Testing for Overfitting

James Schmidt*
Johns Hopkins University
Johns Hopkins University Applied Physics Laboratory

May 9, 2023

Abstract

High complexity models are notorious in machine learning for overfitting, a phenomenon in which models well represent data but fail to generalize an underlying data generating process. A typical procedure for circumventing overfitting computes empirical risk on a holdout set and halts once (or flags that/when) it begins to increase. Such practice often helps in outputting a well-generalizing model, but justification for why it works is primarily heuristic.

We discuss the overfitting problem and explain why standard asymptotic and concentration results do not hold for evaluation with training data. We then proceed to introduce and argue for a hypothesis test by means of which both model performance may be evaluated using training data, and overfitting quantitatively defined and detected. We rely on said concentration bounds which guarantee that empirical means should, with high probability, approximate their true mean to conclude that they should approximate each other. We stipulate conditions under which this test is valid, describe how the test may be used for identifying overfitting, articulate a further nuance according to which distributional shift may be flagged, and highlight an alternative notion of learning which usefully captures generalization in the absence of uniform PAC guarantees.

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. Generally, this tradeoff between *underfitting* and *overfitting*, respectively, is asymmetric: a model which fits data may (and hopefully does) still generalize to the underlying distribution, while a model which underfits data usually does not fit the distribution. Stated differently, underfitting is *detectable* in the course of performance evaluation while overfitting cannot be identified by performance on the training data alone ([3]).

To mitigate the aspect blindness of training data performance to overfitting, standard practice sets aside a holdout set disjoint from training and computes performance separately. Thus, training a model ordinarily incorporates two distinct steps: 1. optimization with training data to fit (model to) data and 2. verification of generalization by evaluating performance on holdout data. While vague heuristics motivating this two-step procedure abound in the literature and

*Department of Applied Mathematics and Statistics, email: aschmi40@jhu.edu

research community, rigorous statistical rationale less ubiquitously accompany justification of its use. Moreover, this two-step process facially treats training data and holdout data as altogether different kinds of things, with different tasks and different intended uses. As such, separating the conclusions we draw from training data and holdout data threatens to undermine the original impetus according to which training data is used for training in the first place, namely *that* optimization with respect to training data should *thereby* optimize an expectation (generalization). We explain the reasons for this paradox, and propose a solution that translates into a statistical test which may be deployed for both defining and identifying overfitting, using modified Law of Large Numbers (LLN) intuition that empirical means should approximate their expectation.

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 on training data to contribute to assessing generalization. In section 3, we detail the statistical test for achieving this end, and give commentary on how this test clarifies the meaning of overfitting. We point out how the test validates generalization even absent strong but restrictive (e.g. PAC) learnability guarantees; we also introduce a weaker but still rich notion of learnability. We end with some plots in section 4 illustrating the use of the test in simulation.

2 Technical Background

2.1 Supervised Machine Learning

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

1. a joint probability space $(\mathcal{X} \times \mathcal{Y}, \mathbb{P}_{\mathcal{X}\times\mathcal{Y}})$,[1]
2. labeled data $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big) \in (\mathcal{X} \times \mathcal{Y})^\omega := \bigcup_{m\in\mathbb{N}} (\mathcal{X} \times \mathcal{Y})^m$,
3. a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ of functions $\tilde{y} : \mathcal{X} \to \mathcal{Y}$,[2] usually finite dimensional, elements $\tilde{y} \in \mathcal{H}$ of which are 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 $c_{\tilde{y}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, whose output $c_{\tilde{y}}(x, y)$ on input (x, y) is a measure of fit between prediction $\tilde{y}(x)$ and label 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

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

having some guarantees of approximation performance in probability. 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 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 this optimization is empirical risk minimization, namely

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

where empirical risk is defined as

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

[1] We leave implicit the σ-algebra of measurable sets and suppose that anything we try measuring is indeed $\mathbb{P}$-measurable.

[2] The notation $\mathcal{Y}^{\mathcal{X}}$ denotes the *set* $\{\tilde{y} : \mathcal{X} \to \mathcal{Y}\}$ of unstructured functions with domain $\mathcal{X}$ and codomain $\mathcal{Y}$. Of course, we require $\mathcal{H}$ to consist only of measurable such functions.

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) may be hoped to be a close approximation of the true goal, in the sense that

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

To the extent that a model $\tilde{y} \in \mathcal{H}$ (approximately) satisfies approximation (4), 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 optimality $\inf_{\tilde{y}} \mathbb{E}(c_{\tilde{y}})$ (5), we say that $\mathcal{H}$ is some kind of *learnable*. The familiar and formal notion of *probably approximately correct* (PAC) learnability, for example, extends guarantees of concentration bounds to an optimization (over $\mathcal{H}$) context, and defines $\mathcal{H}$ to be 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 $|S| > \mu(\varepsilon,\delta)$ ([9], [4]).[3] 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.

