Statistical Test for Overfitting

James Schmidt, EP6 Uncertainty Quantification Team
Johns Hopkins University Applied Physics Laboratory

March 24, 2023

Abstract

High complexity models are notorious in machine learning for overfitting, a phenomenon in which a model well represents data but fails to generalize to an underlying data generating process. A typical procedure for heuristically circumventing overfitting is to compute empirical risk on a hold out set and halt once (or flag that/when) it begins to increase. While this method often assists in outputting a well-generalizing model, we argue for a statistically grounded test in line with concentration inequalities, namely that empirical means with high probability should be close to their true mean, and therefore close to each other. We interpret this test to quantitatively define and for use in detecting overfitting and distribution shift.

1 Introduction

Supervised machine learning is severely underdetermined: a finite labeled data set is used to search a function space for an appropriate model fitting both the data and "from where the data comes." While the full function space is often at least two infinite orders of magnitude greater than the data, practitioners usually restrict search to a hypothesis class that is parametrized as a finite dimensional space. If this hypothesis class is too restricted, the search may output a model which fails to represent or approximate the data well enough; if, on the other hand, the class is too rich, the output model may represent the data *too* well, in that the model fails to represent the underlying distribution from which data is drawn. This tradeoff between *underfitting* and *overfitting*, respectively, touches upon (among other things) hypothesis class complexity, and often one wishes to determine whether a model well generalizes. Pursuing this inquiry presupposes that the model performs well on data, knowledge of which is knowable and known in the course of training. Generalization, on the other hand, deals with performance on data the model is not trained on.

To check for generalization, standard practice often evaluates performance on a holdout set, not used during training. For by itself, satisfactory model performance on the training data does not guarantee generalization performance, simply because the model is optimized to fit the data, and may thus fit *only* the data ([3]). Therefore a two step process—1. optimization with training data and 2. verification with hold out data—mitigates the problem that performance on training data cannot reliably speak to generalization. On the other hand, this two step process treats training data and hold out data as altogether different beasts with different tasks and different intended uses. Decoupling the conclusions we draw from training data and hold out data threatens to undermine the original impetus according to which training data was used in the first place, namely to optimize some function for the express purpose of optimizing its expectation. We explain the reasons for this paradox, and propose a solution which translates into a statistical test which may be deployed for identifying overfitting. In section 2, we review requisite background for the supervised learning problem, discuss the problem with training data, how it relates to overfitting, and why we would still like to use model performance with respect to training data to asses generalization. In section 3, we detail the statistical test for achieving this end, and give commentary on

how this test elucidates the *meaning* of overfitting. We end with some plots in section 4 illustrating the use of the test in simulation.

2 Technical Background

The setting for a supervised machine learning problem starts with the following data:

1. a joint probability space $(\mathcal{X} \times \mathcal{Y}, \mathbb{P}_{\mathcal{X}\times\mathcal{Y}})$,
2. labeled data $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big) \in (\mathcal{X} \times \mathcal{Y})^m$,
3. a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, usually finite dimensional, elements $\tilde{y} \in \mathcal{H}$ of which are typically called *models*, and
4. a cost function generator $c : \mathcal{H} \to \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$ mapping a model $\tilde{y}$ to random variable $c_{\tilde{y}}$, whose output $c_{\tilde{y}}(x, y)$ on input (x, y) is a measure of fit between $\tilde{y}(x)$ and y.

The goal is to concoct an *algorithm* $\hat{y}_{(\cdot)} : (\mathcal{X} \times \mathcal{Y})^\omega \to \mathcal{H}$ which outputs a model $\hat{y}_S$ with small expected cost

$$\mathbb{E}(c_{\hat{y}_S}) \approx \inf_{\tilde{y}\in\mathcal{H}} \mathbb{E}(c_{\tilde{y}}).$$

The measure $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ generating data (x_i, y_i) is usually unknown, and data S is used to proxy approximate expectation and to optimize with respect to the expected risk function

$$\begin{array}{rccc} \mathbb{E}(c_{(\cdot)}) : & \mathcal{H} & \to & \mathbb{R} \\ & \tilde{y} & \mapsto & \mathbb{E}(c_{\tilde{y}}). \end{array} \tag{1}$$

