Bias-Variance Tradeoff: A Property-Casualty Modeler's Perspective

by Joshua Brady and Donald R. Brockmeier

ABSTRACT

The concept of bias-variance tradeoff provides a mathematical basis for understanding the common modeling problem of underfitting vs. overfitting. While bias-variance tradeoff is a standard topic in machine learning discussions, the terminology and application differ from that of actuarial literature. In this paper we demystify the bias-variance decomposition by providing a detailed foundation for the theory. Basic examples, a simulation, and a connection to credibility theory are provided to help the reader gain an appreciation for the connections between the actuarial and machine learning perspectives for balancing model complexity. In addition, we extend the traditional bias-variance decomposition to the GLM deviance measure.

1. Introduction

Judging by the two popular CAS guides on GLMs, the actuarial community's thoughts on variable selection and model validation have evolved in recent years. While the goals of modeling have always been to develop good estimators of future experience and to isolate the signal amidst the noise, our prevailing methods of model-building have evolved. Anderson et al. (2007) was initially published in the early years of actuarial modeling when predictive analytics was *terra incognita* in much of the industry. It rightly emphasized fundamental considerations, such as parameter-estimate standard errors, deviance ("type III") tests, consistency testing of individual variables, and common sense.

As the actuarial community noted the developments in "statistical learning," such subjects as overfitting, data partitioning (train/test or train/test/validate), and cross-validation began to creep more and more into our seminars and literature, and were added in the most recent general GLM "guide," Goldburd, Khare, and Tevet (2016). Regardless of the method, however, considerable judgment and "art" are involved in building truly *predictive* models, as opposed to merely complex descriptive formulae. A foremost concern in this regard is avoiding overfitting.

Overfitting is a normal tendency for analytical types, and actuaries are no exception. We are naturally predisposed to examine model-trial after model-trial, sometimes straining to squeeze that last bit of intelligence from our data. Few are the modelers who, at least at some point in their career, have been immune from overfitting. While overfitting (or, in a more positive sense, optimally fitting a model) is well understood on a conceptual level, the mathematical implications of over- or underfitting may not be as well appreciated. The concept of bias-variance tradeoff can greatly enhance our understanding of the dynamics of overfitting and assist in selecting an optimal predictive model.

2. Model complexity and optimal fit

The data upon which a model is built is called the "training" data or training partition. But since we seek a model that's truly *predictive*, we are less concerned about how accurately a model estimates the training partition than how the model performs on fresh data that was *not* used in model building. Typically, in one way or another, a portion of the available data (the "test" or "holdout" partition) is set aside, not used in model training, and used strictly to assess the model's performance on *unseen* data. The model's prediction accuracy on the test partition is an estimate of its future performance after implementation. A common measurement of the overall accuracy of a model's prediction is the *mean squared error* (MSE), also known as the *prediction error*. If we denote the dependent variable in a dataset of n points as y, the covariates (collectively) as x, and our particular model estimator as $g(x)$,

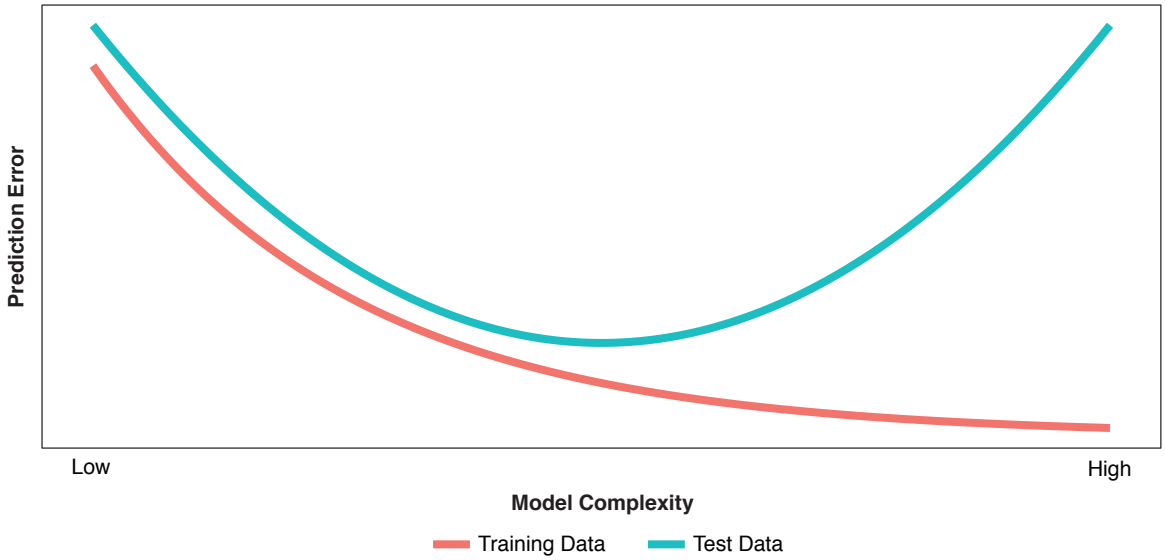
$$\text{MSE} = \frac{1}{n}\sum_i (y_i - g(x_i))^2 .$$

The MSEs of the training and test partitions are called the *training error* and *test error*, respectively.

As discussed by Hastie, Tibshirani, and Friedman (2009), improvement in the training error is a major factor in our decision to add terms or nodes to our model. So, necessarily, as we dig deeper and deeper into the training data and fit the model ever more closely to the training partition, the training error continues to fall. But as depicted in Figure 1, the same is true of the test error only to a point.

As the model becomes more complex, "overfitting" (or "overfit") begins to occur when random variations (noise) in the training partition are misinterpreted as true relationships or "regularities" (signal). An overfit model usually has too many parameters, or variables that are defined with excess complexity. As such, while it fits the training data very well (actually,

Figure 1. Training and test error vs. model complexity



too well), the overfit model "generalizes" poorly to the test partition. The model becomes too sensitive and reactive to small fluctuations in the training data. Even with the proper number of parameters, models fit using typical techniques, such as OLS regression or un-regularized GLMs, can perform poorly on new, unseen data. As characterized by Fortmann-Row (2012), there is a certain "sweet spot" in model complexity between underfit and overfit where the model's performance on unseen data is optimized.