2.2 Overfitting and Generalization

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* ([6, §2.3]) of the cost function generator $c : \mathcal{H} \to \mathbb{R}^{(\mathcal{X}\times\mathcal{Y})}$. The notion of currying reflects pre-fixing arguments of a multivariable function to generate a function of fewer variables, and casting the learning objective in this formalism is helpful for understanding the overfitting problem. For a *fixed* model $\tilde{y} \in \mathcal{H}$, 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 takes *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. The curried cost generator, by contrast, $c_{(\cdot)} : \mathcal{H} \to \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$ 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 learning 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 $\tilde{y} \mapsto e_S(\tilde{y})$, instantiating the curried function $e_{(\cdot)} : \big(\mathcal{X}\times\mathcal{Y}\big)^\omega \to \mathbb{R}^{\mathcal{H}}$.[4]

Order of operations matter. Consider uncurried versions $\mathcal{H}\times(\mathcal{X}\times\mathcal{Y})^\omega \underset{e_{(\cdot)}(\cdot)}{\overset{c_{(\cdot)}(\cdot)}{\rightrightarrows}} \mathbb{R}$ of the cost and empirical risk functions—thereby deprioritizing either data $S \in (\mathcal{X}\times\mathcal{Y})^\omega$ or model $\tilde{y} \in \mathcal{H}$—and define

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

the pullback of diagram $\mathcal{H}\times(\mathcal{X}\times\mathcal{Y})^\omega \underset{id_{\mathcal{H}}\circ\pi_1}{\overset{\hat{y}_{(\cdot)}\circ\pi_2}{\rightrightarrows}} \mathcal{H}$ (see fig. 1). Evaluation of empirical performance for a model $\hat{y}_S$ using training data S lives in $\Gamma(\hat{y})$, which elucidates why performance of

[3] Explicitly, if $m > \mu(\varepsilon,\delta)$ then $\mathbb{P}_{(\mathcal{X}\times\mathcal{Y})^m}\Big(\big|\mathbb{E}(c_{\hat{y}_{(\cdot)}}) - \inf_{\tilde{y}\in\mathcal{H}}\mathbb{E}(c_{\tilde{y}})\big| > \varepsilon\Big) < \delta$. Strictly speaking, PAC learnability only requires the existence of an algorithm $\hat{y} : (\mathcal{X}\times\mathcal{Y})^\omega \to \mathcal{H}$ satisfying this bound, not necessarily that empirical risk minimization is it.

[4] The reversal of 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 $\tilde{y} \in \mathcal{H}$.

