Classical Statistical (In-Sample) Intuitions Don't *Generalize* Well: A Note on Bias-Variance Tradeoffs, Overfitting and Moving from Fixed to Random Designs

Alicia Curth 1

Abstract

The sudden appearance of modern machine learning (ML) phenomena like double descent and benign overfitting may leave many classically trained statisticians feeling uneasy – these phenomena appear to go against the very core of statistical intuitions conveyed in any introductory class on learning from data. The historical lack of earlier observation of such phenomena is usually attributed to today's reliance on more complex ML methods, overparameterization, interpolation and/or higher data dimensionality. In this note, we show that there is another reason why we observe behaviors today that appear at odds with intuitions taught in classical statistics textbooks, which is much simpler to understand yet rarely discussed explicitly. In particular, many intuitions originate in *fixed design* settings, in which in-sample prediction error (under resampling of noisy outcomes) is of interest, while modern ML evaluates its predictions in terms of generalization error, i.e. out-of-sample prediction error in random designs. Here, we highlight that this simple move from fixed to random designs has (perhaps surprisingly) far-reaching consequences on textbook intuitions relating to the bias-variance tradeoff, and comment on the resulting (im)possibility of observing double descent and benign overfitting in fixed versus random designs.

1. Introduction

The strikingly good performance of highly overparametrized machine learning (ML) models trained to zero loss, ubiquitously observed in recent years (Neyshabur et al., 2014; Bartlett et al., 2020; Zhang et al., 2021; Belkin, 2021), appears to contradict all classical statistical intuitions about overfitting – and as such, probably leaves many classically trained statisticians confused and somewhat uneasy. This definitely applies to myself, and has left me puzzled, questioning a lot of what we are taught in graduate statistics

classes. Should we no longer be concerned about overfitting and bias-variance tradeoffs? If modern ML methods defy the intuitions built in decades of statistics research, is there something that has changed recently? Or is modern ML simply a magic bullet we had been missing all along?

This note therefore aims to better understand the sources of the discrepancies between classical statistical intuitions surrounding the bias-variance tradeoff and modern ML phenomena like double descent (Belkin et al., 2019) and benign overfitting (Bartlett et al., 2020). The historical lack of earlier observation of such phenomena is usually attributed to today's reliance on more complex ML methods, overparameterization, interpolation and/or higher data dimensionality (Belkin, 2021). Here, we will explore another reason why we observe behaviors today that appear at odds with intuitions taught in classical statistics textbooks, which is much simpler to understand yet rarely discussed explicitly. We highlight that statistics historically focussed on fixed design settings (Rosset & Tibshirani, 2019), where in-sample prediction error is of interest, while modern ML evaluates its predictions in terms of generalization error, i.e. out-ofsample prediction error – and this seemingly small change has surprisingly far-reaching effects on textbook intuitions.

Outlook. Sec. 2 introduces fixed and random design setups. In Sec. 3, we revisit the bias-variance tradeoff. Using a simple k-Nearest Neighbor estimator on low-dimensional data, we show that the classical bias-variance tradeoff intuition ("Variance increases with model complexity, Bias decreases with model complexity") does not necessarily hold when considering out-of-sample prediction error: there can exist regimes where both bias and variance decrease when complexity is *decreased*. That is, we show that classical intuitions relating bias, variance and model complexity break already in the absence of modern ML methods, overparameterization, interpolation and high-dimensional data, highlighting that they cannot be solely responsible for the emergence of surprising statistical phenomena. In Sec. 4 we then comment on the recent appearance of double descent, and show that one reason for the historical absence of double descent shapes in the literature may be that they cannot appear in fixed design settings. In Sec. 5, we comment on benign overfitting, and discuss when and why it is possible.

¹ University of Cambridge. <amc253@cam.ac.uk>.

2. Problem setup: Fixed vs Random designs

It appears that much of the statistics literature has historically focussed on so-called fixed design settings (Rosset & Tibshirani, 2019), where *in-sample* prediction error (which assumes that test-time inputs will be the same as training inputs but noisy labels are *re-sampled*) is used to measure test-time model performance. The modern ML literature, on the other hand, is almost exclusively interested in *generalization* to new inputs (Goodfellow et al., 2016; Murphy, 2022) – i.e. out-of-sample prediction error, where *both* test inputs and test labels are newly sampled.

Formally, as in e.g. Rosset & Tibshirani (2019), assume we observe a sample of labeled training observations $\{(x_i,y_i)\}_{i=1}^n$, consisting of pairs of inputs $x\in\mathcal{X}\subset\mathbb{R}^d$ and outcomes $y\in\mathbb{R}$, which are jointly sampled i.i.d. from some distribution P. Outcomes are related to inputs as $y=f^*(x)+\epsilon$ where $f^*(x)=\mathbb{E}[y|x]$ and we assume that $\epsilon=y-f^*(x)$ is independent of x, implying homoskedastic variance $\sigma^2=\mathrm{var}(y|x)$.

