use this as the value of C. The network has 3 hidden layers each with 8 neurons. Our training set will consist of 50 samples. The underlying distribution $\mathcal{D}$ will be defined by 500 points which we will have access to and so we know exactly the distribution $\mathcal{D}$. If this were a realistic setting, the learning process is not necessary but we have only assumed this setup for simplicity. The 500 points could be acting as a sample to estimate the true distribution $\mathcal{D}$ using Kernel density estimates, where in this case the approximation is exact. In the more general setting, one could perhaps combine these bounds with asymptotic results to be able to implement a similar method in practice whilst retaining the probabilistic guarantees. With this, we proceed to train the network for 1000 epochs with SGD set at a learning rate of 0.1 and batch size of 10. Once the network is trained we calculate p_Δ as the proportion of the 500 points that are correctly classified by the network. Using this we calculate the bound for different values of δ. In our experiments, we investigated bounds for $\delta \in \{0.01, 0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.25\}$. We repeat this procedure 500 times to get a holistic view of the performance of our bound. What we see in Figure 1 is that we get relatively tight bounds for the true error of our network.

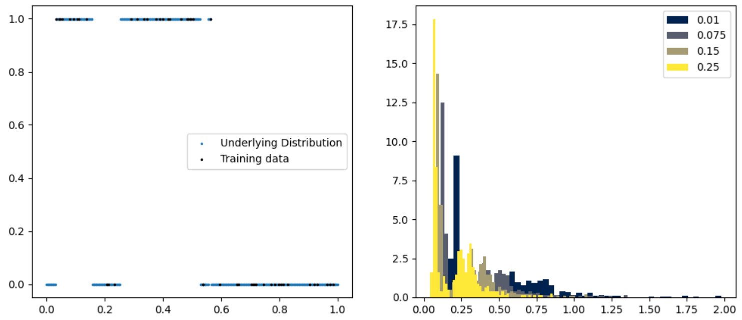

Figure 1: The results of experimentation conducted under the simple setup. **Left:** A plot to show the underlying distribution and the distribution of the training set. **Right:** A plot showing how tight the bound is for different values of δ, with negative values arising when the bound is lower than the true error.

5.2 MNIST

We conduct a similar investigation with the MNIST dataset. For ease of computation, we use 2000 points to define the underlying distribution from which we sample 500 points to train our network. Our network is a 3 layer ReLU network with 128 hidden units in each layer. We use a batch size of 50 to train the network over 50 epochs using SGD with a learning rate of 0.1. The learning in this setting is more stable and so we only repeat this procedure 50 times. We are evaluating bounds for the same values of δ as in the simple setup. From Figure 2 we see that we get bounds that are tighter than those in the simple setup. The empirical probabilities for when these bounds hold more closely follow the theoretical results. That is, sometimes we see the bounds are too optimistic and are lower than the true error.

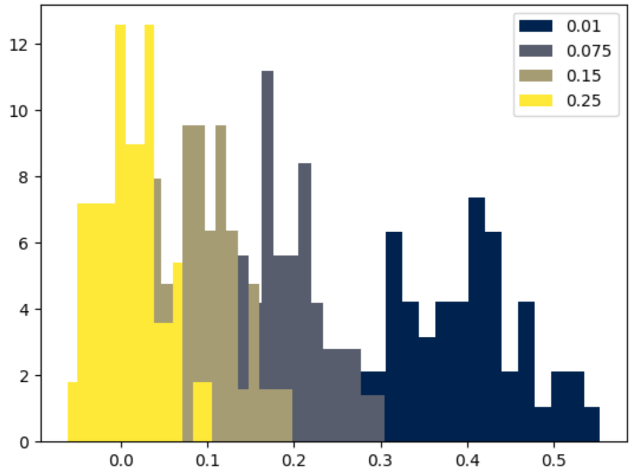

Figure 2: The results of experimentation conducted on the MNIST data. The plot shows how tight the bound is for different values of δ, with negative values arising when the bound is lower than the true error.

We can evaluate the bound of Theorem 2.1 for a specific parameter and observe how it changes when we condition on the training error being zero in a region. Figure 3 shows that the bound is tighter when conditioned and that the improvement is often better than the improvement seen by increasing the sample size by 150%. The plot of Figure 3 shows the value of the bound for varying values of p_Δ. We see that the relative improvement of the conditioned bound is more significant for larger values of p_Δ.

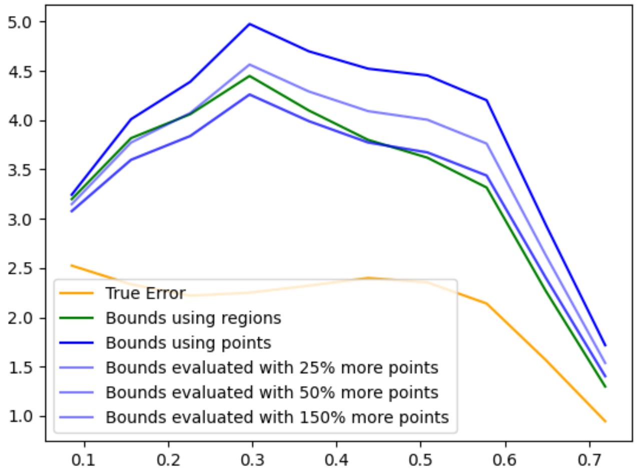

Figure 3: A plot showing how the value of the bound when conditioned on the loss on Δ being zero and when not conditioned on this information.