The standard algorithm for optimizing expectation is empirical risk minimization, namely

$$\hat{y}_S \in \arg\min_{\tilde{y}\in\mathcal{H}} e_S(\tilde{y}), \tag{2}$$

where

$$e_S(\tilde{y}) := \frac{1}{|S|} \sum_{(x,y)\in S} c_{\tilde{y}}(x, y). \tag{3}$$

Law of Large Numbers intuition suggests that

$$e_S(\tilde{y}) \approx \mathbb{E}(c_{\tilde{y}}) \tag{4}$$

when $|S|$ is large, so supposing as much, an output $\hat{y}_S$ of eq. (2) ought to be a close approximation of the true goal, in the sense that

$$\mathbb{E}(c_{\hat{y}_S}) \approx \inf_{\tilde{y}\in\mathcal{H}} \mathbb{E}(c_{\tilde{y}}). \tag{5}$$

To the extent that a model $\tilde{y} \in \mathcal{H}$ (approximately) satisfies eq. (5), we say that the model *generalizes* (ε-generalizes if the error in approximation is bounded by ε), and to the extent that models in $\mathcal{H}$ can be guaranteed to generalize, we say that $\mathcal{H}$ is some kind of *learnable*. (Quick aside to pay passing lipservice: the formal notion of *probably approximately correct* (PAC) learnability, for example, extends guarantees of concentration bounds to an optimization (over $\mathcal{H}$) context, and defines $\mathcal{H}$ as PAC learnable if there is a sample complexity $\mu : (0, 1)^2 \to \mathbb{N}$ for which $\hat{y}_S$ may be guaranteed to ε-generalize with at least $1 - \delta$ probability as long as $|S| > \mu(\varepsilon, \delta)$ ([8] [4]).) Properly quantifying the character and richness of $\mathcal{H}$ (as captured, e.g., by VC dimension) demarcates learnability conditions, and various theoretical results exist providing such guarantees.

Absent formal learnability guarantees, it turns out that LLN reasoning is not sufficient for ensuring generalization. The reasons are multifarious but substantively turn around currying ([5, §2.3]) of the cost function generator $c : \mathcal{H} \to \mathbb{R}^{(\mathcal{X}\times\mathcal{Y})}$. For a *fixed* $\tilde{y} \in \mathcal{H}$, the map $c_{\tilde{y}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a random variable, and therefore defines a measure $\mathbb{P}_{\mathbb{R}}$ on $\mathbb{R}$ by $\mathbb{P}_{\mathbb{R}}([a, b]) := \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}\big(c_{\tilde{y}}^{-1}([a, b])\big)$. This means, among other things, given data $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big) \sim_{iid} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$, that $c_{\tilde{y}}(S) := \big(c_{\tilde{y}}(x_1, y_1), \ldots, c_{\tilde{y}}(x_m, y_m)\big) \sim_{iid} \mathbb{P}_{\mathbb{R}}$. And independence invites valid conclusions of various concentration results.

Searching over a function space, in the supervised machine learning setting, adds complications to otherwise innocuous independence conclusions. For the learning algorithm $\hat{y} : (\mathcal{X} \times \mathcal{Y})^\omega \to \mathcal{H}$ first fixes *data* $S \in (\mathcal{X} \times \mathcal{Y})^\omega$ in search of a certain minimum with respect to *this* data. Given different data, the algorithm outputs a different model. Formally, the curried cost generator $c_{(\cdot)} : \mathcal{H} \to \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$ defines an empirical risk generator $e(\cdot) : \mathcal{H} \to \mathbb{R}^{(\mathcal{X}\times\mathcal{Y})^\omega}$ defined by sending $\tilde{y} \mapsto e_{(\cdot)}(\tilde{y})$, the latter of which is defined by mapping data $S \in (\mathcal{X} \times \mathcal{Y})^\omega$ to $e_S(\tilde{y})$ (eq. (3)), and with respect to which LLN reasoning and the like may properly apply. The optimization procedure, however, flips the currying around: fixing data $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we have a cost on models $c_{(\cdot)}(x, y) : \mathcal{H} \to \mathbb{R}$ defined by $\tilde{y} \mapsto c_{\tilde{y}}(x, y)$, which extends to empirical risk $e_S(\cdot) : \mathcal{H} \to \mathbb{R}$ mapping model $\tilde{y} \mapsto e_{\tilde{y}}(S)$, instantiating the curried function $e_{(\cdot)} : (\mathcal{X} \times \mathcal{Y})^\omega \to \mathbb{R}^{\mathcal{H}}$.[1]

