On the Interpretation of PAC-Bayes Generalisation Bounds

Andrew Foong

December 2020

Abstract

Motivated by a comment regarding the proper interpretation of PAC-Bayes bounds in Rasmussen and Williams 2006 [2], I discuss to what extent PAC-Bayes generalisation bounds can be used to infer generalisation of a machine learning algorithm after it has been trained.

1 Problem Set-up

1.1 The Frequentist View of Data Generation

Consider the following scenario. A machine learner has obtained a dataset, S, with m training examples. It may be helpful to consider a concrete example such as the MNIST dataset. Each element of the dataset $S_i \in Z$ may be viewed as an i.i.d. draw from the data-generating distribution $\mathcal{D}$. Here $Z = \mathcal{X} \times \mathcal{Y}$, the product of the input space and output space. For MNIST, Z may be thought of as $\mathbb{R}^{784} \times \{1, \ldots, 10\}$. From the point of view of the machine learner, the distribution $\mathcal{D}$ is unknown.

From a Bayesian point of view, 'unknown' is synonymous with 'random'. However, it is very difficult to specify a prior distribution on data-generating distributions $\mathcal{D}$ that adequately expresses our beliefs about the situation mathematically. Hence in this document we will be taking the **frequentist view of data-generation**: *There is a fixed (deterministic), but unknown, distribution $\mathcal{D}$ from which S has been sampled. Generalisation performance should be measured according to expectations under $\mathcal{D}$, i.e. our test data will be drawn from $\mathcal{D}$ as well.*

1.2 Generalisation Loss and the Distribution-Free Criterion

Given $\mathcal{D}$, a quantity of great interest to the machine learner is the *generalisation loss*. To define this, we set up some notation. Let $\mathcal{H}$ be the *hypothesis space*, i.e. a measurable space of functions from $\mathcal{X} \to \mathcal{Y}$. In the PAC-Bayesian setting, we consider algorithms that, based on the dataset S, return a *distribution* over predictors. Formally, define a learning algorithm as a map $Q : Z^m \to \mathcal{M}(\mathcal{H})$, where $\mathcal{M}(\mathcal{H})$ is the set of probability measures on the hypothesis space.

Next, we define the empirical and generalisation losses. Define the loss of a hypothesis $h \in \mathcal{H}$ on an example $z \in Z$ by a *loss function* $\ell : \mathcal{H} \times Z \to [0,1]$. Further, for any $P \in \mathcal{M}(\mathcal{H})$, the loss of P on $z \in Z$ is defined as

$$\ell(P, z) = \mathbb{E}_{h \sim P}[\ell(h, z)].$$

Finally, we can define the generalisation loss $L_{\mathcal{D}}(P)$ and empirical loss $L_S(P)$ as:

$$L_{\mathcal{D}}(P) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(P, z)], \tag{1}$$

$$L_S(P) = \frac{1}{m} \sum_{i=1}^{m} \ell(P, S_i). \tag{2}$$

While the empirical loss is easy to evaluate, the generalisation loss is not. In the PAC-Bayes framework, the machine learner seeks to obtain an *upper bound* on $L_{\mathcal{D}}(Q(S))$, given only access to the sample S, not the distribution $\mathcal{D}$.

However, we run into another problem. If we do not make assumptions on $\mathcal{D}$ in the form of a Bayesian prior, what kind of statements can we make? In the PAC-Bayes approach, we answer this by following the **distribution-free criterion**: *We seek bounds on the generalisation loss that hold for* any *choice of data-generating distribution $\mathcal{D}$.* This seems like a very strong condition, but it turns out, in specific situations, non-trivial statements of this form can be made.

2 Possible Desired Bounds

2.1 Bounds With a Known Dataset

We now consider the form of bounds on the generalisation error that the machine learner would *like* to have. Perhaps the most natural way of phrasing the machine learner's inquiry would be: *Given my dataset S, and the output of my algorithm $Q(S)$, can I upper bound the gap between the empirical and generalisation loss of $Q(S)$ in a distribution-free way?* An affirmative answer to this question might provide a bound of the form:

Bound 1 (Known dataset)**.** *For all datasets S, algorithms Q and data distributions $\mathcal{D}$,*

$$L_{\mathcal{D}}(Q(S)) \leq L_S(Q(S)) + \mathrm{gap}(Q(S)). \tag{3}$$

Here, 'gap$(\cdot)$' is some function that takes the learned predictor as input, and returns, in some sense, an estimate of the gap between the generalisation and empirical losses. Note that, since $\mathcal{D}$ is deterministic by the frequentist view of data-generation, and both S and Q have been fixed (think of the MNIST dataset, and a well-defined training algorithm for returning $Q(S)$), *there is no randomness on either side of eq.* (3) — the bound is either true, or false. **I am not aware of any non-vacuous bounds of the form in bound 1, and it indeed seems to me unlikely that such bounds can be found**.