Setting A: Fixed design. In a fixed design setting it is assumed that *the same* inputs $\{x_i\}_{i=1}^n$ as the training inputs 1 are encountered at test-time, but with new realizations of the outcomes $\{\tilde{y}_i\}_{i=1}^n$ drawn independently from the conditional law of $y_i|x_i$. If we learn a predictor $\hat{f}(\cdot): \mathcal{X} \to \mathbb{R}$ from the training data, we thus ultimately want to minimize its in-sample prediction error

$$ERR_{is} = \mathbb{E}_{\tilde{y}} \left[\frac{1}{n} \sum_{i=1}^{n} (\tilde{y}_i - \hat{f}(x_i))^2 \right]$$
 (1)

Setting B: Random design (generalization). In a random design setting, we are instead interested in *generalization* of our learned predictor $\hat{f}(\cdot)$ to *new* inputs x_0 that have *not* been observed during training, but are also randomly sampled at test-time; i.e. both input and output are newly sampled as $(x_0, y_0) \sim P$. We thus ultimately want to minimize the out-of-sample prediction (or: generalization) error:

$$ERR_{oos} = \mathbb{E}_{x_0, y_0} \left[(y_0 - \hat{f}(x_0))^2 \right]$$
 (2)

3. How the move to random design settings affects the bias-variance tradeoff

Below, we now explore how changing from fixed to random design settings affects our intuitions around the biasvariance tradeoff. Using a simple example with k-nearest neighbor estimators, we highlight that – perhaps surprisingly – the classical intuition around the bias-variance tradeoff ("Variance increases with model complexity, bias decreases with model complexity", see below) does not necessarily hold up when we consider the random design setting instead of the classical fixed design.

Preliminaries: k-Nearest Neighbor estimators. To show this, we examine the bias and variance of very simple estimators – k-Nearest Neighbor (k-NN) estimators – to highlight that surprising behaviors of bias and variance are *not actually unique* to more complex modern ML methods. Recall that, for any input x, a k-NN estimator issues predictions that are averages of outcomes across the k nearest training examples whose indices are collected in $N^k(x)$, i.e.

$$\hat{f}(x) = \frac{1}{k} \sum_{i=1}^{n} \mathbf{1} \{ i \in N^k(x) \} y_i = \sum_{i=1}^{n} w_{k,i}(x) y_i$$
 (3)

Here, we sometimes collect the nearest neighbor weights in the $k \times 1$ vector $\mathbf{w}_k(x) = [w_{k,1}(x), \dots, w_{k,n}(x)] = [\frac{1}{k}\mathbf{1}\{1 \in N^k(x)\}, \dots, \frac{1}{k}\mathbf{1}\{n \in N^k(x)\}]$. Recall also that complexity in k-NN estimators is *inversely* related to k: the 1-NN estimator is the most complex estimator in this class while the n-NN estimator is simply the sample mean.

Expressions for bias and variance. For some test input $x \in \mathcal{X}$, the bias and variance of a k-NN estimator for given training inputs $\{x_i\}_{i=1}^n$ are (Hastie et al., 2009, Ch. 7.3)

$$\operatorname{Bias}_{k}(x) = f^{*}(x) - \sum_{i=1}^{n} w_{k,i}(x) f^{*}(x_{i})$$
 (4)

$$\operatorname{Var}_k(x) = \operatorname{Var}_k = \frac{\sigma^2}{k}$$
 (5)

3.1. The classical bias-variance tradeoff intuition: in-sample view

Statistics textbooks often discuss the bias-variance tradeoff by considering what happens to bias and variance of predictions for previously observed training inputs as kincreases (Hastie & Tibshirani, 1990, Ch. 3.3). For the variance term, this is easily read off from Eq. (5): the variance of predictions due to noise in the training labels y_i is always monotonically decreasing in k. When considering how the bias of k-NN estimators at a training input x_j is likely to evolve, note that a 1-NN estimator – which has $N^1(x_j) = \{j\}$ and hence $\mathbf{w}_1(x_j) = \mathbf{e}_j$ (with \mathbf{e}_j the j-th unit vector) – will always have no bias, as $\sum_{i=1}^n w_{1,i}(x_j) f^*(x_i) = f^*(x_j)$. As k increases, this bias is likely to increase because the weighted average $\sum_{i=1}^n w_{k,i}(x_j) f^*(x_i)$ involves more terms with $f^*(\cdot)$ different from $f^*(x_j)$ (Hastie & Tibshirani, 1990, Ch. 3.3).

This is precisely the intuition behind the bias-variance tradeoff as presented in e.g. Hastie & Tibshirani (1990, Ch. 3.3) and Hastie et al. (2009, Ch. 7.3): As the complexity of the estimator increases (k decreases), variance is expected to

¹Classical fixed design settings also assume that the x_i are non-random (e.g. because they were designed). Above, we allow randomness in the x_i and only require train- and test-time realizations of x_i to be the same. Rosset & Tibshirani (2019) refer to this as the 'Same-x' setting. For the purpose of this note, making a distinction between the two is not necessary.

monotonically increase while bias is expected to monotonically decrease².