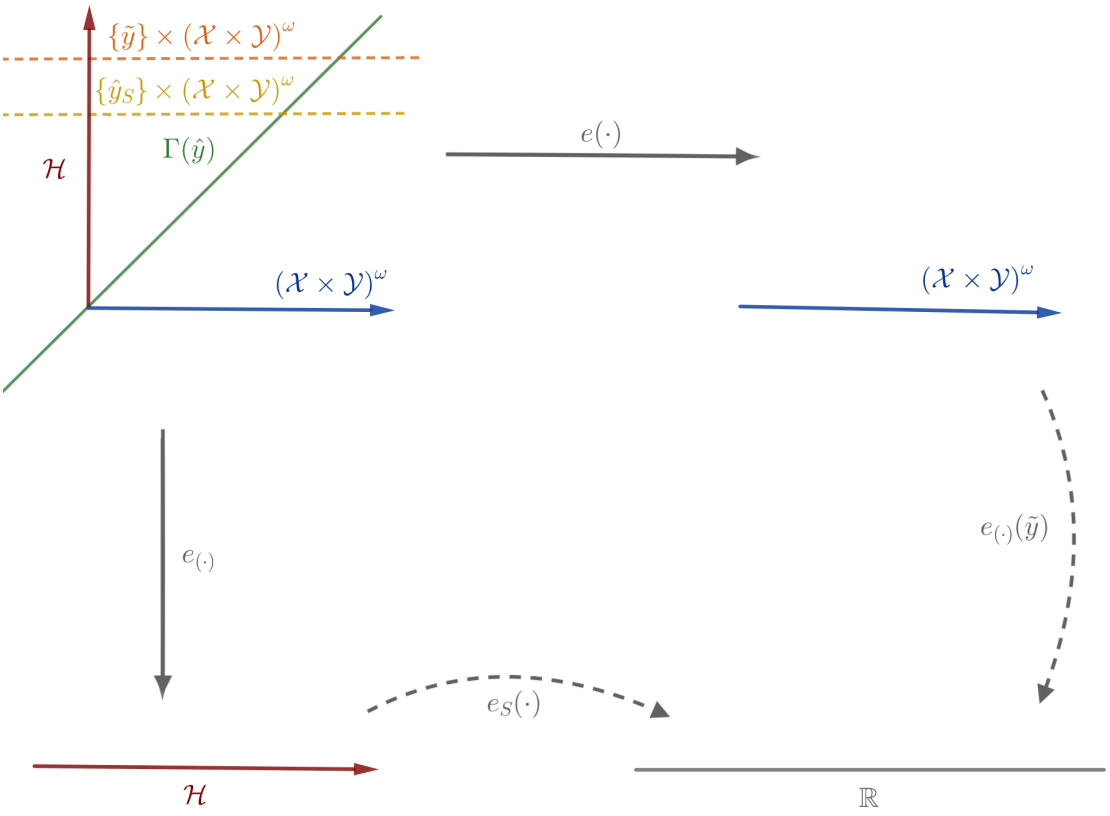

Figure 1: Slicing the Cost Function $c : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y})^\omega \to \mathbb{R}$