5.3 Bayes Bounds

In this next experiment, we implement Corollary 4.9. For this, we are required to define a prior distribution on the set of weights. We could do this by initializing the parameters of our network using $\mathrm{i.i.d}$ samples from

$\mathcal{N}\left(0, \frac{1}{n_h}\right)$, where n_h is the number of hidden units in each layer. Resulting in a prior distribution that is an isotropic Gaussian centred at the origin with probability density function

$$\pi(\mathbf{w}) = \frac{1}{(2n_h\pi)^{\frac{d}{2}}} \exp\left(-\frac{1}{2n_h}\|\mathbf{w}\|_2^2\right).$$

Where d is the total number parameters of our network, that is $\mathbf{w} \in \mathbb{R}^d$. The posterior distribution will be the point mass distribution at the weight vector found through SGD. We also need to consider the non-trivial optimization of the parameter λ. Consider the two components of the bound

$$\frac{\log\left(p_\Delta^m + \sum_{k=1}^m \binom{m}{k} \exp\left(\frac{\lambda^2 C^2}{8k}\right) p_\Delta^{m-k}(1-p_\Delta)^k\right)}{\lambda} \text{ and } \frac{\log\left(\frac{1}{\pi(\tilde{\mathbf{w}})}\right) + \log\left(\frac{1}{\delta}\right)}{\lambda}.$$

As $\lambda \to 0$ the first of these to 0 whilst the other diverges to ∞. On the other hand, as $\lambda \to \infty$ the first term diverges to ∞ whilst the second term converges to 0. In our experiments, we evaluate the bound at different values of δ (the same as considered before) and 50 values of λ spaced evenly on the interval $[2, 15]$. Due to our naive proposal for the prior we get loose bounds. Furthermore, the $\pi(\tilde{\mathbf{w}})$ is often extremely small due to the high dimensionality of the problem. To get the results of Figure 4 we restrict the network to contain a single hidden layer of 8 hidden units. When investigating the influence of δ we fix $\lambda = 5$ and when we investigate λ we fix $\delta = 0.05$. The choice of λ may not be optimal which is not an issue as we are only interested in the behaviour of the bound.

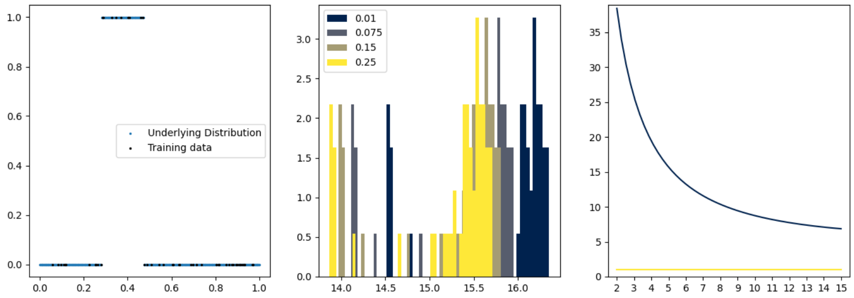

Figure 4: **Left:** The Distribution of the data and the training sample. **Centre:** The distribution of the bounds seen for different values of δ. **Right:** The average value of the bound for different values of λ.

A lot of research into the implementation of PAC-Bayes bounds involves optimizing the prior to get meaningful bounds. We will not implement such methods here for simplicity. For our case, we can simply let $\pi(\tilde{\mathbf{w}}) = 1$ so that we omit any computational restrictions. We proceed in the same way and obtain the results of Figure 5. We performed some preliminary investigations and found that $\lambda \approx 3$ is an optimal value for λ and so we set this for the value of λ when we investigate the effect of δ on our bound.

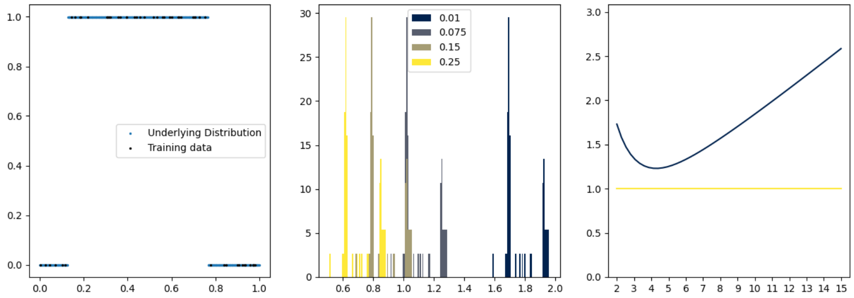

Figure 5: **Left:** The Distribution of the data and the training sample. **Centre:** The distribution of the bounds seen for different values of δ. **Right:** The average value of the bound for different values of λ.