3.2. New territories: Bias-bias-variance tradeoffs in random design settings

Next, we highlight that the bias-variance tradeoff intuition does not necessarily hold up even for simple k-NN estimators once we move to the random design setting. This is because bias no longer monotonically decreases as complexity increases. Intuitively, this is because – as there is no training point x_j with exactly the same input value as the new test point x_0 – there generally is no perfect match with zero bias among the neighbors: that is, there does not necessarily exist any training example x_j so that $f^*(x_0) = f^*(x_j)$, which means that the 1-NN estimator does not necessarily have the lowest bias. This intuition is illustrated in Fig. 1 using a stylized example.

To make this more formal, we note that, by adding *and* subtracting $f^*\left(\sum_{i=1}^n w_{k,i}(x_0)x_i\right)$ from Eq. (4), it is always possible³ to rewrite the bias term as⁴:

$$\begin{aligned} \operatorname{Bias}_k(x_0) &= \underbrace{\left(f^*(x_0) - f^*\left(\sum_{i=1}^n w_{k,i}(x_0)x_i\right)\right)}_{\operatorname{NeighborMatchingBias}_k(x_0)} + \\ &\underbrace{\left(f^*\left(\sum_{i=1}^n w_{k,i}(x_0)x_i\right) - \sum_{i=1}^n w_{k,i}(x_0)f^*(x_i)\right)}_{\operatorname{AveragingBias}_k(x_0)} \end{aligned}$$

This is interesting because the first term captures the bias in prediction arising due to mismatches between the test example and the selected neighbors in *input space*; this term is zero whenever the neighbor weights reconstruct the test input perfectly as $x_0 = \sum_{i=1}^n w_{k,i}(x_0)x_i$. The second term captures the bias arising in any estimator that predicts using weighted averages due to nonlinearity of the true (unknown) prediction function f^* : if f^* was linear, then the averaging

⁴We borrow the idea for this decomposition from the causal inference literature where it appeared in Kellogg et al. (2021) when comparing the biases of matching- and synthetic control estimators.

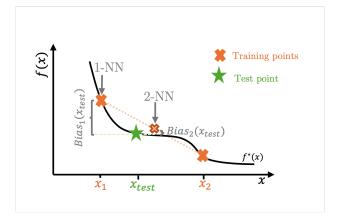


Figure 1. Stylized example: The 1-NN estimator does not necessarily have the lowest bias when considering test inputs different from training inputs.

operation and application of the function would commute (in which case an estimator perfectly reconstructing the inputs will incur zero bias). Note that a 1-NN estimator incurs no averaging bias because $\mathbf{w}_1(x)$ has only one nonzero element, but could incur significant neighbor matching bias if x_0 is far from its nearest training neighbor x_{j^*} . A k-NN estimator with average input $\sum_{i=1}^n w_{k,i}(x_0)x_i$ closer to x_0 in input space than its nearest neighbor x_{j^*} would incur less neighbor matching bias but may incur significant averaging bias depending on the nonlinearity of the underlying $f^*(x_i)$.

When considering *in-sample* prediction at training point x_j , the 1-NN estimator has $\sum_{i=1}^n w_{1,i}(x_j)x_i = x_j$; thus both bias terms are exactly zero. For out-of-sample prediction, the NeighborMatchingBias $_1(x_0)$ term can be substantial depending on the distance of any input to its nearest neighbor in the training data. A k-NN estimator with k>1 can improve this component of the bias, but will likely worsen the averaging bias (if the underlying f^* is nonlinear). This is precisely the reason why the bias term no longer necessarily behaves monotonically in the random design setting and is illustrated in Fig. 1.

3.3. Empirical investigation: How do bias terms evolve in- and out-of-sample?

Next, we empirically investigate whether these theoretical predictions indeed hold up: does the bias term behave differently in- and out-of-sample? Here, we use a nonlinear DGP adapted from Friedman (1991), with

$$f^*(x) = 10sin(\pi x_1 x_2) + 20(x_3 - \frac{1}{2})^2 + 10x_4 + x_5$$
 (6)

and let $y = f^*(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and the $d{=}5$ features $x \in [0, 1]^d$ are sampled independently from Unif(0, 1). We use n = 100 as training data and sample 100 further out-of-sample test examples.

Note that the term bias-variance tradeoff is not to be confused with the term bias-variance decomposition. The term bias-variance decomposition (Hastie et al., 2009, Ch. 7.3) refers to the fact that (for any estimator \hat{f}) the mean squared error of estimation can always be decomposed into a squared bias and a squared variance term, i.e. $\mathbb{E}[(f^*(x) - \hat{f}(x))^2] = Bias^2(x) + Var(x)$. Further, the decomposition of the mean squared prediction error incurs an additional term due to noise in outcome, i.e. $\mathbb{E}[(y - \hat{f}(x))^2] = Bias^2(x) + Var(x) + \sigma^2$.

³Note that this decomposition is useful not only for k-NN estimators but for any estimator issuing predictions that are weighted averages of training outcomes, which includes the broad class of regression smoothers (Hastie & Tibshirani, 1990).