2.2 Bounds With an Unknown Dataset

It turns out, however, that the machine learner can make some progress by *deliberately forgetting* some relevant information that they have access to. In the first instance, consider what happens if the machine learner forgets their knowledge of the dataset S (alternatively, you could imagine that the machine learner has yet to collect S, and is instead speculating about a *future* dataset and training run that they have not yet encountered).

In this case, since the machine learner no longer has knowledge of S, the dataset is *random.* However, by the frequentist view of data generation, there is still a fixed distribution $\mathcal{D}$ that S is an i.i.d. sample from. In this case, the machine learner's inquiry could be phrased as: *Given a dataset S is sampled from $\mathcal{D}$, (but without knowledge of the actual sample S), and my algorithm $Q : Z^m \to \mathcal{M}(\mathcal{H})$, can I upper bound the gap between the empirical and generalisation loss of $Q(S)$ in a distribution-free way?* In this case, an affirmative answer could provide a bound of the form:

Bound 2 (Unknown dataset)**.** *For all algorithms Q and data distributions $\mathcal{D}$, with probability at least $1-\delta$ over the sampling of $S \sim \mathcal{D}$,*

$$L_{\mathcal{D}}(Q(S)) \leq L_S(Q(S)) + \operatorname{gap}(Q(S)). \tag{4}$$

Notice here that we have settled for a bound that holds with high probability. This statement makes sense because the left and right hand sides of eq. (4) are now *random*, precisely because the machine learner has chosen to forget their knowledge of S. In our running example, it is as if the machine learner has lost access to the MNIST dataset, or is trying to compute a bound *before* the MNIST dataset has been collected/they become aware of its contents.

A bound in the form of bound 2 can be obtained by applying the following *PAC-Bayesian theorem*:

Theorem 1 (McAllester's PAC-Bayesian Theorem, [3])**.** *For all data distributions $\mathcal{D}$ and 'prior distributions' $\pi_0 \in \mathcal{M}(\mathcal{H})$, with probability at least $1-\delta$ over the sampling of $S \sim \mathcal{D}$,*

$$\forall \pi \in \mathcal{M}(\mathcal{H}), \quad L_{\mathcal{D}}(\pi) \leq L_S(\pi) + \sqrt{\frac{\mathrm{KL}(\pi \| \pi_0) + \log \frac{m}{\delta}}{2(m-1)}}. \tag{5}$$

We can use theorem 1 to provide a family of bounds in the form of bound 2, by taking advantage of the universal quantifier in eq. (5) to set $\pi = Q(S)$. We are even free, if we so choose, to *define* Q using the form of the bound in eq. (5), e.g.

$$Q(S) = \mathrm{argmin}_{\pi \in \mathcal{M}(\mathcal{H})} \left(L_S(\pi) + \sqrt{\frac{\mathrm{KL}(\pi \| \pi_0) + \log \frac{m}{\delta}}{2(m-1)}} \right). \tag{6}$$

This is the approach taken, for example, in [1]. Each choice of prior π_0 then defines a bound of the form in bound 2, with

$$\mathrm{gap}(Q(S)) = \sqrt{\frac{\mathrm{KL}(Q(S) \| \pi_0) + \log \frac{m}{\delta}}{2(m-1)}}. \tag{7}$$

This is the standard PAC-Bayesian framework. There are, however, some possibly unsatisfactory aspects to this approach. Firstly, is it *permissible*, from the perspective of statistical inference, for the machine learner to ignore/forget relevant information that they possess in order to obtain tractable/non-vacuous bounds? We will discuss this in a later section.

2.3 Other Bounds

First, however, we discuss another possible issue with bounds of the form of bound 2: the *choice* of what to forget (or, equivalently, which information to retain) can seem rather ad hoc. To illustrate this, we consider yet another situation, which is modelled after the one suggested in Section 7.4.1 of [2].