6 Conclusion

We have seen how the knowledge that a network operates as intended on a region of the data space can tighten PAC bounds. Through experimentation, we saw that the improvement that the conditioned bounds provided were significant relative to improvements seen by increasing the sample size. The tightened bounds only hold for a particular parameter value, however, that is not an issue as that parameter is the one that we are interested in developing a bound. The updates we were conducting in this work could not have been made if the region was replaced by a singular point (unless the underlying distribution was discrete) as we require a positive probability mass on Δ for the updates to incur any improvements. Our approach differs from the traditional method of improving bounds which involves increasing the size of the training set and retraining. The traditional approach requires union bounds to be developed as the retraining changes the parameter value. Intuitively our approach is updating the bound with more information and hence should yield better results. Here we have not looked into how such regions of performance could be determined in practice. In our experiments, we adopted a fairly unrealistic approach to doing this. Future work may look at how to account for the additional step of determining Δ to yield updated bounds.

References

[1] David A. McAllester. “Some PAC-Bayesian Theorems”. In: *Machine Learning* 37 (1998), pp. 355–363.

[2] David A. McAllester. “PAC-Bayesian model averaging”. In: *Annual Conference Computational Learning Theory*. 1999.

[3] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[4] Olivier Catoni. “A PAC-Bayesian approach to adaptive classification”. In: (Jan. 2009).

[5] David A. McAllester. “A PAC-Bayesian Tutorial with A Dropout Bound”. In: *CoRR* (2013).

[6] Clayton Scott. *Hoeffding’s Inequality*. 2014.

[7] Gintare Karolina Dziugaite and Daniel M. Roy. “Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data”. In: *CoRR* (2017).

[8] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang. “Stronger generalization bounds for deep nets via a compression approach”. In: *CoRR* (2018).

[9] 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*. 2019.

[10] Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, and Daniel M. Roy. “On the role of data in PAC-Bayes bounds”. In: *CoRR* (2020).

[11] Sanae Lotfi, Marc Finzi, Sanyam Kapoor, Andres Potapczynski, Micah Goldblum, and Andrew Gordon Wilson. *PAC-Bayes Compression Bounds So Tight That They Can Explain Generalization*. 2022.

[12] Pierre Alquier. *User-friendly introduction to PAC-Bayes bounds*. 2023.

7 Appendix

7.1 Detailing the Proofs

Lemma 7.1 ([6]). *Let $U_1, \ldots, U_n$ be independent random variables taking values in an interval $[a, b]$. Then for any $t > 0$ we have that*

$$\mathbb{E}\left(\exp\left(t\sum_{i=1}^{n}(U_i - \mathbb{E}(U_i))\right)\right) \leq \exp\left(\frac{nt^2(b-a)^2}{8}\right).$$

Proof. For $s > 0$ the function $x \mapsto e^{sx}$ is convex so that

$$e^{sx} \leq \frac{x-a}{b-a}e^{sb} + \frac{b-x}{b-a}e^{sa}.$$

Let $V_i = U_i - \mathbb{E}(U_i)$, then as $\mathbb{E}(V_i) = 0$ it follows that

$$\mathbb{E}\left(\exp(sV_i)\right) \leq \frac{b}{b-a}e^{sa} - \frac{a}{b-a}e^{sb}.$$

With $p = \frac{b}{b-a}$ and $u = (b-a)s$ consider

$$\psi(u) = \log\left(pe^{sa} + (1-p)e^{sb}\right) = (p-1)u + \log\left(p + (1-p)e^{u}\right).$$

This is a smooth function so that by Taylor's theorem we have that for any $u \in \mathbb{R}$ there exists $\xi = \xi(u) \in \mathbb{R}$ such that

$$\psi(u) = \psi(0) + \psi'(0)u + \frac{1}{2}\psi''(\xi)u^2.$$

As

$$\psi'(u) = (p-1) + 1 - \frac{p}{p + (1-p)e^u}$$

we have that $\psi(0) = 0$ and $\psi'(0) = 0$. Furthermore, as

$$\psi''(u) = \frac{p(1-p)e^u}{(p+(1-p)e^u)^2}, \text{ and } \psi^{(3)}(u) = \frac{p(1-p)e^u(p+(1-p)e^u)(p-(1-p)e^u)}{(p+(1-p)e^u)^2}$$

we see that $\psi''(u)$ has a stationary point at $u^* = \log\left(\frac{p}{p-1}\right)$. For u slightly less than u^* we have $\psi^{(3)}(u) > 0$ and for u slightly larger than u^* we have $\psi^{(3)}(u) < 0$. Therefore, u^* is a maximum point and so