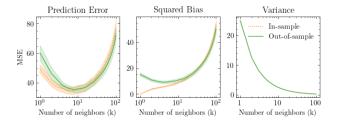


Figure 2. The classical bias-variance tradeoff occurs in insample prediction error, but not in out-of-sample prediction error – where decreasing k can decrease both bias and variance. The behavior of Prediction error, Bias and Variance by k for kNN estimators, in-sample (orange) and out-of-sample (green). Data simulated using $f^*(x)$ from Eq. (6) with $\sigma=5$.

Non-monotonic behavior of out-of-sample bias. In Fig. 2, we plot prediction error (ERR_{is} and ERR_{oos}), squared bias and variance while simulating data with $\sigma = 5$. We observe that the in-sample terms behave as expected (orange): Bias monotonically decreases in complexity (increases in k), while variance monotonically increases in complexity (decreases in k). This leads to the classical Ushaped tradeoff in in-sample prediction error ERR_{is} . While the variance behaves similarly out-of-sample, the bias term shows a strikingly different behavior: there is a U-shape in the bias itself, as the most complex 1-NN estimator indeed does not have the lowest bias. Instead, intermediate values of k incur lowest bias. Therefore, the out-of-sample prediction error ERR_{oos} presents a more pronounced Ushape than ERR_{is} : due to the higher bias, the low-k k-NN estimators perform worse out-of-sample than in-sample.

The effect of sampling noise. This difference between in- and out-of-sample prediction becomes even more salient when we vary the outcome-noise level σ in Fig. 3. (Note that when σ changes, bias remains constant and only the variance term changes; see Eqs. (4) and (5).) In the absence of outcome noise ($\sigma = 0$), k-NN estimators with higher

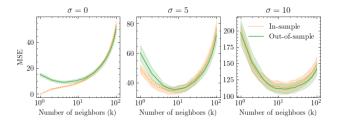


Figure 3. Bias alone can cause the U-shape in out-of-sample prediction error (while in-sample the U-shape is caused by the bias-variance tradeoff and thus appears only when $\sigma > 0$). The behavior of prediction error by k for kNN estimators, in-sample (orange) and out-of-sample (green) across different levels of noise in outcomes σ .

complexity are *always better* for in-sample prediction. This is *not true* for out-of-sample prediction: at $\sigma=0$, the prediction error is equal to the bias term, which by Fig. 2 itself has a U-shape in k – thus, k-NNs with intermediate level of complexity perform best at prediction *even in the absence of outcome noise*. With $\sigma>0$, we observe a bias-variance tradeoff for in-sample prediction, while bias and variance lead to the *the same out-of-sample ranking* of estimators for $1 \le k \le 10$ (i.e. there is no tradeoff between bias and variance in this interval of k). In Appendix B, we also show that the terms NeighborMatchingBias $_k(x_0)$ and AveragingBias $_k(x_0)$ indeed behave as expected.

3.4. Conclusion: The bias-variance tradeoff does not necessarily hold out-of-sample as it does in-sample

In this section, we discovered that (and why) moving from in-sample prediction to out-of-sample prediction can have surprisingly stark effects on the classical textbook intuition relating model complexity to bias and variance. In particular, we demonstrated that while the bias-variance tradeoff intuition "Variance increases with model complexity, bias decreases with model complexity" applies when considering the in-sample setting, it does not necessarily hold when considering out-of-sample prediction: we showed that there exist regimes where both bias and variance decrease when model complexity is decreased.⁵

That is, we showed that classical statistical intuitions regarding the bias-variance tradeoff can break already in the (standard) underparametrized regime for extremely simple estimators and data-generating processes – as a consequence of simply moving from in-sample prediction to out-of-sample prediction! As we show in the remainder of this note, this observation is crucial to understanding *why* recently observed phenomena like double descent and benign overfitting contradict textbook wisdoms: overparameterization and interpolation are not responsible for breaking the bias-variance tradeoff intuitions on their own. Only in combination with a move in interest from fixed to random designs do the surprising modern phenomena arise.

4. Reconciling double descent with textbook intuitions about the bias-variance tradeoff

The double descent phenomenon (Belkin et al., 2019) received considerable attention recently as its existence *appears to* contradict the classical bias-variance tradeoff. In particular, Belkin et al. (2019) highlighted that when plotting *generalization error* against the total number of model

⁵Note that this is related to yet different from Hastie et al. (2022), who show for linear regression that both (out-of-sample) bias and variance can decrease as model complexity *increases* in the *overparameterized* regime.

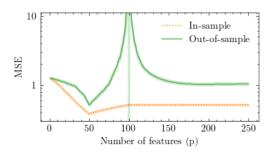


Figure 4. A double descent shape appears only in out-of-sample prediction error, not in in-sample prediction error. The behavior of in- and out-of-sample prediction error (ERR_{is} and ERR_{oos}) as we vary the number of features p included in a linear regression with n=100 training examples. In the underlying DGP, $\sigma=\frac{1}{2}$ and only the first s=50 features are used in f^* , all other p-s features are irrelevant for prediction.

parameters p, one observes a U-shaped error curve while $p \le n$ (where error first decreases and then increases). Once one lets p > n, however, error experiences a *second* descent in the so-called interpolation regime where the training data can be fit perfectly – resulting in a double descent shape.