So what are we really trying to do when we model? Despite our mindfulness of the need to "separate signal from noise," it might be easy to think that the goal of modeling is to develop an estimator for a set of observed data. It is not. Our goal as modelers is to develop an estimator *for the signal underlying the observed data* to the best of our ability. Our techniques, in one fashion or another, are founded upon the idea of optimizing some loss function, such as minimizing squared error or maximizing likelihood. But given a particular dataset (that is, a particular instance of all possible datasets), minimizing squared error (in the strict sense) might produce an optimal fit of the particular training partition, but not necessarily an optimal fit of the underlying signal (predictiveness).

Test error is a better estimate of predictiveness than training error, and training error is a poor estimate of test error. So even though some of the principal diagnostics used during the model-building process (parameter-estimate standard errors, deviance tests, etc.) are driven by our training partition, we need to focus on the *test* error in assessing predictiveness.

3. Irreducible and explainable test error

For a given point in the *test* partition, our observable total prediction error (the error we can see) is $y - g(x)$. In seeking to better understand the sources of this error, it can be expressed as the sum of two sources: *irreducible error* and *explainable error*.

It helps at this point to postulate the existence of a true function, $f(x)$[1], that underlies (or generates)

[1]While it might be more conventional to denote the model $g(x)$, an estimator of $f(x)$, as $\hat{f}(x)$, a distinct letter was chosen for the notation in this paper to foster easier visual distinction between the true function and estimator.

the dependent variable in the observed data. We can express the prediction error as:

$$y - g(x) = [y - f(x)] + [f(x) - g(x)]$$

The first component, $y - f(x)$, is *irreducible error*: the difference between the observed target value and the true functional value of that point. This error is due to randomness intrinsic to the phenomenon itself. It is the natural error that is out of reach and that cannot be explained by any model.

The second component is the *explainable error*: the difference between the true functional value and the value estimated by our particular model for the particular point in question. This error reflects the fact that our limited training data does not fully represent all possible datasets, and it reveals the inability to produce an optimal estimator given limited data. Presumably, with infinite data and ideal estimation techniques, the explainable error could be eliminated.

This distinction between irreducible error and explainable error helps us focus on the real goal of modeling: to develop an estimator for the signal . . . the regularities or the true form . . . underlying the observable data. From this perspective there are three facts of life for modelers to accept and manage:

1. There is no such thing as infinite data. Any dataset of any size falls short of representing the totality of the true signal. We are always working with limited data, and our model needs to operate on data *unseen*.
2. No modeling technique (or ensemble of techniques) is perfect. There will always be a gap between the best estimator we can fashion and the true function.[2]
3. Some error is irreducible. A portion of our prediction error not only cannot be explained by our model, but we must take steps to *avoid* trying to capture it in our model to avoid overfitting.

4. Expected value in the context of bias-variance tradeoff

Before we dive into a detailed discussion of bias and variance, we first lay the groundwork for exactly what is meant by expectation in the context of bias-variance tradeoff. In practice we are given a single training sample (X_1, Y_1) on which to fit a model g. Perhaps in an alternate reality we may have been given a different training sample (X_2, Y_2) on which to train g. Let $\mathcal{F}$ represent the entire sample space, that is, the set of all possible training samples. We often denote the expectation of $g(x)$ as $\mathbb{E}_{\mathcal{F}}[g(x)]$ to emphasize that the expectation of $g(x)$ is over the entire sampling space $\mathcal{F}$. This is a subtle, but crucial, notion for the proper appreciation of the expected value of the model estimate in understanding bias-variance tradeoff. At certain points in this discussion, this subscript is omitted for simplicity.

Ultimately, our goal is to fit a model that will perform best on unseen data. Let the space (X, Y) represent the holdout set on which to test the performance of our model. We assume that there is some true relationship $f(x) = \mathbb{E}[y|x]$ between the target variable and covariates. It is assumed that our holdout sample (X, Y) is independent of our sampling space $\mathcal{F}$. In practice, this independence assumption often does not hold, although violation rarely results in poor-performing models.

Suppose we have some loss function, $L(g(x), y)$, measuring the error of the model at a point (x, y). We would like to know the expected error over all training sets and over the distribution of the test point (x, y). That is, when we write $\mathbb{E}[L(g(x), y)]$ the expectation is implicitly over the space of sample sets $\mathcal{F}$ on which the model g is fitted and over the

[2] In fact, even the true predictors are unknown. The predictors within our grasp are most likely correlates, albeit useful correlates, (or correlates of correlates of correlates . . .) of any true predictive variable. This becomes pertinent (sometimes painfully so) when we're tasked with explaining to a client or product manager the appropriateness of (what are perceived as) mysterious and arcane variables in our models, such as ratios derived from census data that are associated with a policy by the policy's zip code or census tract.

distribution of y given x. Specifically we can express the expectation as

$$\mathbb{E}[L(g(x), y)] = \mathbb{E}_{\mathcal{F}}[\mathbb{E}_{y|x}[L(g(x), y)|x]]$$
$$= \mathbb{E}_{y|x}[\mathbb{E}_{\mathcal{F}}[L(g(x), y)|x]]$$

With the above framework, we now define bias and variance for a model as follows:

Definition 4.1. *(Bias) Let $\mathcal{F} = \{(X, Y)\}$ be the space of training sets for fitting model g. The bias associated with a model g at a test point (x, y) is given by*

$$\text{Bias}_{f(x)}[g(x)] = \mathbb{E}[g(x) - y|x] = \mathbb{E}_{\mathcal{F}}[g(x)] - \mathbb{E}[y|x]$$
$$= \mathbb{E}_{\mathcal{F}}[g(x)] - f(x),$$

where $f(x) = \mathbb{E}[y|x]$.