To distill the overfitting problem with respect to this discussion on currying, define

$$\Gamma(\hat{y}) := \{(\tilde{y}, S) \in \mathcal{H} \times (\mathcal{X} \times \mathcal{Y})^\omega : \tilde{y} = \hat{y}_S\},$$

a diagonal of sorts—explicitly, the pullback of $\mathcal{H} \times (\mathcal{X} \times \mathcal{Y})^\omega \underset{\mathrm{id}\circ\pi_1}{\overset{\hat{y}_{(\cdot)}\circ\pi_2}{\rightrightarrows}} \mathcal{H}$ —and observe that evaluation of training performance is confined to $\Gamma(\hat{y})$, whereas the focal point for performance evaluation a la statistical generalization guarantees is with respect to each slice (as indicated by orange dashes in fig. 1) $\{\tilde{y}\} \times (\mathcal{X} \times \mathcal{Y})^\omega$. One may place a measure $\mathbb{P}_\Gamma$ on $\Gamma(\hat{y})$—in fact, this pullback naturally inherits measures on $(\mathcal{X} \times \mathcal{Y})^\omega$—but statements of events on this set do not translate directly to the ones we care about on $\{\hat{y}_S\} \times (\mathcal{X} \times \mathcal{Y})^\omega$.

[1] The reversal or roles in subscripts between c and e is unfortunate, but otherwise reflective of the primary purpose of each function, namely that $c_{\tilde{y}}$ measures performance of model $\tilde{y}$ on a datapoint $(x, y) \in \mathcal{X} \times \mathcal{Y}$ while e_S measures empirical risk of fixed data on a model $\tilde{y} \in \mathcal{H}$.

While we cannot rely on empirical risk $e_S(\hat{y}_S)$ by itself to relay generalization performance, we *may* in concert with $e_{S'}(\hat{y}_S)$ for some *other* data $S' \in (\mathcal{X} \times \mathcal{Y})^\omega$, usually called a holdout or validation set. Typically performance at each training stage is evaluated on the holdout set, and early stopping conditions check that validation performance improves. An onset of validation performance degradation is often interpreted as or taken to be indication of overfitting. Illustrations of overfitting in the literature (e.g. [1], [3], [9]) display performance on training data compared with performance on hold out data, often parameterized by model complexity or training step. Notable in such graphics is a general profile *shape* with little attention paid to how close one ought to expect both curves to be.

While many investigations of overfitting consider performance across various models ([7], [6]), we introduce a statistical test, based on classic concentration inequalities, with respect to which overfitting may *quantitatively* be defined, relying on comparison of validation performance to training set performance. We argue that because model construction uses and depends on (minimization with respect to $\tilde{y}$ of) $e_S(\tilde{y})$, we should be able to conclude performance with *it*. In fact, comparison against empirical risk $e_S(\hat{y}_S)$ provides an anchor against which we may statistically draw probabilistic conclusions. While the test we provide amounts to much of the same as common stopping criteria for training, the reasoning we give is both grounded in the math and provides threads for distinguishing causes of error.

3 Detecting Overfitting

3.1 The Test

We consider the case where cost $c_{(\cdot)} : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$ is bounded as $c_{\hat{y}_S} \subset [0, 1]$, such as most classification problems or restricted classes of regression problems. In this case, Hoeffding-like bounds abound and we expect that

$$\mathbb{P}_{(\mathcal{X}\times\mathcal{Y})^k}\left(\left|\mathbb{E}(c_{\hat{y}_S}) - e_{(\cdot)}(\hat{y}_S)\right| > \varepsilon\right) < 2e^{-2\varepsilon^2 k}. \tag{6}$$

In other words, for independently sampled data $S' \in (\mathcal{X} \times \mathcal{Y})^k$, $e_{S'}(\hat{y}_S) \approx \mathbb{E}(c_{\hat{y}_S}) \pm \varepsilon$ with probability at least $1 - e^{-2\varepsilon^2/k}$.[2] While $S \in (\mathcal{X} \times \mathcal{Y})^m$ is also drawn independently, by assumption, we cannot quite conclude the same of $e_S(c_{\hat{y}_S})$ because with respect to the $c_{\hat{y}_S}$-induced measure on $\mathbb{R}$, the data sequence $(c_{\hat{y}_S}(x_1, y_1), \ldots, c_{\hat{y}_S}(x_m, y_m))$ is not. We may, however, suppose that a *consequence* of independence holds, namely that

$$|\mathbb{E}(c_{\hat{y}_S}) - e_S(\hat{y}_S)| < \varepsilon/2, \tag{7}$$

and use this (possibly counterfactual) supposition to test its truth. While possibly counterintuitive, the kind of bound in eq. (7) is exactly what we desire from a generalizing model $\hat{y}_S$.