$$\psi''(u) \leq \psi''(u^*) = \frac{1}{4}.$$

Hence, $\psi(u) \leq \frac{u^2}{8}$ which implies that

$$\log\left(\mathbb{E}\left(\exp(sV_i)\right)\right) \leq \frac{u^2}{8} = \frac{s^2(b-a)^2}{8}.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left(\exp\left(t\sum_{i=1}^{n}(U_i - \mathbb{E}(U_i))\right)\right) &= \prod_{i=1}^{n}\mathbb{E}\left(\exp\left(t(U_i - \mathbb{E}(U_i))\right)\right)\\ &\leq \prod_{i=1}^{n}\exp\left(\frac{t^2(b-a)^2}{8}\right)\\ &\leq \exp\left(\frac{nt^2(b-a)^2}{8}\right)\end{aligned}$$

which completes the proof. □

Proof. (Theorem 3.1). Using the law of total expectation and Lemma 7.1 we observe that

$$
\begin{aligned}
&\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(t\sum_{i=1}^{m}(\mathbb{E}(l_{z_i}(\mathbf{w}))-l_{z_i}(\mathbf{w}))\right)\middle| l_\Delta=0\right)\\
&\qquad=\sum_{k=0}^{m}\binom{m}{k}\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(t\sum_{i=1}^{m}(\mathbb{E}(l_{z_i}(\mathbf{w}))-l_{z_i}(\mathbf{w}))\right)\middle| z_{[i]}\notin\Delta, z_{[[k]]}\in\Delta, l_\Delta=0\right)p_\Delta^{m-k}(1-p_\Delta)^k\\
&\qquad\leq\sum_{k=0}^{m}\binom{m}{k}\exp\left(\frac{kt^2C^2}{8}\right)p_\Delta^{m-k}(1-p_\Delta)^k.
\end{aligned}
$$

Then applying Markov's inequality we get that

$$
\begin{aligned}
\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w})>\hat{R}(\mathbf{w})+s\middle| l_\Delta=0\right)&\leq\frac{1}{\exp(mts)}\sum_{k=0}^{m}\binom{m}{k}\exp\left(\frac{kt^2C^2}{8}\right)p_\Delta^{m-k}(1-p_\Delta)^k\\
&=\frac{1}{\exp(mts)}\left(p_\Delta+(1-p_\Delta)\exp\left(\frac{t^2C^2}{8}\right)\right)^m.
\end{aligned}
$$

This holds for all $t\geq 0$, hence, we would like to find the t for which this bound is minimized. For $p_\Delta=0$ this is done by letting $t=\frac{4s}{C^2}$. However, for $p_\Delta\in(0,1)$ we cannot find explicitly the minimizer of this expression and so we will simply use $t=\frac{4s}{C^2}$ as well. If $p_\Delta=1$ we know that $R(\mathbf{w})=0$ and so the result holds trivially. Proceeding for the case where $p_\Delta\in[0,1)$ we see that

$$
\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w})>\hat{R}(\mathbf{w})+s\middle| l_\Delta=0\right)\leq\frac{1}{\exp\left(\frac{4ms^2}{C^2}\right)}\left(p_\Delta+(1-p_\Delta)\exp\left(\frac{2s^2}{C^2}\right)\right)^m.
$$

Therefore, letting

$$
\delta=\frac{1}{\exp\left(\frac{4ms^2}{C^2}\right)}\left(p_\Delta+(1-p_\Delta)\exp\left(\frac{2s^2}{C^2}\right)\right)^m
$$

we can rearrange to get that

$$
s=\sqrt{\frac{C^2\log\left(\frac{(1-p_\Delta)+\sqrt{(1-p_\Delta)^2+4\delta^{\frac{1}{m}}p_\Delta}}{2\delta^{\frac{1}{m}}}\right)}{2}}
$$

which completes the proof. □

Proof. Theorem 3.2. Here we will proceed directly with the law of total probability

$$
\begin{aligned}
&\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w})>\hat{R}(\mathbf{w})+C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w})=0\right)\\
&\quad=\sum_{k=0}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w})>\hat{R}(\mathbf{w})+C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w})=0, z_{[k]}\notin\Delta, z_{[[k+1]]_m}\in\Delta\right)p_\Delta^{m-k}(1-p_\Delta)^k.
\end{aligned}
$$

For each k we have that $l_{z_i}=0$ for $i\in[[k+1]]_m$, so we can think of each empirical error as $\frac{k}{m}$ of the empirical error of k samples. Therefore,

$$\begin{aligned}
&\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) > \hat{R}(\mathbf{w}) + C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w}) = 0\right) \\
&\qquad = \sum_{k=0}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) > \frac{k}{m}\hat{R}_k(\mathbf{w}) + C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w}) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\right) p_\Delta^{m-k}(1-p_\Delta)^k \\
&\qquad = \sum_{k=1}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^k}\left(\frac{m}{k}R(\mathbf{w}) > \hat{R}_k(\mathbf{w}) + \frac{mC}{k}\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w}) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\right) p_\Delta^{m-k}(1-p_\Delta)^k \\
&\qquad\quad + \mathbb{I}\left(R(\mathbf{w}) > C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w}) = 0, z_{[m]} \in \Delta\right) p_\Delta^m \\
&\qquad = \sum_{k=1}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^k}\left(\frac{m}{k}R(\mathbf{w}) > \hat{R}_k(\mathbf{w}) + \frac{mC}{k}\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w}) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\right) p_\Delta^{m-k}(1-p_\Delta)^k
\end{aligned}$$

where in the last inequality we have used the fact that $R(\mathbf{w}) = 0$ to deduce that

$$\mathbb{I}\left(R(\mathbf{w}) > C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w}) = 0, z_{[m]} \in \Delta\right) = 0.$$