Let us say the machine learner has chosen to forget the dataset S, but still has knowledge of the trained predictor $Q(S)$, and also the training loss $L_S(Q(S))$. If Q is not invertible, the machine learner cannot identify S uniquely. In the case of the MNIST example, this is like the machine learner having access to the trained (random) network and its empirical loss, and knowing that the dataset S was sampled from the 'MNIST distribution', but without having access to S itself. In a sense, this situation is in between those considered in section 2.1 and section 2.2, since the machine learner has some knowledge of S through $Q(S)$, but not total knowledge as in section 2.1.

In this case, the machine learner's inquiry could be phrased as follows: *Given a dataset S is sampled from $\mathcal{D}$, (but without knowledge of the actual sample S), and given the output of my training algorithm $Q(S)$, can I upper bound the gap between the empirical and generalisation loss of $Q(S)$ in a distribution-free way?* An affirmative answer to this question could be provided by a bound of the form:

Bound 3. *For all algorithms Q, predictors $Q(S) \in \mathrm{range}(Q)$ and data distributions $\mathcal{D}$, with probability at least $1-\delta$ over the sampling of S from the* conditional *distribution $\mathcal{D}|Q(S)$*[1]*,*

$$L_{\mathcal{D}}(Q(S)) \leq L_S(Q(S)) + \mathrm{gap}(Q(S)). \tag{8}$$

[1] That is, the distribution of S conditioned on the fact that $Q(S)$ takes the particular observed value.

I am not aware of any non-vacuous bounds of the form in bound 3.[2]

In fact, we could consider yet more possibly desired bounds by 'forgetting' and 'retaining' other pieces of information. For example, the machine learner could 'forget' that they know the algorithm Q and the dataset S, while still 'retaining' knowledge of the empirical loss $L_S(Q(S)) \in \mathbb{R}$. Then an appropriate bound might look like:

Bound 4. *For all predictors $Q_1 \in \mathcal{M}(\mathcal{H})$, and data distributions $\mathcal{D}$, with probability at least $1-\delta$ over the sampling of S from the* conditional *distribution $\mathcal{D}|L_S(Q_1)^3$,*

$$L_{\mathcal{D}}(Q_1) \leq L_S(Q_1) + \mathrm{gap}(Q_1). \tag{9}$$

I am not aware of any non-vacuous bounds of the form in bound 4.[4] And so on.

2.4 Which of These Bounds is 'Correct'?

Having established that there are many possible bounds that one may wish to compute on the generalisation loss, the natural question to ask is: *which, if any of these bounds, is the correct one for the machine learner to use?*

We note that this is not simply a question of having many possible bounds, which are all compatible with each other (with some perhaps stronger and some looser), and the machine learner simply having to pick the ones that are easy to compute. This is clear from contrasting bound 1 and bound 2: if the machine learner wants to retain their knowledge of the dataset S and obtain a non-trivial distribution-free generalisation bound, the answer is simply: *it cannot be done — some assumptions* must *be made on $\mathcal{D}$ in order to make progress*[5]. However, if the machine learner forgets S, the entirely machinery of PAC-Bayesian analysis becomes open to them.

2.5 Ignoring Relevant Information

But is it legitimate to forget relevant information (i.e., S) for the sake of computational/statistical expediency? One possible defence goes like this: 'Shouldn't the machine learner be free to forget relevant information? Surely this is a *conservative* course of action. It is not as if the machine learner is *assuming* knowledge of information that they do not have access to.'

I do not find this argument entirely convincing. Consider, for example, a discrete random variable X taking values in $\{1, \ldots, 100\}$ with equal probability. Furthermore, let Y be a random variable defined by $Y = \mathbb{1}[X \in \{96, \ldots 100\}] + \epsilon$, where $\epsilon \sim \mathcal{P}$, and $\mathcal{P}$ is an unknown, fixed distribution.[6] Suppose the machine

[2] Note the right hand side of eq. (8) is stochastic, since $L_S(\cdot)$ is stochastic, but the left hand side and gap term are deterministic, since $Q(S)$ is considered known/fixed.

[3] That is, the distribution of S conditioned on the fact that $L_S(Q(S))$ takes the particular observed value.

[4] Here, again, in eq. (9), the only stochasticity is from $L_S(\cdot)$.

[5] Unless, of course, non-vacuous bounds of the form of bound 1 can be found after all.