Definition 4.2. *(Variance) The variance of a model g at a point x is given by*

$$\mathbb{E}\left[(g(x) - \mathbb{E}[g(x)])^2\right] = \mathbb{E}_{\mathcal{F}}\left[(g(x) - \mathbb{E}_{\mathcal{F}}[g(x)])^2\right].$$

5. The expected value of the model estimate, bias, and variance

Any *particular* model under consideration is but one manifestation of that model (or specifically, one manifestation of that functional form or algorithm), based on the data upon which it was trained, which as we've seen is one particular instance, or sample, of training data among the multitude of possible sample training datasets in our sampling space $\mathcal{F}$. As such, there is a distribution of possible model estimates. Accordingly, for any particular observed point, (x, y), we can speak of the expected value of $g(x)$, or $\mathbb{E}[g(x)]$. $\mathbb{E}[g(x)]$, then, is the *expected prediction* of the model (or expected value of the "estimator") for observed point (x, y).

Like the *true* prediction, $f(x)$, the expected prediction of the model, $\mathbb{E}[g(x)]$, is of a speculative, or conjectural, nature; it cannot be directly measured. However, if multiple training datasets and model parameterizations could be amassed, one could approximate $\mathbb{E}[g(x)]$.

The explainable error, $f(x) - g(x)$, can now be expressed as the sum of two components that are particularly relevant to our deliberation:

$$f(x) - g(x) = (f(x) - \mathbb{E}[g(x)]) + (\mathbb{E}[g(x)] - g(x))$$

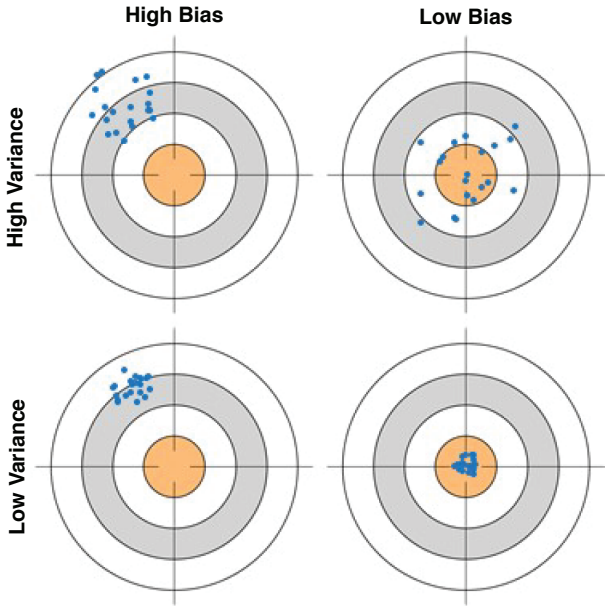
As noted in Definition 4.1, the negative of the first component, $\mathbb{E}[g(x)] - f(x)$, the difference between the *expected* prediction of the model and the value we are trying to predict (i.e., the true functional value), is the bias at point (x, y). Bias quantifies the error of the model (the algorithm, estimator or functional form, in a general sense) prediction from the true value. Consistent with Definition 4.2, the square of the negative of the second term, $(g(x) - \mathbb{E}[g(x)])^2$, is the point's contribution to the model's variance.

It is important to note that the exact meaning of the term "bias" in this context is different than its classical definition. In the classical context, a statistic, or estimator, is said to be "unbiased" if it equals a population parameter. In a modeling or machine-learning context, bias refers to the difference between the expected prediction of a particular model and the point value it is intended to predict. The latter definition is presumed in this paper unless otherwise noted.

As illustrated by Hastie, Tibshirani, and Friedman (2009) and Fortmann-Row (2012), the bullseye diagram in Figure 2 helps clarify the preceding explanations of bias and variance.

The center of the target represents perfect prediction of the true value at a test point (x, y), while the portions of the target away from its center represent predictions with error. Each point represents one manifestation of the estimator, or model, given its particular training dataset. The extent to which the various points cluster tightly represents the variance of the estimator. The extent to which the center of the point cluster approximates the center of the target represents the bias of the estimator. The top-left bullseye shows a scatter of points with both

Figure 2. Visual depiction of bias and variance



high bias and high variance: they tend to be at a distance from the true value and are broadly dispersed. The bottom-left bullseye similarly depicts a model with high bias at this value of x, but having low variance among the various possible model estimates. The bullseyes on the right illustrate low bias, where the points cluster around a center that accurately predicts the true value. The bottom-right bullseye exemplifies the modeler's goal: an estimator algorithm that accurately predicts its true value with high reliability.

6. Decomposition of the expected squared prediction error

The preceding division of prediction error into irreducible error and explainable error, and the further division of explainable error into bias and variance components, assist us in better understanding the components of MSE on unseen (test) data. As derived in Appendix A, test prediction error can be decomposed as follows:

$$\mathbb{E}[(y-g(x))^2] = \mathbb{E}[(y-f(x))^2] + (\mathbb{E}[g(x)] - f(x))^2 + \mathbb{E}[(g(x)-\mathbb{E}[g(x)])^2]$$

The components of the above equation are recognizable, as the total test prediction error $\mathbb{E}[(y-g(x))^2]$ is the sum of . . .

Irreducible squared error: $\mathbb{E}[(y-f(x))^2]$
Squared bias: $(\mathbb{E}[g(x)] - f(x))^2$
and the variance of the estimator: $\mathbb{E}[(g(x)-\mathbb{E}[g(x)])^2]$

As noted, while the first component, irreducible error (random noise in the observed data with respect to its true value) is beyond our control and reach, the two components of explainable error can be manipulated to achieve an optimal model.

In terms of model complexity, underfit models tend to have high bias and low variance, while overfit models tend to have low bias with high variance (low reliability in the face of new data). When starting with a moderately overfit model, the "sweet spot" of optimal predictiveness can be obtained by reducing model complexity, thereby enhancing reliability (reducing variance) at the cost of greater bias: the so-called "bias-variance tradeoff."

It's interesting at this point to reflect on how this concept of *inviting* bias into our model might be met with resistance, especially by actuaries who first learned statistics before the machine-learning era. Many statistics courses emphasized the benefits of *unbiased* estimators (in the classical sense of the term) to the virtual exclusion of all others, and essentially rewarded the quest for a tighter and tighter fit of the data. In essence, while this education did a good job training us to produce high-quality *descriptive* statistics, it may have failed to anticipate today's greater appreciation of *predictive* analysis.