In Curth et al. (2023), we demonstrated that in the non-deep examples of double descent in Belkin et al. (2019) (using trees, boosting and linear regressions), this non-monotonic behavior in error is directly caused by *a change* in how model parameters are added at p=n. Further, we showed that once a measure for the *effective parameters* (Hastie & Tibshirani, 1990, Ch. 3.5) used by a model is placed on the x-axis, the double descent curves fold back into U-shaped curves, as adding additional raw model parameters in the interpolation regime leads to a decrease in test-time *effective* parameters. In doing so, we already presented two resolutions to the ostensible tension between double descent and textbook intuitions regarding the bias-variance tradeoff.

Here, we briefly wish to elaborate on a *third* resolution — more closely related to the topic of this note — that we only alluded to in Curth et al. (2023, Appendix C.2). In particular, we wish to highlight that while Belkin et al. (2019) argued that the historical absence of double descent curves in the statistics literature is likely due to a lack of the use of (unregularized) overparameterized models, there is a second reason we consider at least as important: as highlighted above, the statistics literature which developed many of the intuitions around the U-shaped curve historically considered mainly fixed-design settings, while double descent curves have been shown exclusively in random design settings.

This is not a coincidence: it is easy to see that it is actually *impossible* to observe a second descent in in-sample prediction error in the interpolation regime. That is, as the name suggests, any model in the interpolation regime issues

predictions $\hat{f}(x_i) = y_i$ for training points x_i regardless of how it is trained and parameterized (else it could not fit the training data perfectly). Thus, both in-sample bias (=0) and in-sample variance $(=\sigma^2)$ are the same for any model in the interpolation regime, and as a consequence, so is insample prediction error. In fact, it is also easy to see that the in-sample prediction error of any interpolating model is $2\sigma^2$, regardless of the number of model parameters it uses.

Using a linear regression experiment adapted from Maddox et al. (2020), we show this empirically in Fig. 4. Here, we fit a linear regression to an increasing number of features d (resulting in p=d parameters), where we use the ground truth DGP $f^*(x)=\frac{1}{\sqrt{s}}\sum_{k=l}^s x_s$ so that there are d-s irrelevant features. We observe that, indeed, a double descent shape appears only in out-of-sample prediction (generalization) error; in-sample prediction error follows the familiar U-shape in the underparameterized regime and is constant in the overparameterized regime – as is expected.

Strictly speaking, it is thus not really necessary to do any reconciling of double descent with the classical bias-variance tradeoff – they appear in different settings and thus *do not actually contradict each other*. Indeed, as we highlighted in the preceding section, even in the underparameterized regime the bias-variance tradeoff itself does not necessarily hold in the random design setting in the same way as it does in the fixed design setting. In this light, it should thus be less surprising that model behavior in the overparameterized regime differs too.

5. Can overfitting be benign? Understanding overfitting requires refining our vocabulary

Entangled in the literature on double descent is the so-called *benign overfitting* effect (Bartlett et al., 2020) – the observation that ML models can sometimes perform well despite fitting the training data *perfectly*. This appears to stand in direct contradiction with textbook intuitions, which hold that "a model with zero training error is overfit to the training data and will typically generalize poorly"(Hastie et al., 2009, Ch. 7.2, p. 221).

In this section, we argue that understanding when and why "overfitting" can sometimes be benign requires more care and *precision* in the vocabulary we use when discussing these phenomena. In fact, and somewhat surprisingly, Bartlett et al. (2020) do not precisely define the term "overfitting" (and neither do Hastie et al. (2009)). Instead, overfitting in the literature on "benign overfitting" appears to be used exchangeably with the term "interpolation" – which, in turn, refers to models that fit the training data perfectly, i.e. attain zero training error $e\hat{r}r_{train} = \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$.

There is, however, a crucial difference between the term

interpolation and the implied meaning of the term *overfitting*. Indeed, the Oxford dictionary defines "overfitting (statistics)" as: "The production of an analysis which corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably." Importantly, note that this definition implies that interpolation itself does not suffice for overfitting – the term *overfitting* already implies that model performance suffers due to interpolation. Instead, Efron & Tibshirani (1997), for example, define the amount of overfitting as $ERR - e\hat{r}r_{train}$, the excess prediction error (relative to the training error), also sometimes referred to as the degree of optimism of the training error (Efron, 1986).

This makes immediately obvious that in the original sense of the term, it is *impossible* for overfitting to be benign. In fact, "benign overfitting" is a tautology – the term overfitting itself already implies that interpolation of the training data is malignant (reflected in $ERR >> e\hat{r}r_{train}$). Similar to e.g. Muthukumar et al. (2020), who alternatively refer to the phenomenon as "harmless interpolation", we will therefore instead use the term *benign interpolation* in the remainder.