Letting

$$\delta_k = \frac{1}{\left(\frac{1}{\delta}\right)^{\frac{m^2}{k^2}}} \leq \delta$$

for $k = 1, \ldots, m$ we get that

$$\begin{aligned}
&\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) > \hat{R}(\mathbf{w}) + C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}\middle| l_\Delta(\mathbf{w}) = 0\right) \\
&\qquad = \sum_{k=1}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^k}\left(R(\mathbf{w}) > \hat{R}_k(\mathbf{w}) + C\sqrt{\frac{\log\left(\frac{1}{\delta_i}\right)}{2k}}\middle| l_\Delta(\mathbf{w}) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\right) p_\Delta^{m-k}(1-p_\Delta)^k \\
&\qquad \leq \sum_{k=1}^{m}\binom{m}{k}\delta_i p_\Delta^{m-k}(1-p_\Delta)^k
\end{aligned}$$

which completes the proof of the theorem. □

Theorem 7.2 ([3]). *Suppose $X_1, \ldots, X_n$ are independent random variables with range $\{0, 1\}$. Let $\mu = \sum_{i=1}^{n} X_i$. Then for $\delta \in (0, 1)$ we have*

$$\mathbb{P}\left(X \leq (1-\delta)\mu\right) \leq \exp\left(-\frac{\mu\delta^2}{2}\right).$$

Proof. Theorem 3.4. We deal with the case that $C = 1$ for simplicity as we can just rescale the loss function as required. Let

$$\epsilon(\mathbf{w}) = \sqrt{\frac{2R(\mathbf{w})\log\left(\frac{1}{\delta}\right)}{m}},$$

then using Theorem 7.2 we get that

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(\hat{R}(\mathbf{w}) \le R(\mathbf{w}) - \epsilon(\mathbf{w}) - \epsilon(\mathbf{w})\right) \le \exp\left(-\frac{m\epsilon(\mathbf{w})}{2R(\mathbf{w})}\right) = \delta.$$

Therefore,

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) \le \hat{R}(\mathbf{w}) + \sqrt{\frac{2R(\mathbf{w})\log\left(\frac{1}{\delta}\right)}{m}}\right) \ge 1-\delta.$$

Using $\sqrt{ab} = \inf_{\lambda>0}\left(\frac{a}{2\lambda} + \frac{\lambda b}{2}\right)$ we get that

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\mathbf{w}) \le \hat{R}(\mathbf{w}) + \frac{R(\mathbf{w})}{2\lambda} + \frac{\lambda\left(\log\left(\frac{1}{\delta}\right)\right)}{m}\right) \ge 1-\delta$$

which upon re-arrangement completes the proof of the theorem. □

Proof. Theorem 3.5. We proceed by applying the law of total probability and the reasoning we explained in the proof of Theorem 3.2 to get that

$$\begin{aligned}
&\mathbb{P}_{S\sim\mathcal{D}^m}\left(\hat{R}(\mathbf{w}) \le R(\mathbf{w}) - \epsilon(\mathbf{w}) \middle| l_\Delta(\mathbf{w}) = 0\right)\\
&\qquad= \sum_{k=1}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^m}\left(\frac{k}{m}\hat{R}_k(\mathbf{w}) \le R(\mathbf{w}) - \epsilon(\mathbf{w}) \middle| l_\Delta(\mathbf{w}) = 0, z_{[k]} \notin \Delta, z_{[[k]]_m} \in \Delta\right) p_\Delta^{m-k}(1-p_\Delta)^k\\
&\qquad\quad+ \mathbb{I}\left(\epsilon(\mathbf{w}) \le R(\mathbf{w}) | l_\Delta(\mathbf{w}) = 0, z_{[m]} \in \Delta\right) p_\Delta^m\\
&\qquad= \sum_{k=1}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^k}\left(\hat{R}_k(\mathbf{w}) \le R(\mathbf{w}) + \frac{m-k}{k}R(\mathbf{w}) - \frac{m}{k}\epsilon(\mathbf{w}) \middle| l_\Delta(\mathbf{w}) = 0, z_{[k]} \notin \Delta, z_{[[k]]_m} \in \Delta\right) p_\Delta^{m-k}(1-p_\Delta)^k\\
&\qquad\le \sum_{k=1}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^k}\left(\hat{R}_k(\mathbf{w}) \le R(\mathbf{w}) - \frac{m}{k}\epsilon(\mathbf{w}) \middle| l_\Delta(\mathbf{w}) = 0, z_{[k]} \notin \Delta, z_{[[k]]_m} \in \Delta\right) p_\Delta^{m-k}(1-p_\Delta)^k.
\end{aligned}$$