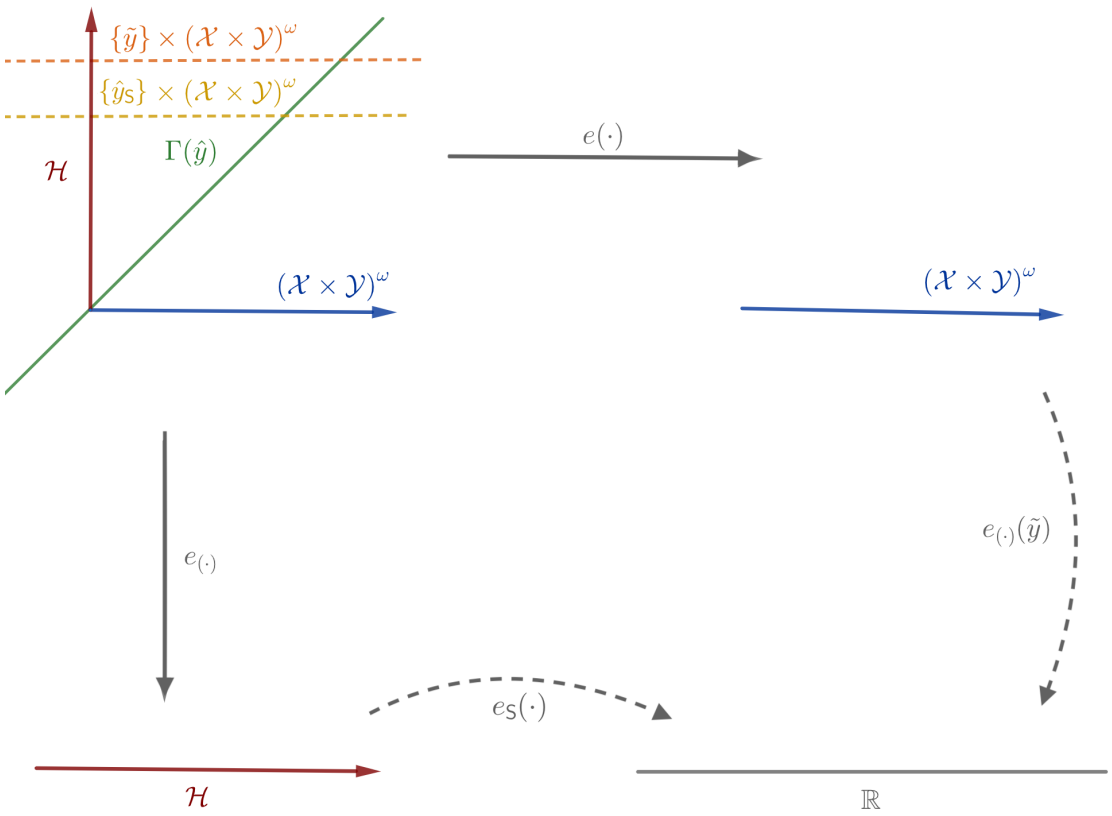

Figure 1: Slicing the Cost Function $c : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y})^\omega \to \mathbb{R}$

training data is "aspect blind" to overfitting: Law of Large Numbers reasoning does not apply in this regime.

Consider that a sequence of datasets

$$\begin{array}{rcl} S_1 & = & (x_1, y_1) \in (\mathcal{X} \times \mathcal{Y})^1, \\ S_2 & = & (S_1, (x_2, y_2)) \in (\mathcal{X} \times \mathcal{Y})^2, \\ & \vdots & \\ S_m & = & (S_{m-1}, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m, \\ & \vdots & \end{array}$$

with each $S_j \sim_{\mathrm{iid}} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$, induces a sequence of models

$$\hat{y}_{S_1}, \hat{y}_{S_2}, \ldots, \hat{y}_{S_m}, \ldots \in \mathcal{H}.$$

The sequence of models consequently induces a sequence of finite sequences of costs

$$\begin{array}{rcl} c_{\hat{y}_{S_1}}(S_1) & = & c_{\hat{y}_{S_1}}(x_1, y_1) \in \mathbb{R}, \\ c_{\hat{y}_{S_2}}(S_2) & = & (c_{\hat{y}_{S_2}}(x_1, y_1), c_{\hat{y}_{S_2}}(x_2, y_2)) \in \mathbb{R}^2, \\ & \vdots & \\ c_{\hat{y}_{S_m}}(S_m) & = & (c_{\hat{y}_{S_m}}(x_1, y_1), \ldots, c_{\hat{y}_{S_m}}(x_m, y_m)) \in \mathbb{R}^m, \\ & \vdots & \end{array}$$

which clearly is not guaranteed to be iid, unless miraculously the cost functions

$$c_{\hat{y}_{S_1}}, c_{\hat{y}_{S_2}}, \ldots, c_{\hat{y}_{S_m}}, \ldots$$

all induce the same measure $\mathbb{P}_{c_{\hat{y}}(\mathcal{X}\times\mathcal{Y})}$ on $\mathbb{R}$, for which there is no apriori reason to suppose. For each such $\hat{y}_{S_j}$, LLN still holds in the sense that there is, for $\varepsilon, \delta > 0$, a number $m_j > 0$ for which

$$\mathbb{P}_{(\mathcal{X}\times\mathcal{Y})^m}\left(\left|e_{(\cdot)}(\hat{y}_{S_j}) - \mathbb{E}(c_{\hat{y}_{S_j}})\right| > \varepsilon\right) < \delta$$

whenever $m > m_j$. Still, there is no reason to suppose that $m_j < j$, and even if it were, small probability events still exist: the search over $\mathcal{H}$ in the supervised learning setting incentivizes discovery of such events, c.f. section 3.2.

One may place an appropriate measure $\mathbb{P}_\Gamma$ on $\Gamma(\hat{y})$—in fact, this pullback naturally inherits from measures on $(\mathcal{X}\times\mathcal{Y})^\omega$—and certainly iid samples $S \in (\mathcal{X}\times\mathcal{Y})^\omega$ induce iid samples $\hat{y}_S \sim_{\text{iid}} \mathbb{P}_\Gamma$, but statements of events on this set do not extend to iid conditions on sequences of costs. For such, we must fix and isolate our attention to slices $\{\hat{y}_S\} \times (\mathcal{X}\times\mathcal{Y})^\omega \subset \Gamma(\hat{y})$ (see fig. 2), for which sample $S' = ((x'_1, y'_1), \ldots, (x_k, y'_k)) \sim_{\text{iid}} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ induces truly independent and identically distributed sample $c_{\hat{y}_S}(S') := (c_{\hat{y}_S}(x'_1, y'_1), \ldots, c_{\hat{y}_S}(x'_k, y'_k)) \sim_{\text{iid}} \mathbb{P}_{c_{\hat{y}_S}(\mathcal{X}\times\mathcal{Y})}$.