Proposition 3.1 (Test for Overfitting)**.** Suppose that bound (7) holds. Then

$$\mathbb{P}_{(\mathcal{X}\times\mathcal{Y})^k}\left(\left|e_S(\hat{y}_S) - e_{S'}(\hat{y}_S)\right| > \varepsilon\right) \leq 2e^{-\frac{\varepsilon^2 k}{2}}. \tag{8}$$

Therefore, the null hypothesis that trained model $\hat{y}_S$ $\frac{\varepsilon}{2}$-generalizes may be tested using probability bound eq. (8).

Proof. Since

$$\begin{aligned}\left|e_S(\hat{y}_S) - e_{S'}(\hat{y}_S)\right| &= \left|e_S(\hat{y}_S) - \mathbb{E}(c_{\hat{y}_S}) + \mathbb{E}(c_{\hat{y}_S}) - e_{S'}(\hat{y}_S)\right| \\ &\leq \left|e_S(\hat{y}_S) - \mathbb{E}(c_{\hat{y}_S})\right| + \left|\mathbb{E}(c_{\hat{y}_S}) - e_{S'}(\hat{y}_S)\right| \\ &< \tfrac{\varepsilon}{2} + \left|\mathbb{E}(c_{\hat{y}_S}) - e_{S'}(\hat{y}_S)\right|\end{aligned}$$

we conclude

$$\left\{\left|e_S(\hat{y}_S) - e_{(\cdot)}(\hat{y}_S)\right| > \varepsilon\right\} \subseteq \left\{\left|\mathbb{E}(c_{\hat{y}_S}) - e_{(\cdot)}(\hat{y}_S)\right| > \frac{\varepsilon}{2}\right\}.$$

Inclusion of events implies inequality of measures, and we apply Hoeffding (inequality (9)) to bound the right hand side probability $\mathbb{P}\left(\left|\mathbb{E}(c_{\hat{y}_S}) - e_{(\cdot)}(\hat{y}_S)\right| > \frac{\varepsilon}{2}\right)$. □

[2] The fact that $\mathbb{E}(c_{(\cdot)})$ and $e(\cdot)$ both take $\hat{y}_S$ as argument is irrelevant: the bound holds for any $\tilde{y} \in \mathcal{H}$.

3.2 Interpreting the Output

Overfitting is a heuristic notion which suggests a model has fit the data and not the distribution which generated it. On closer inspection, however, the test we propose does not provide indication of *only* overfitting. In fact, the supposition of generalization is one with respect to a certain (fixed) distribution; this test thus additionally assumes that the test data $\mathsf{S}' \sim_{\mathtt{iid}} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ as well. It may not. For there may be some form of distributional shift according to which $\mathsf{S}' \sim_{\mathtt{iid}} \mathbb{P}'_{\mathcal{X}\times\mathcal{Y}}$, in which case we cannot guarantee inquality (8), at least not if the expectation $\mathbb{E}(\mathrm{c}_{\hat{\mathsf{y}}_\mathsf{S}})$ is computed with respect to the original measure $\mathrm{d}\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$. In other words, instantiation of event $|\mathrm{e}_\mathsf{S}(\hat{\mathsf{y}}_\mathsf{S}) - \mathrm{e}_{(\cdot)}(\hat{\mathsf{y}}_\mathsf{S})| > \varepsilon$ by inequality $|\mathrm{e}_\mathsf{S}(\hat{\mathsf{y}}_\mathsf{S}) - \mathrm{e}_{\mathsf{S}'}(\hat{\mathsf{y}}_\mathsf{S})| > \varepsilon$ may suggest:

1. an unlikely sample S' was received (all the hypotheses hold),
2. $\hat{\mathsf{y}}_\mathsf{S}$ does not generalize $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ with respect to $\mathrm{c}_{(\cdot)}$ (overfitting), or
3. $\mathsf{S}' \not\sim_{\mathtt{iid}} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ (possible distributional shift).