Applying Theorem 7.2 to this we can deduce that

$$\begin{aligned}
\mathbb{P}_{S\sim\mathcal{D}^m}\left(\hat{R}(\mathbf{w}) \le R(\mathbf{w}) - \epsilon(\mathbf{w}) \middle| l_\Delta(\mathbf{w}) = 0\right) &\le \sum_{k=1}^{m}\binom{m}{k}\exp\left(-\frac{kR(\mathbf{w})\left(\frac{m\epsilon(\mathbf{w})}{kR(\mathbf{w})}\right)^2}{2}\right) p_\Delta^{m-k}(1-p_\Delta)^k\\
&= \sum_{k=1}^{m}\binom{m}{k}\exp\left(-\frac{m^2\epsilon(\mathbf{w})^2}{2kR(\mathbf{w})}\right) p_\Delta^{m-k}(1-p_\Delta)^k.
\end{aligned}$$

With this one can proceed in the same way as we do in the proof of Theorem 3.4 to complete the proof of this theorem. □

Proof. Theorem 4.2. For ease of notation let

$$B(\rho, \pi, \delta, m) = \sqrt{\frac{\mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\delta}\right) + \frac{5}{2}\log(m) + 8}{2m-1}}.$$

We first observe that for a sample S if we have $z_{[k]} \notin \Delta$ and $z_{[[k+1]]_m} \in \Delta$, for $k \neq 0$, then

$$\begin{aligned}
\hat{R}(\rho|\mathbf{w} \in \Omega) &= \mathbb{E}_{\mathbf{w}\sim\rho}\left(\hat{R}(\mathbf{w})\middle|\mathbf{w} \in \Omega\right)\\
&= \frac{k}{m}\mathbb{E}_{\mathbf{w}\sim\rho}\left(\hat{R}(\mathbf{w})\right)\\
&= \frac{k}{m}\hat{R}(\rho).
\end{aligned}$$

So for $k \neq 0$,

$$\begin{aligned}
&\mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) > \hat{R}(\rho) + B(\rho,\pi,\delta,m)\Big| l_\Delta(\Omega) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\Big)\\
&\qquad = \mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) > \hat{R}(\rho|\mathbf{w}\in\Omega) + B(\rho,\pi,\delta,m)\Big| l_\Delta(\Omega) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\Big)\, p_\Omega\\
&\qquad\quad + \mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) > \hat{R}(\rho|\mathbf{w}\notin\Omega) + B(\rho,\pi,\delta,m)\Big| l_\Delta(\Omega) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\Big)\,(1-p_\Omega)\\
&\qquad = \mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) > \frac{k}{m}\hat{R}(\rho) + B(\rho,\pi,\delta,m)\Big| l_\Delta(\Omega) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\Big)\, p_\Omega\\
&\qquad\quad + \mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) > \hat{R}(\rho) + B(\rho,\pi,\delta,m)\Big| l_\Delta(\Omega) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\Big)\,(1-p_\Omega)\\
&\qquad \leq \mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) > \hat{R}(\rho) + \frac{m}{k}B(\rho,\pi,\delta,m)\Big)\, p_\Omega + \delta(1-p_\Omega)\\
&\qquad = \delta_k p_\Omega + \delta(1-p_\Omega)
\end{aligned}$$

If $k = 0$ then the inequality clearly doesn't hold so that

$$\mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) > \hat{R}(\rho) + B(\rho,\pi,\delta,m)|l_\Delta(\Omega)) = 0, z_{[m]} \in \Delta\Big) = 0$$

Therefore,

$$\begin{aligned}
&\mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) > \hat{R}(\rho) + B(\rho,\pi,\delta,m)\Big| l_\Delta(\Omega) = 0\Big)\\
&\qquad = \sum_{k=0}^{m}\binom{m}{k}\mathbb{P}_{S\sim\mathcal{D}^m}\Big(R(\rho) \leq \hat{R}(\rho) + B(\rho,\pi,\delta,m)\Big| l_\Delta(\Omega) = 0, z_{[k]} \notin \Delta, z_{[[k+1]]_m} \in \Delta\Big)\, p_\Delta^{m-k}(1-p_\Delta)^k\\
&\qquad \leq \sum_{k=1}^{m}\binom{m}{k}\left(\delta_k p_\Omega + \delta(1-p_\Omega)\right) p_\Delta^{m-k}(1-p_\Delta)^k.
\end{aligned}$$