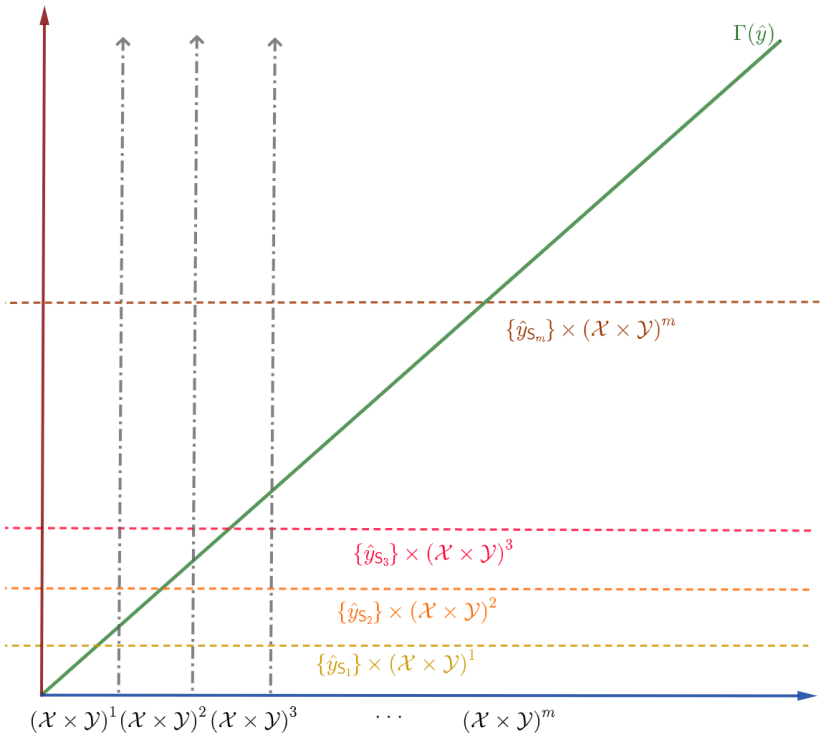

Figure 2: Sequences of models $\hat{y}_{S_1}, \hat{y}_{S_2}, \ldots, \hat{y}_{S_m}, \ldots$

While we therefore cannot rely on empirical risk $e_S(\hat{y}_S)$ by itself to reflect 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 verify that validation performance continues to improve [5]. An onset of validation performance degradation can be interpreted as indication of overfitting. Illustrations of overfitting in the literature (e.g. [1], [3], [10]) display performance on training data compared with performance on holdout data, often parameterized by model complexity or training step ([8], [7]).

While discussions of overfitting many times consider generalization against model complexity, and therefore present performance across *models*, we introduce a test, for a fixed model, based on classic concentration inequalities, with respect to which overfitting may be *quantitatively* defined, relying on comparison of validation performance to training set performance. We reason that because model construction uses and depends on (minimization with respect to $\tilde{y}$ of) $e_S(\tilde{y})$,

we ought to 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 draw rigorous statistical conclusions. The test we provide amounts to much of the same as common stopping criteria for training, though the grounds we give are both grounded in the math and provide threads for distinguishing causes of error.

3 Detecting Overfitting

3.1 The Test

We consider only 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 and identically distributed 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 $1 - e^{-2\varepsilon^2 k}$.[5] 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 (as discussed above) with respect to the $c_{\hat{y}_S}$-induced measure on $\mathbb{R}$, the 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, a bound of the form in (7) is exactly what we desire from a generalizing model $\hat{y}_S$.

We first collect some definitions.

Definition 3.1. Let $S \sim_{iid} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ and $\hat{y}_S \in \mathcal{H}$. We say that $\hat{y}_S$ *ε-overfits* S if

$$e_S(\hat{y}_S) < \mathbb{E}(c_{\hat{y}_S}) - \varepsilon.$$

Similarly, $\hat{y}_S$ *ε-underfits* S if $e_S(\hat{y}_S) > \mathbb{E}(c_{\hat{y}_S}) + \varepsilon$. Finally, $\hat{y}_S$ *ε-generalizes* model $\hat{y}_S$ neither ε-overfits nor ε-underfits S.