To illustrate these ideas we consider two simple models, a fully-parameterized model and an intercept model. For the fully-parameterized model we assume there is a single covariate labeled x with no other differentiating information considered by the model. In the case of multiple covariates, we can form the Cartesian product of all available covariates to arrive at a fully-parameterized model that considers all available information. We assume that it is possible (but not required) for each unique value of x to

be sampled multiple times. The model is simply the average of the observations over each level of the covariate. For example, suppose our single covariate is whether an insured has been claims-free in the past three years. That is, $x \in \{\text{Yes, No}\}$. In this simple case we presume to have multiple observations for both of the values "Yes" and "No." The fitted model for "Yes" would be the average of observations for the claims-free insureds.

Example 6.1 *(Fully-Parameterized Model) The fully-parameterized model is defined as* $g(x) = avg(y|x)$. *That is, our prediction at x is simply the average of y over x on the training set. For each value of x we have a unique prediction.*[3]

Bias:

Clearly this estimate is unbiased at the granularity of covariate x as $\mathbb{E}_{\mathcal{F}}[g(x)] = \mathbb{E}_{\mathcal{F}}[\text{avg}(y|x)] = avg(\mathbb{E}_{\mathcal{F}}[y|x]) = avg(\mathbb{E}_{y|x}[y|x]) = \mathbb{E}[y|x]$. The key idea is that $\mathbb{E}_{\mathcal{F}}[y|x] = \mathbb{E}_{y|x}[y|x]$. That is, the expected value of the average of $y|x$ over $\mathcal{F}$ is the expected value of $y|x$.

Variance:

As the model has a separate parameter for each value of x, the resulting variance at a particular x is $\mathbb{E}[(\mathbb{E}[g(x)] - g(x))^2] = \mathbb{E}[(\mathbb{E}[y|x] - \text{avg}(y|x))^2]$. This is simply the variance of $avg(y|x)$ over $\mathcal{F}$. When the covariate x has many values, we would generally expect the variance to be very high. An example of such a model would be the direct use of zip code as the covariate in a territorial model. While the bias would be zero at the zip code level, the direct use of zip code would result in very high variance.

Example 6.2 *(Intercept Model) The other simple model is the intercept model. That is,* $g(x) = \bar{y}$ *over the entire sample set.*

Bias:

The bias for the intercept model is given by

$$\mathbb{E}_{\mathcal{F}}[g(x)] - \mathbb{E}[y|x] = \mathbb{E}_{\mathcal{F}}[\bar{y}] - \mathbb{E}[y|x],$$

which clearly is biased except when $\mathbb{E}_{\mathcal{F}}[\bar{y}] = \mathbb{E}[y|x]$.

Variance:

The variance of the intercept model can be expressed as follows:

$$\mathbb{E}\left[(\mathbb{E}[g(x)] - g(x))^2\right] = \mathbb{E}\left[(\mathbb{E}[y] - \bar{y})^2\right]$$

This is the variance of the population mean, which is relatively stable except on small data sets. Of course a constant model $g(x) = c$ would have zero variance, but this is a less interesting model, as it has no dependence on the data.

As an illustration of these two extreme models, consider a simple situation in which $f(x) = 2x$, the set of x values consists of the first five positive integers, and we have three training samples, each consisting of three randomly-generated y values for each value of x.

Table 1 shows the squared bias and the sample variance for each unique value of x for each of the three samples. The sample variance, given by $(g(x) - \mathbb{E}[g(x)])^2$, is used to illustrate the contributions to variance by each model.

As noted above, for the fully parameterized model $\mathbb{E}[g(x)] = \mathbb{E}[y|x]$ which equals $f(x)$. For the intercept model, each sample's $g(x) = \bar{y}$, and $\mathbb{E}[g(x)] = \mathbb{E}_{\mathcal{F}}[\bar{y}]$. This illustrates that the fully parameterized model is unbiased, as compared to the considerable bias of the intercept model. The variance, on the other hand, is substantially lower for the simpler intercept

[3]Importantly the fully-parameterized model is different from what is referred to as a saturated model. With a saturated model each observation is assigned the value of the observed target value. In fact, multiple observations corresponding to the same value of covariate x generally will not have the same observed value y; thus, the saturated model cannot in general be used to make predictions on unseen data. Saturated models do have their use for calculating a likelihood-based deviance measure of model fit. In this case, a saturated model is "evaluated" on whichever set is being used for goodness-of-fit evaluation.

Table 1. Illustration of Examples 6.1 and 6.2

Training Sample	x	f(x) = 2x	y1	y2	y3	Fully-Parameterized Model				Intercept Model			
						g(x)	E[g(x)]	Sq. Bias	Variance	g(x)	E[g(x)]	Sq. Bias	Variance
1	1	2	0.8	2.8	3.2	2.3	2	0	0.090	6.5	6	16	0.230
	2	4	3.9	5.9	3.8	4.5	4	0	0.250	6.5	6	4	0.230
	3	6	5.1	4.9	7.1	5.7	6	0	0.090	6.5	6	0	0.230
	4	8	10.2	8.3	10.3	9.6	8	0	2.560	6.5	6	4	0.230
	5	10	11.3	10.2	9.4	10.3	10	0	0.090	6.5	6	16	0.230
2	1	2	2.1	2.7	0.7	1.8	2	0	0.028	5.9	6	16	0.018
	2	4	4.5	3.6	5.3	4.5	4	0	0.218	5.9	6	4	0.018
	3	6	6.8	4.0	6.7	5.8	6	0	0.028	5.9	6	0	0.018
	4	8	8.0	9.3	7.9	8.4	8	0	0.160	5.9	6	4	0.018
	5	10	7.4	8.4	10.6	8.8	10	0	1.440	5.9	6	16	0.018
3	1	2	1.9	–0.4	0.7	0.7	2	0	1.604	5.7	6	16	0.111
	2	4	6.3	2.5	5.0	4.6	4	0	0.360	5.7	6	4	0.111
	3	6	4.8	5.9	4.9	5.2	6	0	0.640	5.7	6	0	0.111
	4	8	5.8	7.4	8.8	7.3	8	0	0.444	5.7	6	4	0.111
	5	10	9.4	10.3	11.7	10.5	10	0	0.218	5.7	6	16	0.111
					Sample Means:			0	0.548			8	0.120

model, reflecting the stability of the sample mean. This example illustrates extremes that may be encountered between fitting an overly simple model and an over-parameterized model. As we discuss in section 9, the intercept model does not always have lower variance than a more complex "fully parameterized" model.