Taking the complement completes the proof of the theorem. □

Proof. Corollary 4.6. We first observe that

$$\begin{aligned}
\mathrm{KL}\left(\tilde{\rho},\pi\right) &= \int_{\mathcal{W}} \tilde{\rho}(\mathbf{w})\log\left(\frac{\tilde{\rho}(\mathbf{w})}{\pi(\mathbf{w})}\right)d\mathbf{w}\\
&= \int_{\mathcal{W}\setminus\{\tilde{\mathbf{w}}\}} (1-\gamma)\rho(\mathbf{w})\log\left(\frac{(1-\gamma)\rho(\mathbf{w})}{\pi(\mathbf{w})}\right)d\mathbf{w} + \gamma\log\left(\frac{\gamma}{\pi(\tilde{\mathbf{w}})}\right)\\
&= (1-\gamma)(\mathrm{KL}(\rho,\pi) + \log(1-\gamma)) + \gamma\log\left(\frac{\gamma}{\pi(\tilde{\mathbf{w}})}\right).
\end{aligned}$$

Next we see that

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\rho})\big|l_\Delta(\tilde{\mathbf{w}}) = 0\right) &= \gamma\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\rho})\Big|l_\Delta(\tilde{\mathbf{w}}) = 0, \mathbf{w} = \tilde{\mathbf{w}}\right) + (1-\gamma)\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\rho})\Big|l_\Delta(\tilde{\mathbf{w}}) = 0, \mathbf{w} \neq \tilde{\mathbf{w}}\right)\\
&= \gamma(1-p_\Delta)\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\right) + (1-\gamma)\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\rho)\right).
\end{aligned}$$

Now using Theorem 4.4 we get that

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{D}^m}\left(R(\tilde{\rho})\right) \leq& \gamma(1-p_\Delta)\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\tilde{\mathbf{w}})\right) + (1-\gamma)\mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(\rho)\right)\\
&+ \frac{1-\gamma}{\lambda}\left(\mathrm{KL}(\rho,\pi) + \log(1-\gamma)\right) + \frac{\gamma}{\lambda}\log\left(\frac{\gamma}{\pi(\tilde{\mathbf{w}})}\right) + \frac{\lambda C^2}{8m},
\end{aligned}$$

which upon re-arrangement completes the proof of the corollary. □

Lemma 7.3. *For any measurable, bounded function $f : \mathcal{W} \to \mathbb{R}$ we have,*

$$\log\left(\mathbb{E}_{\mathbf{w}\sim\pi}\left(e^{f(\mathbf{w})}\right)\right) = \sup_{\rho\in\mathcal{M}(\mathcal{W})}\left(\mathbb{E}_{\mathbf{w}\sim\rho}\left(f(\mathbf{w})\right) - \mathrm{KL}(\rho,\pi)\right).$$

Proof. Theorem 4.8. Using the law of total expectation we have that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{w}\sim\pi}\Big(\mathbb{E}_{S\sim\mathcal{D}^m}\Big(\exp\Big(tm\Big(R(\mathbf{w}) - \hat{R}(\mathbf{w})\Big)\Big)\Big)\Big|l_\Delta(\Omega) = 0\Big) \\
&\qquad = \mathbb{E}_{\mathbf{w}\sim\pi}\left(\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(tm\left(R(\mathbf{w}) - \hat{R}(\mathbf{w})\right)\right)\Big|l_\Delta(\tilde{\mathbf{w}}) = 0, \mathbf{w}\in\Omega\right)\right)p_\Omega \\
&\qquad\quad + \mathbb{E}_{\mathbf{w}\sim\pi}\left(\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(tm\left(R(\mathbf{w}) - \hat{R}(\mathbf{w})\right)\right)\Big|l_\Delta(\Omega) = 0, \mathbf{w}\in\Omega\right)\right)(1 - p_\Omega).
\end{aligned}$$

Using Lemma 7.1 we have that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{w}\sim\pi}\left(\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(tm\left(R(\mathbf{w}) - \hat{R}(\mathbf{w})\right)\right)\Big|l_\Delta(\tilde{\mathbf{w}}) = 0, \mathbf{w}\in\Omega\right)\right) \\
&\qquad \le p_\Delta^m + \sum_{k=1}^m \binom{m}{k}\exp\left(\frac{\lambda^2 C^2}{8k}\right)p_\Delta^{m-k}(1 - p_\Delta)^k
\end{aligned}$$

and

$$\mathbb{E}_{\mathbf{w}\sim\pi}\left(\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(tm\left(R(\mathbf{w}) - \hat{R}(\mathbf{w})\right)\right)\Big|l_\Delta(\Omega) = 0, \mathbf{w}\in\Omega\right)\right) \le \exp\left(\frac{\lambda^2 C^2}{8m}\right).$$

Therefore,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{w}\sim\pi}\Big(\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(\lambda\left(R(\mathbf{w}) - \hat{R}(\mathbf{w})\right)\right)\right)\Big|l_\Delta(\Omega) = 0\Big) \\
&\qquad \le p_\Omega\left(p_\Delta^m + \sum_{k=1}^m \binom{m}{k}\exp\left(\frac{\lambda^2 C^2}{8k}\right)p_\Delta^{m-k}(1 - p_\Delta)^k\right) \\
&\qquad\quad + (1 - p_\Omega)\exp\left(\frac{\lambda^2 C^2}{8m}\right).
\end{aligned}$$