Proposition 3.1 (Test for Overfitting)**.** Suppose that model $\hat{y}_S$ $\varepsilon/2$-generalizes (definition 3.1, inequality (7)). 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 $\mathbb{P}\left(\left|\mathbb{E}(c_{\hat{y}_S}) - e_{(\cdot)}(\hat{y}_S)\right| > \frac{\varepsilon}{2}\right)$. □

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

Notice that use of holdout data for evaluation by itself provides an absolute approximation of performance, while in tandem with training data, we gain quantified (un)certainty specifically about generalization. Finally, the probability in (8) depends on the size of validation data, but not on the size of training data. This conclusion is correct: while we would like more training data to correlate with higher likelihood of performance, the problem in section 2.2 indicates that such intuition may not find a straightforward grounding in probability. Presumably, one may be less inclined to hypothesize satisfactory model performance when training with little data. The intuition finds security in PAC learnability, absent which there is no obvious guaranteed connection between size of (training) data and performance; we discuss this issue further in section 3.3.

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 $S' \sim_{\text{iid}} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ as well. It may not be. For there may be some form of distributional shift according to which $S' \sim_{\text{iid}} \mathbb{P}'_{\mathcal{X}\times\mathcal{Y}}$, in which case we cannot guarantee the bound in (8), at least not if the expectation $\mathbb{E}(c_{\hat{y}_S})$ is computed with respect to the original measure $d\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$. In other words, instantiation of event $\left\{\left|e_S(\hat{y}_S) - e_{(\cdot)}(\hat{y}_S)\right| > \varepsilon\right\}$ by inequality $|e_S(\hat{y}_S) - e_{S'}(\hat{y}_S)| > \varepsilon$ may suggest:

1. an unlikely sample S' was received (all the hypotheses hold),
2. $\hat{y}_S$ does not generalize $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ with respect to $c_{(\cdot)}$ (overfitting), or
3. $S' \not\sim_{\text{iid}} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ (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 $\hat{y}_S$ $\frac{\varepsilon}{2}$-generalizes (eq. (7)) and 2. $S' \sim_{\text{iid}} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$—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.

Finally, while this test is expressed with respect to the cost function $c_{\tilde{y}}$ or $c_{\hat{y}_S}$, it need not be so limited. In fact, any map $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ may be used to probe the distribution, substituting the appropriate concentration inequality depending on the range of f. When $f(\mathcal{X} \times \mathcal{Y})$ 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, as a means of testing to eliminate or isolate the above obfuscating condition #3.

3.3 Loosening Uniform Bounds

We conclude with commentary on the merits of this test. The bound in eq. (8) is perhaps unsurprising and at first glance offers little upside beyond the performance guarantee as provided by eq. (6), which ensures approximation of empirical mean (of holdout data) to the true mean. Indeed, one may argue that overfitting, in the sense of definition 3.1, induces little cause for concern: as long as performance $\mathbb{E}(c_{\hat{y}_S})$ is "good enough," (as approximated by $e_{S'}(\hat{y}_S)$) it may not particularly matter whether or that training data performance matches a model's generalization performance. On the other hand, guarantees of the sort which PAC learnability provides ensure that the output of a training algorithm is near optimal in a hypothesis class. In the presence of overfitting, one may not know whether better than 'good enough' is achievable. Generalization *with training data* provides confidence that empirical risk minimization (2) approximately realizes risk minimization (5) *in the absence of uniform (PAC) guarantees*. The test is a workable mechanism for checking that there is little gap between performance a hypothesis class may achieve on data and on the data's distribution.

Consider, for example, fig. 3 which compares level sets for $\mathbb{E}(c_{(\cdot)})$ and $e_S(\cdot)$. Learnability, as described by uniform convergence and notions of representability (c.f. [9, §4.1]), guarantees that these profiles roughly track each other, which is *sufficient* for generalization of output model $\hat{y}_S$: if