7. A simulation

The following simulation (patterned after the simulation in Stansbury 2013) exemplifies the dynamics of bias-variance tradeoff within a family of models of various levels of complexity.

Assume that we know the "true" function that generates our observable data, as shown in Figure 3.

Figure 4 shows a sample of points, or training data, generated from the true function ("Ftrue") with noise, as well as a series of polynomials of various orders (models) parameterized to the sample.

This allows us to visualize the decomposition of the prediction error into irreducible and explainable error. In terms of the preceding discussion, for any of the four models and for any particular x-value along the curve, the vertical difference between the sample value and the model-fit value is the prediction error, or $y - g(x)$. The difference between the observed sample value and the true function is irreducible error, $y - f(x)$, and the difference between the true functional value and the model-fit value, or $f(x) - g(x)$, is the explainable error.

In the strict sense of the term, since this illustration involves only a single sample, we cannot measure the bias, the definition of which is based on the *expected value* of the estimator and the true functional value. However, if we assume for the purpose of illustration that each of the particularly-parameterized models in Figure 4 is representative of its expected value, this exhibit illustrates the relationship between bias and model complexity. The simplest model (of order of 1, or "g1") has high bias along most of the curve, that is, it fits the true function poorly. The third-order polynomial, "g3," begins to form itself to the true function better but still has observable bias. The higher-order polynomials, "g5" and "g9," appear to fit the true function very well, exhibiting low bias for

Figure 3. Bias-variance tradeoff simulation: the "true function"

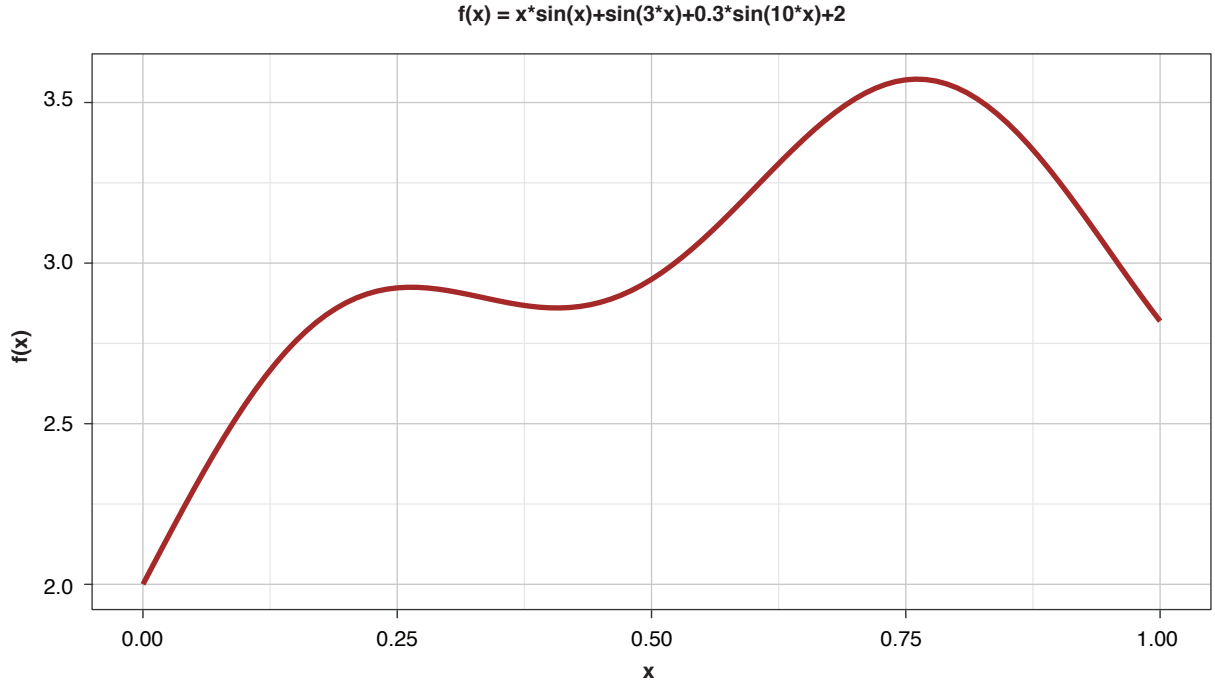
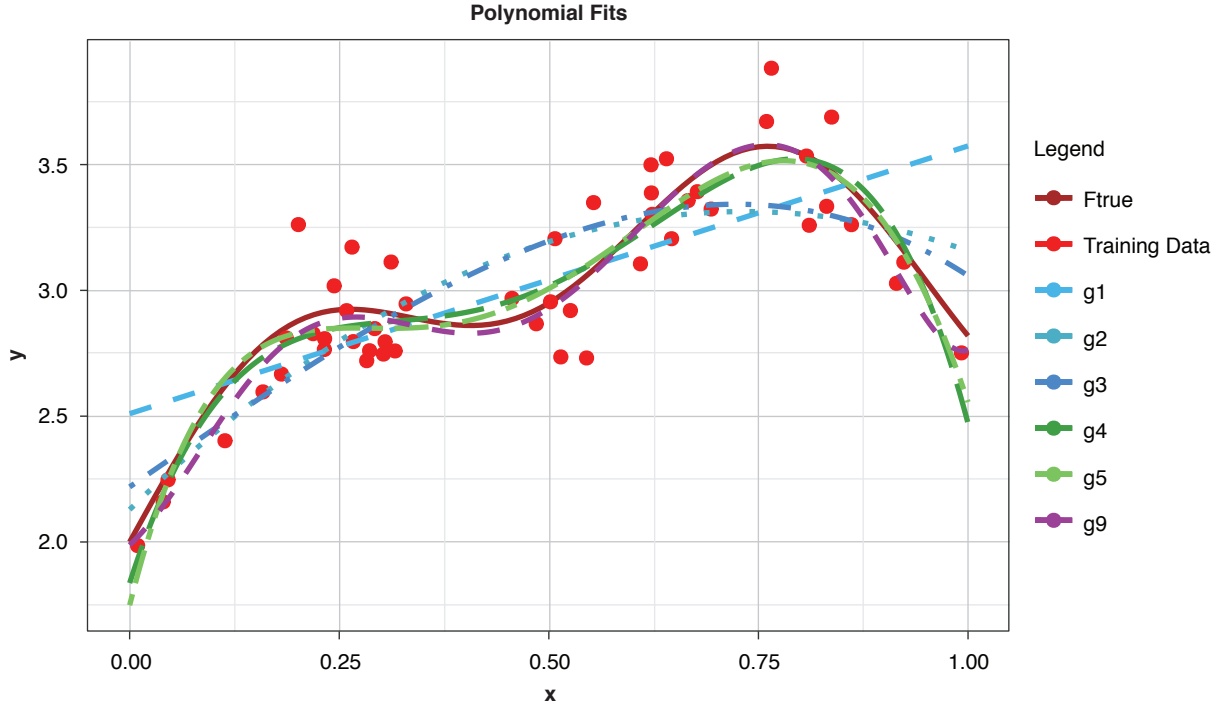