Applying Fubini's theorem we can exchange the order of taking expectations to deduce that

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(\lambda\left(R(\pi) - \hat{R}(\pi)\right)\right)\Big|l_\Delta(\Omega) = 0\right) \le & p_\Omega\left(p_\Delta^m + \sum_{k=1}^m \binom{m}{k}\exp\left(\frac{\lambda^2 C^2}{8k}\right)p_\Delta^{m-k}(1 - p_\Delta)^k\right) \\
& + (1 - p_\Omega)\exp\left(\frac{\lambda^2 C^2}{8m}\right) =: B(\lambda, m, p_\Delta, p_\Omega).
\end{aligned}$$

We can now apply Lemma 7.1 to deduce that

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left(\exp\left(\sup_{\rho\in\mathcal{M}(\mathcal{W})}\left(\lambda\left(R(\rho) - \hat{R}(\rho)\right)\right) - \mathrm{KL}(\rho,\pi) - \log\left(B(\lambda, m, p_\Delta, p_\Omega)\right)\right)\Bigg|l_\Delta(\Omega) = 0\right) \le 1$$

to which we can apply Markov's inequality to get that

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(\sup_{\rho\in\mathcal{M}(\mathcal{W})}\left(\lambda\left(R(\rho) - \hat{R}(\rho)\right)\right) - \mathrm{KL}(\rho,\pi) - \log\left(B(\lambda, m, p_\Delta, p_\Omega)\right) > s\Bigg|l_\Delta(\Omega) = 0\right) \le e^{-s}$$

for fixed $s > 0$. Letting $s = \log\left(\frac{1}{\delta}\right)$ and taking the complement we get that

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(R(\rho) \le \hat{R}(\rho) + \frac{\log\left(B(\lambda, m, p_\Delta, p_\Omega)\right) + \mathrm{KL}(\rho,\pi) + \log\left(\frac{1}{\delta}\right)}{\lambda}\Bigg|l_\Delta(\Omega) = 0\right) \ge 1 - \delta,$$

which completes the proof of the theorem. □

7.2 Union Bounds

The power of our assumptions is that they tells us the form of the underlying distribution on the region $\Delta \subset \mathcal{Z}$ which allows us to calculate

$$p_\Delta = \int_\Delta \mathcal{D}(z)dz.$$

If we let $\Delta' = \mathcal{Z} \setminus \Delta$ we can define the distributions

$$\mathcal{D}_\Delta(z) = \begin{cases} \frac{\mathcal{D}(z)}{p_\Delta} & z \in \Delta \\ 0 & \text{otherwise} \end{cases} \text{ and } \mathcal{D}_{\Delta'}(z) = \begin{cases} \frac{\mathcal{D}(z)}{1-p_\Delta} & z \in \Delta' \\ 0 & \text{otherwise.} \end{cases}$$

Consequently, we can also define

$$R_\Delta(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}_\Delta}(l_z(\mathbf{w})), \text{ and } R_{\Delta'}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}_{\Delta'}}(l_z(\mathbf{w})).$$

Despite $l_\Delta(\mathbf{w}) = 0$ only holding for a potentially small region of the parameter space, we can explicitly calculate $R_\Delta(\mathbf{w})$ for any parameter value as we know the distribution $\mathcal{D}_\Delta$. Throughout we have had to derive results that only applied to a specific parameter value as we were utilizing the property that the empirical error on Δ is zero. Here we will instead see how we could potentially derive results that hold all parameter values. However, to do this we will not be able to use the fact that the training error is zero to tighten our results. Note that we can decompose the true error as

$$R(\mathbf{w}) = p_\Delta R_\Delta(\mathbf{w}) + (1-p_\Delta)R_{\Delta'}(\mathbf{w}).$$

For any $\mathbf{w} \in \mathcal{W}$ we can calculate explicitly each term on the right-hand side except for $R_{\Delta'}(\mathbf{w})$. With PAC methods we can deduce results of the form,

$$\mathbb{P}_{S\sim\mathcal{D}^m_{\Delta'}}\left(R_{\Delta'}(\mathbf{w}) \leq \hat{R}_{\Delta'}(\mathbf{w}) + B(\delta, m)\right) \geq 1-\delta,$$

that hold for all $\mathbf{w} \in \mathcal{W}$ and $\delta \in (0,1)$. Combining this with the above decomposition we can conclude that

$$\mathbb{P}_{S\sim\mathcal{D}^m_{\Delta'}}\left(R(\mathbf{w}) \leq p_\Delta R_\Delta(\mathbf{w}) + (1-p_\Delta)\left(\hat{R}_{\Delta'}(\mathbf{w}) + B(\delta, m)\right)\right) \geq 1-\delta$$

for all $\mathbf{w} \in \mathcal{W}$.