Having cleared this initial semantic hurdle, we can now move to the statistical question at the heart of this section, rephrasing it slightly: When can *interpolation* be benign? That is, when are models that interpolate the training data overfit in the sense that their prediction performance is poor, and when can interpolating models perform well in terms of prediction? Intuitively, models that can interpolate any set of training data points have the capacity to fit pure noise, thus one may expect interpolation to always lead to overfitting. Below, we highlight that – as before – the answer to this question is strongly dependent on whether one is interested in the fixed or random design setting.

5.1. Can interpolation be benign in fixed design settings? (A: No!)

If the (fixed design) in-sample prediction error is of interest, the answer is simple: interpolation cannot be benign. Indeed, by the bias-variance decomposition we trivially know that $ERR_{is} = Bias^2(\hat{f}) + Var(\hat{f}) + \sigma^2$. Because any interpolating model – for example the 1-NN estimator – predicts $\hat{f}(x_i) = y_i$, we have that $Bias(\hat{f}) = f^*(x_i) - \mathbb{E}(\hat{f}(x_i) = f^*(x_i) - f^*(x_i) = 0$ and $Var(\hat{f}) = \sigma^2$. Thus, the prediction error is dominated by the variance in outcome generation. In this case, the only time an interpolating solution is not overfit is if there is no noise in outcomes. Thus, overfitting in fixed design settings is caused by *variance due to outcome noise alone*.

5.2. Can interpolation be benign in random design settings? (A: Yes, sometimes!)

If you have made it this far in this note, you may not be surprised to discover that the switch to the random design setting changes also the answer to this question. Indeed, interpolating models can be either *more or less* overfitted in the out-of-sample setting!

Why can overfitting behavior be worse in the out-ofsample setting? In the fixed design setting, interpolating models trivially have no bias. However, as we showed in Sec. 3.2, models that interpolate the training data will generally have non-zero bias for a new test input x_0 . Thus, interpolating models incur both bias and variance terms in the out-of-sample setting. This is most easily seen by revisiting Figs. 2 and 3: the 1-NN estimator interpolates the training data but incurs no ERR_{is} in the absence of noise σ . It does, however, incur ERR_{oos} even when there is no noise in outcomes because of non-zero bias. Thus, in this example for $\sigma = 0$, $ERR_{is} = e\hat{r}r_{train} = 0$ while $ERR_{oos} >> e\hat{r}r_{train}$. That is, the model is not overfit if a fixed design setting is of interest but it is overfit if a random design setting is of interest because – unlike the fixed design setting where overfitting is due to variance alone – in random design settings overfitting can be a consequence of both the bias and the variance term!

Why can overfitting behavior be less pronounced in the out-of-sample setting? It is of course the opposite case that has received popular attention recently: some interpolating models generalize well despite their ability to fit the training data perfectly (Zhang et al., 2021; Belkin et al., 2018a;b; Bartlett et al., 2020). Intuitively, this is because some ML models can behave very differently around new test points compared to inputs observed during training. This behavior is very different from e.g. classical k-NN estimators, which always use exactly k neighbors for prediction regardless of whether point was observed during training or not. In Curth et al. (2024), we show that this is different for e.g. interpolating random forests, which act like 1-NN estimators on the training data, but can act like k-NN estimators with k > 1 on previously unobserved inputs. Wyner et al. (2017) call this behavior spiked-smooth, which provides a good metaphor: benignly interpolating models have the capacity to create sharp regions around training examples ('spike') where one may wish to retain precise knowledge of the known label but are much *smoother* in regions of the input space where no information has been observed at training time and the problem is hence underdetermined.

In recent work, we demonstrated that such a difference in behavior at train- and test-time is indeed quantifiable *with-out access to test-time labels* by measuring the effective parameters (Hastie & Tibshirani, 1990, Ch. 3.5) a model uses when issuing predictions. While such complexity mea-

sures were originally developed in the fixed design context and are therefore usually only computed for train-time predictions, we showed in Curth et al. (2023) that they can be adapted to the random design context and be computed for train and test-examples separately. We then showed that this can be used to predict when interpolation is benign in linear regression (Curth et al., 2023), random forests (Curth et al., 2024) and even neural networks (Jeffares et al., 2024), as in all cases benignly interpolating models use substantially less effective parameters when issuing predictions for new test examples than for train-time predictions. (To intuitively see why this may improve generalization performance, recall from Sec. 3 that lowering model complexity can lead to a reduction in *both bias and variance* for new test inputs.)

6. Conclusion

In this note, we highlighted that one seldomly discussed yet immensely important factor in the emergence of modern (apparently counterintuitive) ML phenomena is that model performance today is evaluated in terms of generalization to new inputs, while the classical statistics literature, in which many of the intuitions regarding bias-variance tradeoffs and overfitting were developed, often considered in-sample prediction error (where only noisy outcomes are resampled but input points are the same as during training). We showed that the move from fixed to random designs changes the classical bias-variance tradeoff even in the underparameterized regime for simple k-NN estimators, highlighting that behaviors that would appear to contradict classical statistical textbook intuitions can arise even in the absence of high-dimensional data, modern ML estimators and overparameterization – factors that are usually held responsible for counterintuitive ML phenomena. We then demonstrated that this is another reason for the historical lack of observations of phenomena like double descent and benign overfitting: when fixed design prediction is of interest – as was the case historically in statistics –, it is impossible to observe such behavior.