Figure 4. Bias-variance tradeoff simulation: the "true function"



the preponderance of x-values. If we're simply looking to optimize bias, we might conclude that the higher the order (the greater the complexity), the better.

But while the preceding example may demonstrate how bias varies with model complexity based on this *single sample* of training data, it gives us no insight into either bias in the strict sense, $\mathbb{E}[g(x)] - f(x)$, or the variance of the estimator, $\mathbb{E}[(g(x) - \mathbb{E}[g(x)])^2]$. To accomplish that, the simulation must be expanded to involve multiple training samples (and their corresponding test samples), allowing estimation of the expected value of the estimators.

Fifty datasets of forty points each were randomly generated, and split into training and test sets of thirty and ten points, respectively. Polynomial models of orders one through ten were fit to each training set. In the real world we're left to work with a single "instance" (a single sample), so this is like being able to peer into fifty "parallel universes."

Figure 5 demonstrates the fifty simulated instances of the simplest polynomial model with fifty thin blue lines. The thicker green line in the middle of the cluster of the fifty instances is the mean of the fifty fits, providing an estimate of the expected value of the estimator, or $\mathbb{E}[g(x)]$. While this simplest of models has high bias for most x values, the tight clustering of the individual fits demonstrates the model's low variance.

As noted previously, and as shown in Figure 6, if the model's complexity is increased slightly (to a power of two), the bias is lessened somewhat compared to the simplest model. However, the individual fits are not as tightly clustered as in Figure 5, demonstrating the higher variance of the more complex estimator.

Appendix B contains all such illustrations from polynomial orders one through ten, wherein one can observe the general decrease in bias and increase in variance as model complexity increases. Figure 7 shows the most complex model.

The approximated expected value of the estimator is practically coincident with the true function, indicating that bias has been almost eliminated by fitting the training data to an extreme. But the variance of the estimator is disturbingly large. This image portrays the changeability of an overfit model's re-parameterization on new data, such as a recent policy year.

Figure 8 summarizes the prediction error of both the training and test samples, as well as the squared bias and variance of the test data, for each of the models. The training prediction error decreases as