It is important when running a statistical test to respect the scope of what it purports to evaluate: namely, *if* a set of assumptions hold—in this case 1. that $\hat{\mathsf{y}}_\mathsf{S}$ generalizes (eq. (7)) and 2. $\mathsf{S}' \sim_{\mathtt{iid}} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$—then the probability that a certain kind of event occurs is bounded by some value which is explicitly calculable. Realization of the unlikely and unlucky event by S' can either mean S' really is unlucky or that one of the assumptions fails.

While this test is expressed with respect to the cost function $\mathrm{c}_{\tilde{\mathsf{y}}}$ or $\mathrm{c}_{\hat{\mathsf{y}}_{\mathsf{S}'}}$, it need not be so limited. In fact, any map $\mathrm{f} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ could be used to probe the distribution, substituting the appropriate concentration inequality depending on the range of f. When $\mathrm{f}(\mathcal{X} \times \mathcal{Y})$ is bounded, we may rely on a version of Hoeffding, which converges exponentially. Subsequent work will investigate the use of *random projections* to examine distribution shift and uncertainty quantification.

Finally, it is worth acknowledging that concentration bounds may be used for estimation of mean from *only* a (the) holdout set, still providing some confident approximation with precision. An operative assumption in supervised machine learning optimization is that fitting a model to training data thereby (if with some give) fits the model to the distribution. Use of concentration bounds on holdout set *only* indicate (absolute) model performance. In tandem with the training set we may rigorously define and test for overfitting: a model $\hat{\mathsf{y}}_\mathsf{S} : \mathcal{X} \to \mathcal{Y}$ ε-overfits data S—implicitly failing to fit distribution $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$)—when $\mathrm{e}_\mathsf{S}(\mathrm{c}_{\hat{\mathsf{y}}_\mathsf{S}}) < \mathbb{E}(\mathrm{c}_{\hat{\mathsf{y}}_\mathsf{S}}) - \varepsilon$. While the test we introduce keeps open the possibility of distribution shift, we leave the reader in anticipation of forthcoming work illustrating that judicious application of random projection provides a test to eliminate or otherwise draw out this possibility.

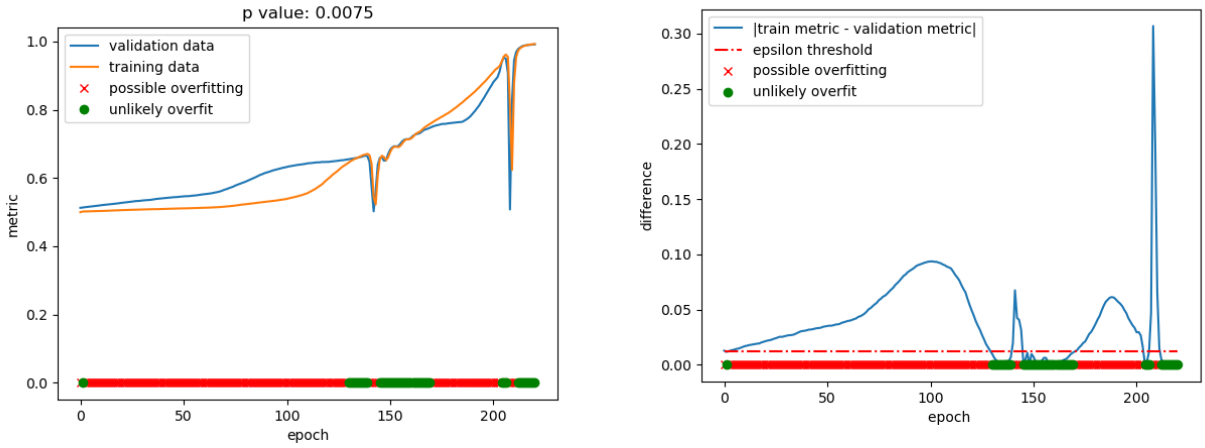

Figure 2: $\mathrm{k} = 20,000$

4 In Simulation

Code implementing this test can be found at https://github.com/schmidttgenstein/qudos.git.