Implications. Returning to the opening question "Should we no longer be concerned about overfitting and biasvariance tradeoffs?" the answer is thus "It depends!". It depends on the setting that is of interest in practice, and *how* the used method interpolates the data (if it does). If generalization to new inputs is of interest – as is the predominant setting in the modern ML literature – and if models interpolate the training data in ways that are likely to be benign because predictions are smoother on test- than on training points – as appears to be the case for e.g. neural networks and random forests – then worrying about restricting the model's ability to perfectly fit the training data may no longer be necessary.

If, however, training inputs are likely to reoccur at test-time then overfitting should be a concern. This would be the case not only in classical fixed design settings where input points are somehow designed, but also when observed inputs are coarser than the latent variables that determine outcomes in the true underlying DGP - e.g. when continuous characteristics are dichotomized during recording so that individuals with different underlying characteristics get mapped to the same input x. Another context in which training inputs can reoccur at (a different type of) test-time is in causal inference, where ML is sometimes used to impute nuisance functions that are then further processed in downstream analysis steps and therein evaluated at the training points (Van der Laan et al., 2011; Chernozhukov et al., 2018). There, sample-splitting is regularly used to forgo potential bias due to overfitting – a practice which should indeed continue despite observations of benign interpolation in out-of-sample context.

Finally, the contents of this note showcase that introductory textbooks and courses on statistical learning could potentially benefit from a makeover of their sections on the bias-variance tradeoff and overfitting, in particular, by being *more precise about the settings* in which different intuitions are likely to apply (and why). Beyond questions relating to bias and variance, it would also be interesting and important to investigate whether the move from fixed to random designs affects any further fundamental statistical intuitions taught to statistics students around the world.

Acknowledgements

I would like to thank Alan Jeffares for countless thoughtprovoking discussions on the topic and for helpful comments on earlier versions of this note.

Appendix

A. Bibliographical notes

The primary purpose of this note is to be *pedagogical*; it is hence written with only limited references to tangentially related work. Below, we briefly expand on some work that is related yet different from the topic of this note.

The investigation contained in this note relates to and was partially inspired by Rosset & Tibshirani (2019), who show that excess bias and variance terms appear in the biasvariance decomposition of the expected prediction error when comparing random to fixed designs and provide precise characterisations of these terms for the linear regression setting. (They do not, however, discuss implications for the bias-variance tradeoff and behavior of error as a function of model complexity or interpolation as we do here).

With a similar goal of disentangling methodological com-

plexity of modern ML methods from modern ML phenomena, Belkin et al. (2018b) show that benign interpolation is not unique to modern deep learning methods – they show theoretically and empirically that benign interpolation appears in kernel methods too. Similarly, Belkin et al. (2018a) study generalization properties of a specific interpolating weighted nearest neighbor rule. Belkin et al. (2019) showed that double descent occurs not only in deep neural networks were they were first observed (Bös & Opper, 1996), but also in other more classical ML methods like linear regression. Refer to Loog et al. (2020) for a brief historical note on observations of double descent prior to Belkin et al. (2019).

Neal et al. (2018) and Neal (2019) also note that textbooks may require an update regarding the bias-variance tradeoff as they empirically discover a lack of such tradeoff in deep neural networks, but do not link this to differences between in- and out-of-sample prediction. Instead, they highlight that in neural network training there are additional sources of variance beyond sampling noise in outcomes. Adlam & Pennington (2020) also revisit the bias-variance decomposition to better understand double descent. They provide a more fine-grained decomposition of the prediction error that takes into account all sources of randomness in modern ML, and show that this helps to understand deep double descent. (For the simple k-NN and linear regression methods that we consider here, this is not necessary as there are no such additional sources of randomness).

Mallinar et al. (2022) provide a taxonomy and theoretical analysis of different classes of interpolating models, distinguishing between "benign", "tempered" and "catastrophic" behavior. There is a rich theoretical literature determining precise conditions when interpolation can be benign (for generalization error) in linear regression, see e.g. Bartlett et al. (2020); Muthukumar et al. (2020); Hastie et al. (2022).

B. Empirically decomposing the bias term

Here, we empirically investigate whether the terms $\operatorname{NeighborMatchingBias}_k(x_0)$ and $\operatorname{AveragingBias}_k(x_0)$ indeed behave as expected. As we discussed in Sec. 3.2, the degree of nonlinearity of f^* will play a role in this. Therefore, in addition to Eq. (6),we borrow a linear DGP from Hastie et al. (2017),

$$f^*(x) = \sum_{l=1}^{s} x_l$$
 (7)

where only the first s dimensions of $x \in \mathbb{R}^d$ (here: s=5, d=10) enter the regression specification and $x \sim \mathcal{N}(0, \Sigma)$, where the feature covariance matrix has $\Sigma_{ij} = 0.35^{|i-j|}$.