Figure 5. The simplest model: high bias and low variance

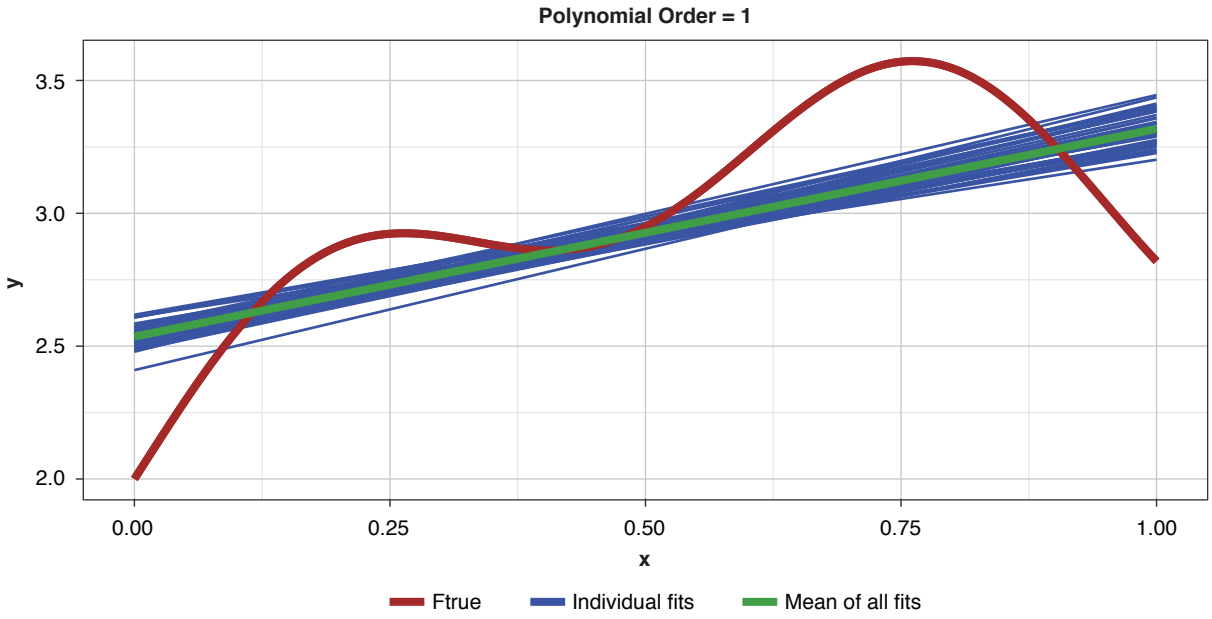


Figure 6. Slight increase in complexity: less bias, more variance

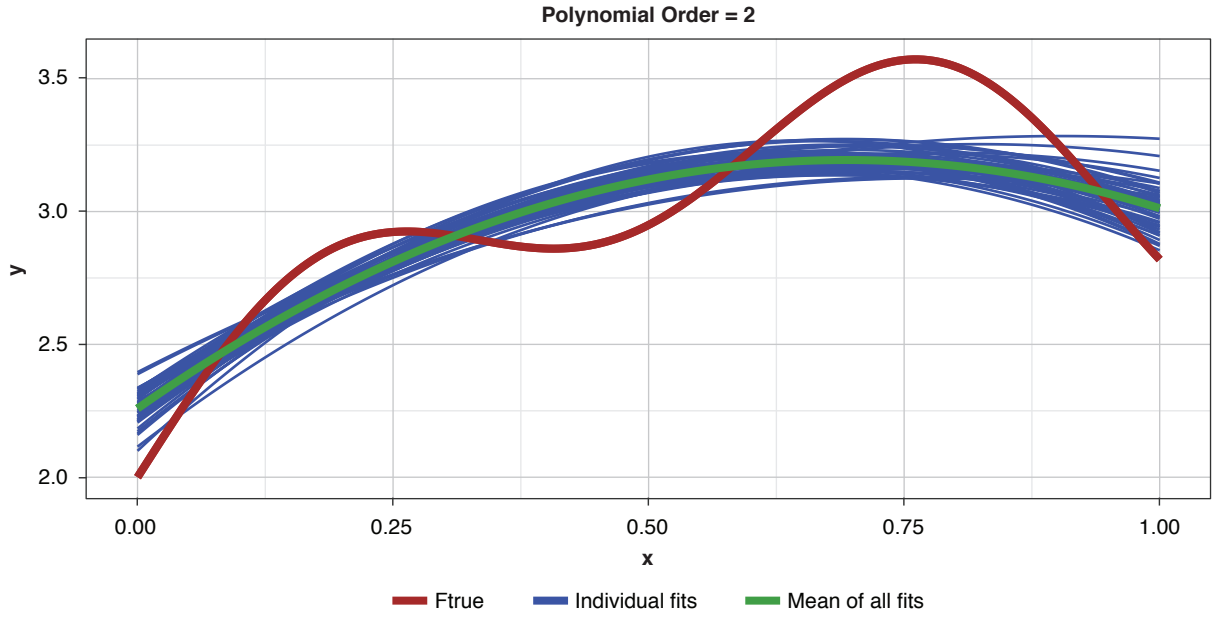


Figure 7. Most complex model: virtually no bias, high variance

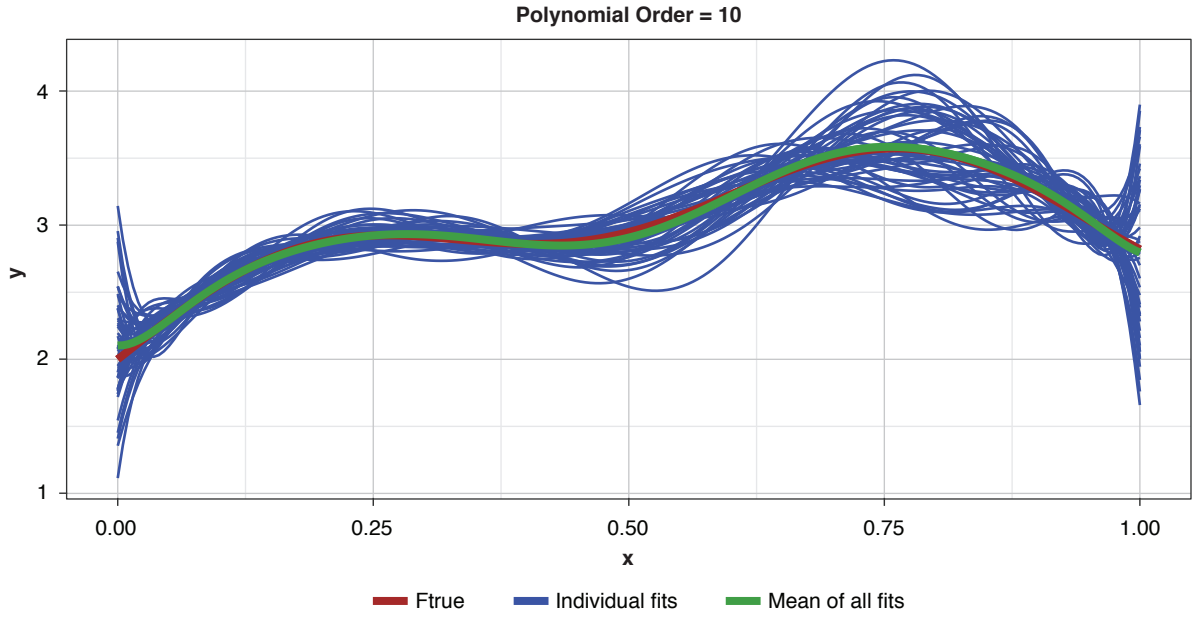
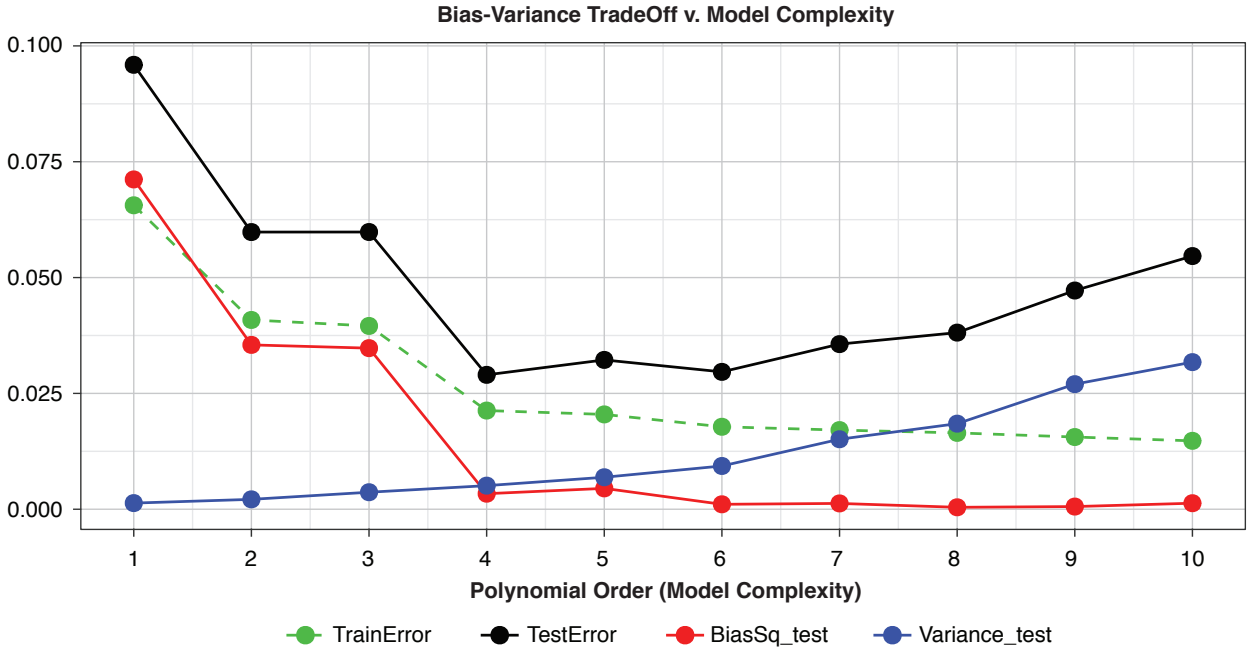