In each of the following illustrations, we plot empirical performance (accuracy for a binary classification problem) with respect to both training and hold out ("validation") data[3] on the left, and the absolute difference on the right. These curves are plotted against training epoch, and each pair uses a different size for S' with respect to which either probability or precision depend (appendix A). Fixing the confidence at $p = 0.0075$ (so that we sample a set satisfying error less than precision with probability at least $1 - 0.0075$), we denote the precision with the dashed red line in the right figures. Per inequality (9), this precision may be made finer with more data.

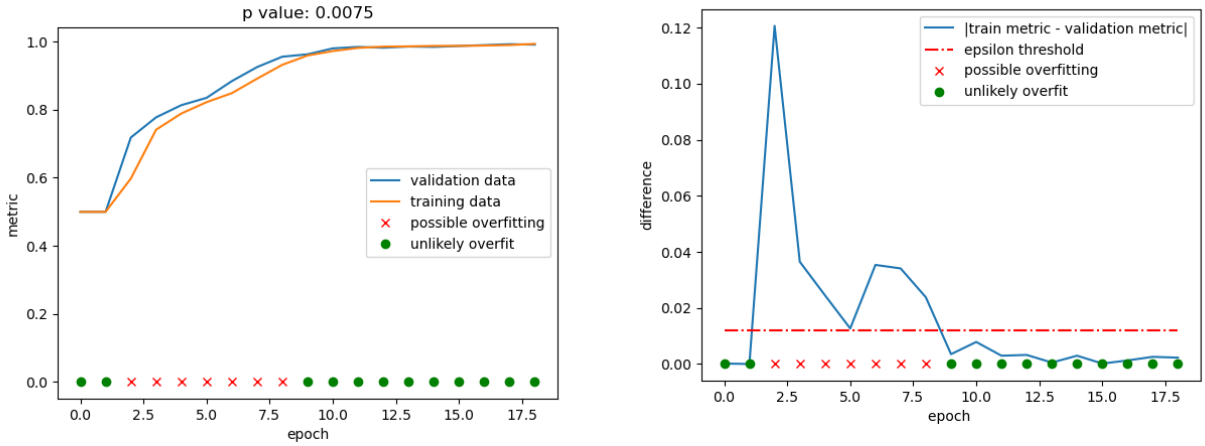

Figure 3: $k = 100,000$

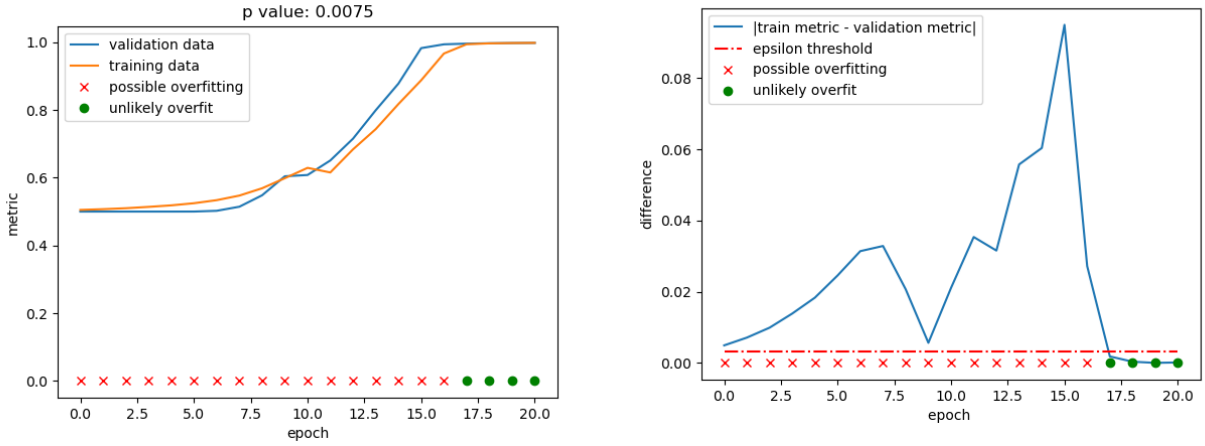

Figure 4: $k = 500,000$

[3] In these runs, validation data is not used for anything other than evaluation (e.g. hyperparameter selection, etc.), so could with impunity be called 'test' data.