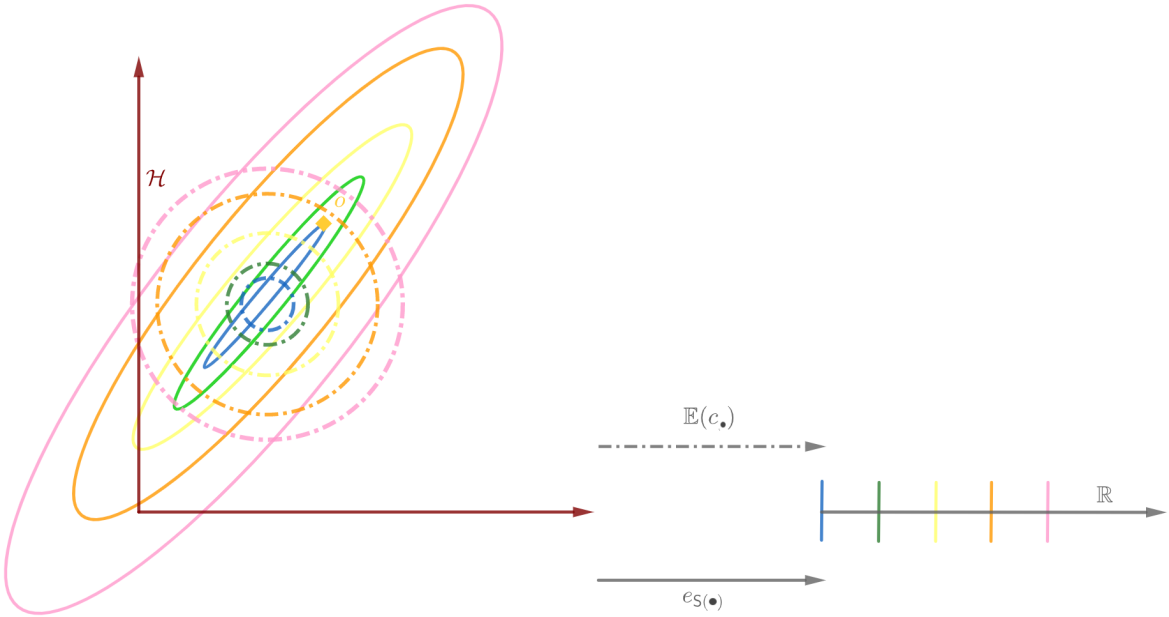

Figure 3: Model o overfits.

the value of $\mathbb{E}(c_{(\cdot)})$ and $e_S(\cdot)$ are roughly approximate *everywhere* in $\mathcal{H}$, then they certainly are at a particular point. On the other hand, learnability objectives ultimately seek generalization of the output, namely that $\mathbb{E}(c_{(\cdot)})$ and $e_S(\cdot)$ are roughly approximate *at* $\hat{y}_S$; how they compare in other regions of $\mathcal{H}$ may be immaterial.

We underscore the point. PAC results guarantee not only that an algorithm will return an optimal (in the hypothesis class) model, but that the sample complexity with respect to which the algorithm is expected to reliably work is *independent of distribution*. Guarantees of this form are helpful in providing confidence ahead of time that the learning endeavor is not misguided. On the other hand, practitioners often engage in the tackling the learning problem irrespective of knowledge or other assurances that their class is PAC learnable. Moreover, PAC learnability does not cover the intermediate case that some distributions may require a larger sample complexity (some tasks are harder to learn than others), and that there may be no uniform bound over all measures, even if there are some over subsets. Still, assurance that the output of training generalizes does not *require* that the hypothesis class be PAC learnable, i.e. that uniform bounds hold. Rather: uniform bounds, when they exist, provide a conceptual framework and analytic setting wherein a class of results may be generated, in the absence of which, we would nevertheless like to be able to say *something*.

We collect this commentary into a definition.

Definition 3.2. Let $\mathcal{P}$ be a collection of probability measures on $\mathcal{X} \times \mathcal{Y}$. We say that a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is *$\mathcal{P}$-learnable* if there is sample complexity $\mu : (0,1) \to \mathbb{N}$ and algorithm $\hat{y} : (\mathcal{X} \times \mathcal{Y})^\omega \to \mathcal{H}$ for which $\mathbb{P}_{(\mathcal{X}\times\mathcal{Y})^m}\left(\mathbb{E}(c_{\hat{y}_{(\cdot)}}) - \inf_{y \in \mathcal{H}} \mathbb{E}(c_y) > \varepsilon\right) < \delta$ whenever $m > \mu(\varepsilon, \delta)$ and $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}} \in \mathcal{P}$. We say that $\mathcal{H}$ is *$\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$-learnable* if it is $\{\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}\}$-learnable.

Whereas PAC learnability typically guarantees (approximate) optimality independent of measure, $\mathcal{P}$-learnability expressly restricts measures for which $\mathcal{H}$ is (uniformly) suited to learn. While PAC learnability is powerful in providing guarantees absent prior assumptions about the measure from which data is drawn, this generality also inhibits the usefulness of prior knowledge: for a class is PAC learnable or not irrespective of what is known regarding the data. One might imagine, for example, that assurance data is subgaussian may be relevant, in that subgaussianity demarcates a class of measures for which a hypothesis class is suited to fit.