[6] Here $\mathbb{1}[\cdot]$ is the indicator function.

learner observes $Y = 1$ and wishes to upper bound X by 95 with high probability, in a *distribution-free* (i.e. true for all $\mathcal{P}$) way. The machine learner finds this very difficult to do, since X may be strongly dependent on Y, depending on $\mathcal{P}$. For example, if $\mathcal{P}$ is a Dirac measure at $\epsilon = 0$, then we know that X is uniformly distributed in $\{96, \ldots 100\}$, and it seems we cannot upper bound X by 95 with high probability unless we make some assumptions on $\mathcal{P}$.[7]

However, the machine learner has a plan to circumvent this — they decide to 'forget' that they ever had knowledge of Y. Hence they have that X is uniformly distributed between $\{1, \ldots, 100\}$, and they can trivially upper bound X by 95 with probability 0.95. Furthermore, the machine learner has managed to do this in a distribution-free way — this upper bound holds for all $\mathcal{P}$, since $\mathcal{P}$ only affected Y and the machine learner has decided to ignore Y entirely. I think most people would agree that this is not a satisfactory way for the machine learner to perform statistical inference. Although this is an extreme example, I believe it illustrates the danger of assuming that we are simply allowed to forget information without consequences in a statistical inference problem.

2.6 A Pragmatic Approach

Another defence might go like this: 'In any statistical inference problem, there will inevitably be *some* relevant information that we cannot include for the sake of tractability. PAC-Bayes is no different. All statistical inferences are wrong, but some are useful.' I find this argument somewhat more convincing than the previous one, but still not entirely satisfactory.

In any statistical inference problem, it is true that things must be simplified somewhat in order for us to get a handle on them. However, if there is a 'big' piece of information that we know is relevant to the problem, but are ignoring, we should make an explicit note of that, and *be willing to provide an argument for why it will not affect the main conclusions of our statistical inference drastically.* **I am not aware of specific arguments to this effect in the PAC-Bayes setting**.

2.7 The Long-Term Frequencies Perspective

Another perspective can be brought on the problem by considering the long-term validity of the bounds over many samples of the dataset. This is perhaps natural as PAC-Bayesian bounds are framed within a frequentist standpoint. For example, we may consider sampling many times from 'the MNIST distribution'[8], computing our PAC-Bayes bound as in bound 2, and seeing whether they hold.[9] Then the PAC-Bayesian theorem assures us that, in the long run, the bound will hold a fraction greater than $1 - \delta$ of the time.

[7] cf. the Bayesian approach.

[8] Perhaps, asking many different sets of people to write digits.

[9] for the sake of argument, assume an oracle has given us access to the true distribution $\mathcal{D}$ so we can compute $L_{\mathcal{D}}(Q(S))$ exactly

This sounds like good news — if δ is high, we have a very reliable bound, in this sense, and we have avoided making any assumptions on $\mathcal{D}$. However, I do not think this removes the *ad hoc* nature of the 'forgetting' that the machine learner performs. For example, if we know our empirical loss is $l \in \mathbb{R}$, we could ask, in the long run: *Of the training runs performed that obtained a specific loss $L_S(Q(S)) = l$, for what fraction of these runs does my generalisation bound hold?* Answering this question essentially amounts to obtaining a bound of the form in bound 4. However, there are, as far as I know, no non-trivial PAC-Bayesian bounds of this form. Why is looking at this fraction any less 'legitimate' than looking at all of the training runs?

3 Tentative Conclusions

My tentative conclusions on the matter are as follows: while standard PAC-Bayes bounds of the form in bound 2 are valid method of performing statistical inference *a priori*[10], they *cease* to be a valid grounds of inferring (in a distribution-free way) generalisation, *once the dataset S has become known to the machine learner.*[11] If the machine learner wants to upper bound generalisation error in a distribution-free way after sampling the dataset, they need to prove a bound of the form in bound 1.[12] Furthermore, it is *not* legitimate for the machine learner to simply 'forget' their knowledge of the dataset S without providing some form of justification, as explained in section 2.5.

References

[1] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[2] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning.* 2006.

[3] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge university press, 2014.

[10] That is, before the machine learner is aware of the contents of the dataset S

[11] We note here that in PAC-Bayes, Q is often chosen by explicitly optimising the bound in a way that is dependent on S — indeed, that is one of the strengths of the PAC-Bayes framework as compared to standard PAC bounds.

[12] Proving a bound of this form may seem hopelessly difficult — but this is unsurprising, since no assumptions have been made on $\mathcal{D}$ whatsoever. A Bayesian approach to the problem would make explicit assumptions on $\mathcal{D}$ in order to make progress (as mentioned before, this is not without its own difficulties).