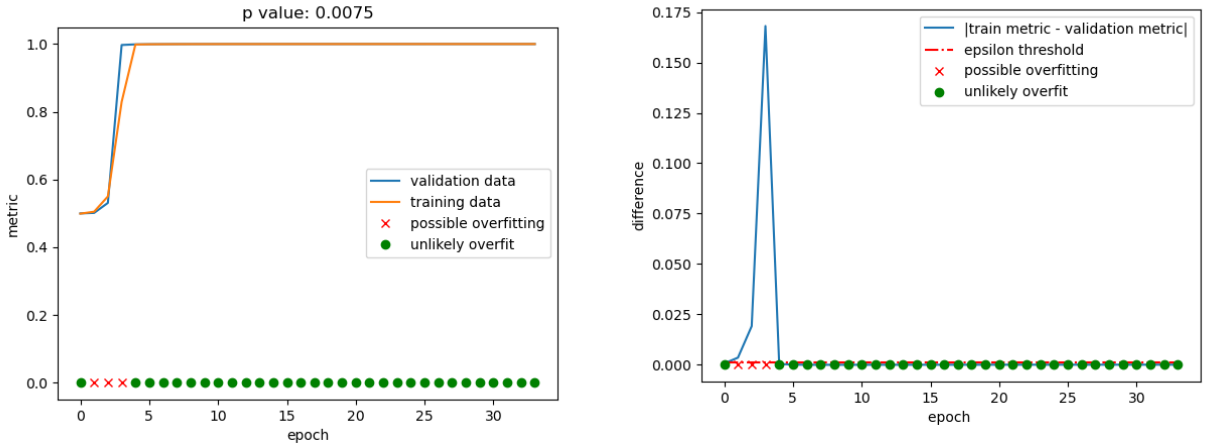

Figure 5: $k \approx 2,000,000$

A Hoeffding's Inequality for Statistical Hypothesis Testing

Hoeffding's inequality gives a probability bound for independent sample $S = (x_1, \ldots, x_m) \sim_{iid} \mathbb{P}_{\mathcal{X}}$ when $\mathcal{X} = [0, 1]$, namely:

$$\mathbb{P}_{\mathcal{X}}\left(\left|\int_{\mathcal{X}} x d\mathbb{P}_{\mathcal{X}}(x) - \frac{1}{|S|}\sum_{x \in S}\right|\right) < 2e^{-2\varepsilon^2|S|}. \tag{9}$$

Therefore, given any two of confidence specification $\delta \in (0, 1)$, data set sized $|S| = m$, and precision bound $\varepsilon \in (0, 1)$, one may readily solve for the third.

Proof of its verity and other applications may be found in various probability texts ([2], [8], [4]).

References

[1] J. Bilmes. Underfitting and overfitting in machine learning. *UW ECE Course Notes*, 2020. 4

[2] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 02 2013. 7

[3] B. Ghojogh and M. Crowley. The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial. *arXiv [stat.ML]*, 2019. 1, 4

[4] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018. 2, 7

[5] E. Riehl. *Category Theory in Context*. Dover, 2016. 2

[6] R. Roelofs. Measuring generalization and overfitting in machine learning. 2019. 4

[7] Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. A meta-analysis of overfitting in machine learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 4

[8] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning*. Cambridge University Press, 2014. 2, 7

[9] M. Valdenegro-Toro and M. Sabatelli. Machine learning students overfit to overfitting. *arXiv [cs.LG]*. 4