4 In Simulation

Code implementing this test can be found at https://github.com/schmidttgenstein/qudost.git. In each of the following illustrations, we plot empirical performance (accuracy for classification problems) with respect to both training and holdout data 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). We fix the precision in each case, denoted by the dashed red line in the right figures, and report the probability bound a lá proposition 3.1. Per Hoeffding's inequality (9), this precision may be made finer with more data. Worth noting that the test in proposition 3.1 does not intrinsically relate to early stopping: a model may overfit and cease to overfit at various epochs in training (see, e.g., fig. 5).

Results in fig. 4 and fig. 5 use generated data and a multilayer perceptron for binary classification. Results in fig. 6 and fig. 7 use a simple convnet on MNIST data.

4.1 Simulated Data

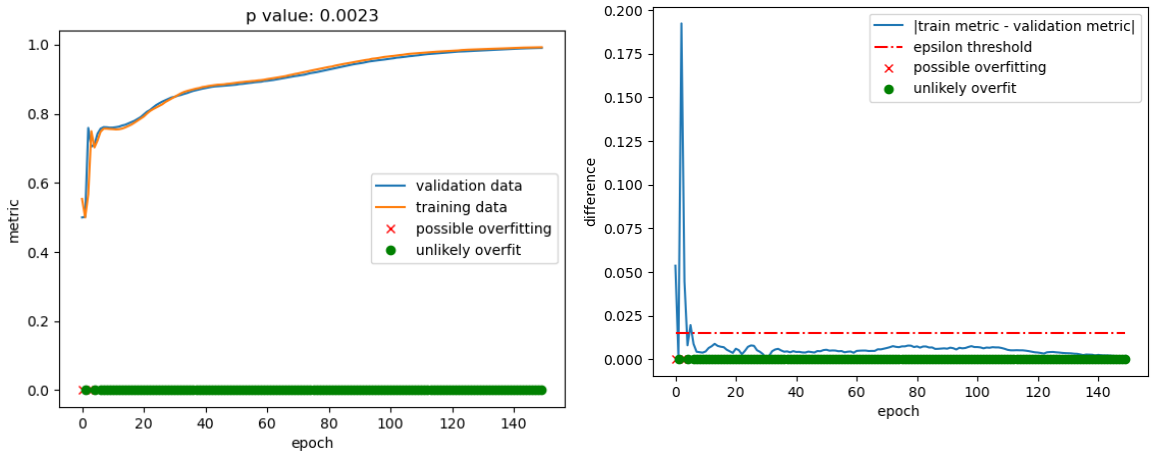

Figure 4: $k = 15,000$

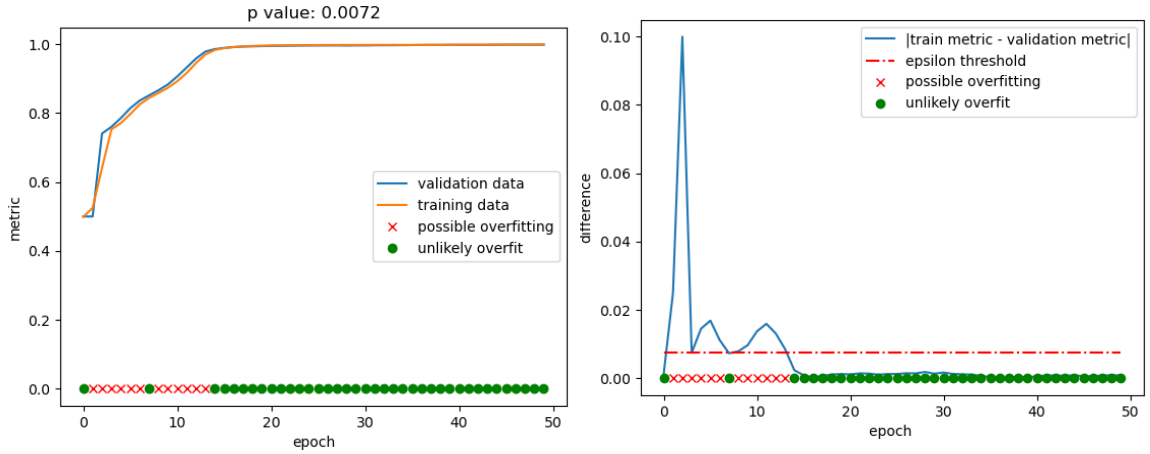

Figure 5: $k = 50,000$

4.2 MNIST

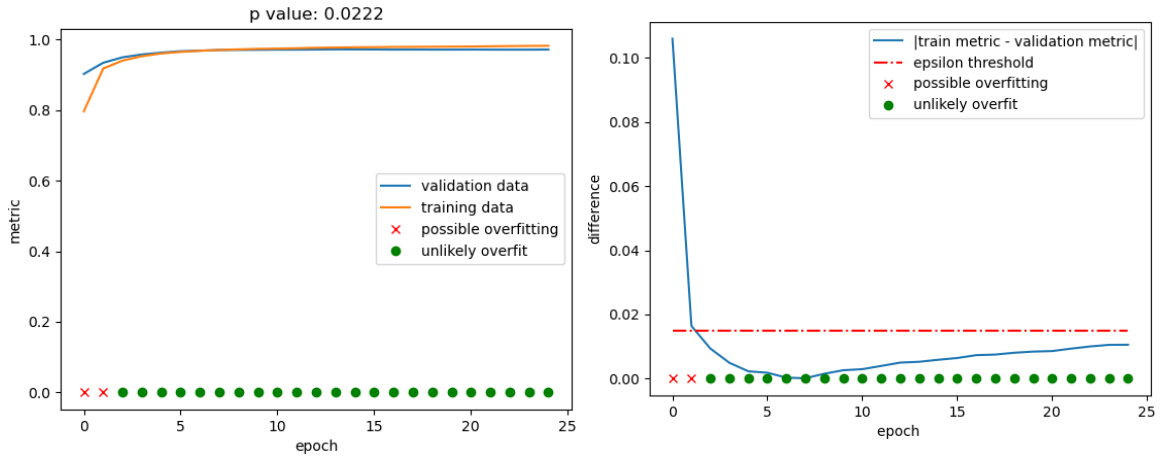

Figure 6: MNIST $k = 1000$

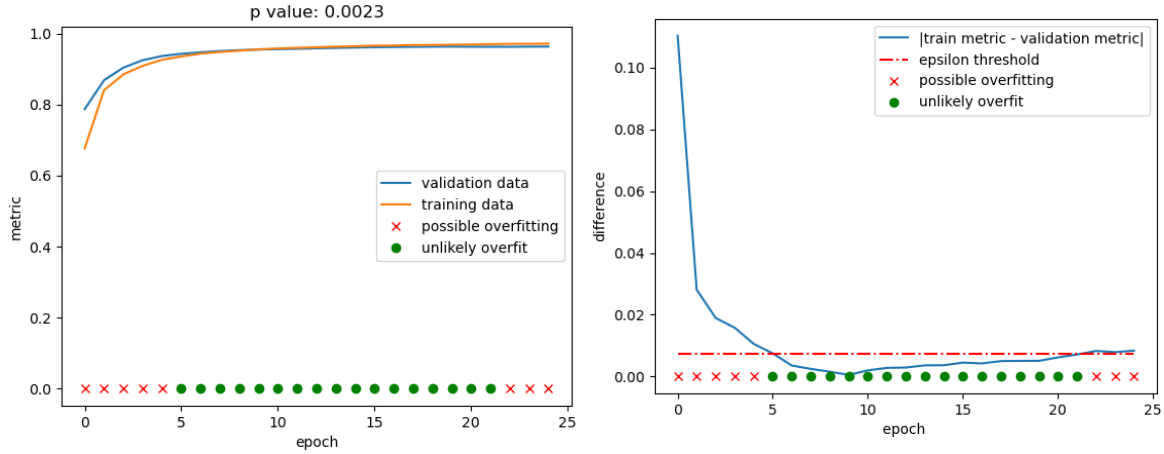

Figure 7: MNIST $k = 6000$

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:

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

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], [9], [4]).

References

[1] J. Bilmes. Underfitting and overfitting in machine learning. *UW ECE Course Notes*, 2020. 5

[2] S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 02 2013. 10

[3] B. Ghojogh and M. Crowley. The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial. *arXiv [stat.ML]*, 2019. 1, 5

[4] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018. 3, 10

[5] L. Prechelt. *Early Stopping — But When?*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 5

[6] E. Riehl. *Category Theory in Context*. Dover, 2016. 3

[7] R. Roelofs. Measuring generalization and overfitting in machine learning. 2019. 5

[8] R. Roelofs, V. Shankar, B. Recht, S. Fridovich-Keil, M. Hardt, J. Miller, and L. Schmidt. A meta-analysis of overfitting in machine learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 5

[9] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning*. Cambridge University Press, 2014. 3, 7, 10

[10] M. Valdenegro-Toro and M. Sabatelli. Machine learning students overfit to overfitting. *arXiv [cs.LG]*. 5