In Fig. 5, we observe that it is indeed the neighbor matching bias component that distinguishes in- and out-of-sample prediction: at k=1, decreasing the model complexity by adding an additional neighbor worsens the in-sample

NeighborMatchingBias, but improves the out-of-sample bias. As expected, AveragingBias does not appear when the DGP is linear, but increases in k in the nonlinear case as we add neighbors relative to the 1-NN estimator. (Note that it finally appears to decrease again once k is very large; this may be due to an interplay of the DGP with the uniform distribution of inputs). Overall, for the DGPs considered here, it appears that it is the NeighborMatchingBias that dominates the total bias term (by orders of magnitude).

References

- Adlam, B. and Pennington, J. Understanding double descent requires a fine-grained bias-variance decomposition. *Advances in neural information processing systems*, 33: 11022–11032, 2020.
- Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.
- Belkin, M. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.
- Belkin, M., Hsu, D. J., and Mitra, P. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. *Advances in neural information processing systems*, 31, 2018a.
- Belkin, M., Ma, S., and Mandal, S. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pp. 541–549. PMLR, 2018b.
- Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical biasvariance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.
- Bös, S. and Opper, M. Dynamics of training. *Advances in Neural Information Processing Systems*, 9, 1996.
- Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. Double/debiased machine learning for treatment and structural parameters, 2018.
- Curth, A., Jeffares, A., and van der Schaar, M. A u-turn on double descent: Rethinking parameter counting in statistical learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Curth, A., Jeffares, A., and van der Schaar, M. Why do random forests work? understanding tree ensembles as self-regularizing adaptive smoothers. *arXiv preprint arXiv:2402.01502*, 2024.

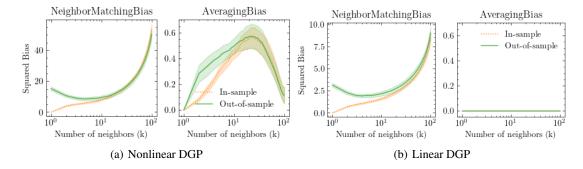


Figure 5. The bias due to lack of a perfect close neighbor match dominates the bias term out-of-sample. The behavior of the Squared NeighborMatchingBias and Squared AveragingBias by k for kNN estimators, in-sample (orange) and out-of-sample (green) for a nonlinear (left) and linear DGP (right)

- Efron, B. How biased is the apparent error rate of a prediction rule? *Journal of the American statistical Association*, 81(394):461–470, 1986.
- Efron, B. and Tibshirani, R. Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.
- Friedman, J. H. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67, 1991.
- Goodfellow, I., Bengio, Y., and Courville, A. Deep learning. MIT press, 2016.
- Hastie, T. and Tibshirani, R. Generalized additive models. *Monographs on statistics and applied probability. Chapman & Hall*, 43:335, 1990.
- Hastie, T., Tibshirani, R., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- Hastie, T., Tibshirani, R., and Tibshirani, R. J. Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*, 2017.
- Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949, 2022.
- Jeffares, A., Curth, A., and van der Schaar, M. A closer look at deep learning phenomena through a telescoping lense. In *Thirty-eighth Conference on Neural Information Processing Systems*, 2024.
- Kellogg, M., Mogstad, M., Pouliot, G. A., and Torgovitsky, A. Combining matching and synthetic control to tradeoff biases from extrapolation and interpolation. *Journal of the American statistical association*, 116(536):1804–1816, 2021.

- Loog, M., Viering, T., Mey, A., Krijthe, J. H., and Tax, D. M. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, 2020.
- Maddox, W. J., Benton, G., and Wilson, A. G. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020.
- Mallinar, N., Simon, J., Abedsoltan, A., Pandit, P., Belkin, M., and Nakkiran, P. Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting. *Advances in Neural Information Processing Systems*, 35:1182–1195, 2022.
- Murphy, K. P. *Probabilistic machine learning: an introduction*. MIT press, 2022.
- Muthukumar, V., Vodrahalli, K., Subramanian, V., and Sahai, A. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 1 (1):67–83, 2020.
- Neal, B. On the bias-variance tradeoff: Textbooks need an update. *arXiv preprint arXiv:1912.08286*, 2019.
- Neal, B., Mittal, S., Baratin, A., Tantia, V., Scicluna, M., Lacoste-Julien, S., and Mitliagkas, I. A modern take on the bias-variance tradeoff in neural networks. *arXiv* preprint arXiv:1810.08591, 2018.
- Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.
- Rosset, S. and Tibshirani, R. J. From fixed-x to random-x regression: Bias-variance decompositions, covariance penalties, and prediction error estimation. *Journal of the American Statistical Association*, 2019.
- Van der Laan, M. J., Rose, S., et al. *Targeted learning:* causal inference for observational and experimental data, volume 4. Springer, 2011.

- Wyner, A. J., Olson, M., Bleich, J., and Mease, D. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18(1):1558–1590, 2017.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.