Figure 8. Prediction error, squared bias and variance by model



model complexity increases. The test prediction error (the error on unseen data), however, decreases until the polynomial order is from four to six, then steadily increases for the most complex models, suggesting that the noise in the training data is being misinterpreted as signal. The squared bias measured on the test data decreases even to the sixth order and is minimal for the higher orders. The variance of the estimator increases progressively with model complexity. As the parameterizations are influenced increasingly by noise in the training data, the estimator becomes less and less stable. The "sweet spot" appears to be around the fourth order, where bias and variance are both modest in size. This exhibit clearly shows how, if one were working with a model of higher order, simplifying the model would result in less variability in the estimator at a price of increased bias: the bias-variance tradeoff.

8. Decomposition formula for deviance

Actuaries rarely model insurance data assuming a normal distribution. Generally a distribution from the two-parameter exponential family is relied upon.

$$f(y|\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right\}$$

We do not go into detail here, but we note that θ is a function of the linear predictor. That is, we can write $\theta = \theta(\mathbf{X}\beta)$, where $\mathbf{X}$ is the design matrix for the model and β are the model coefficients. We have omitted the weights from this formulation for notational simplicity. See Ohlsson and Johansson (2010) or McCullagh and Nelder (1989) for a comprehensive treatment.

For some distributions (e.g., Poisson), the dispersion parameter ϕ is taken to be 1 and we have a single-parameter distribution. When the dispersion parameter is applicable the general form above has two parameters. In practice we often treat ϕ as fixed and known with regard to estimating the coefficients of the model. How can this be? Importantly, the maximum likelihood estimate of θ is independent of ϕ. If one wishes to perform likelihood ratio tests or estimate the covariance of the coefficients then ϕ must be estimated.

As shown in Jørgensen (1992), every distribution in the exponential family is fully determined by the

variance relationship linking the variance of y by function, v, of the mean. That is,

$$\text{Var}[y] = \phi v[\mu],$$

where $\mu = \mathbb{E}[y|x]$.

Using the variance relationship we can define the deviance at a point y with parameter μ as

$$d(y,\mu) = 2\int_{\mu}^{y} \frac{y-t}{v(t)}\text{d}t.$$

This is referred to as the unscaled deviance. The scaled deviance is given by

$$d^*(y,\mu) = \frac{1}{\phi} d(y,\mu).$$

In this section we will primarily work with the unscaled deviance. If we assume ϕ is fixed and constant then the results in this section for the unscaled deviance also hold for the scaled deviance.

The above integral-based formula for deviance is discussed in Anderson, et al. (2007). More common in the literature is the log-likelihood based definition of deviance, which is given by the difference between the saturated model and the fitted model. For our purposes the integral-based formula is more a convenient representation. In Appendix C we briefly show the equivalence between the two formulations.

The total deviance is the sum of the deviance over the data set:

$$D(\mathbf{y},\mu) = \sum d(y_i,\mu_i).$$

Deviance has the following properties:

1. $d(y, y) = 0$
2. $d(y, \mu) > 0$ for $y \neq \mu$
3. The deviance increases as μ moves away from y. That is, $d(y, \mu_2) > d(y, \mu_1)$ for $\mu_2 > \mu_1 > y$ and $\mu_2 < \mu_1 < y$.

That is, deviance is a loss function that has larger penalties as μ moves away from y. For the normal distribution the deviance is the standard squared error:

$$d(y,\mu) = (y-\mu)^2.$$

For the Poisson distribution,

$$d(y,\mu) = \mu - y + y log\left(\frac{y}{\mu}\right).$$

Before proceding, we discuss the concept of minimizing the expected deviance for model g at a point x. We refer to this is the deviance-minimizing estimator.

Definition 8.1. *The deviance-minimizing estimator, $\tilde{g}(x)$ for model g at x is defined as*

$$\tilde{g}(x) = \underset{h}{\text{argmin}}\, \mathbb{E}_{\mathcal{F}}[d(h, g(x))].$$

That is, the deviance-minimizing estimator for model g is the value, h, that minimizes the expected deviance relative to fitted values $g(x)$ over all sampled data sets $\mathcal{F}$. To help motivate this definition, recall that the deviance for the normal distribution evaluated at $(h, g(x))$ is $d(h, g(x)) = (h - g(x))^2$. Suppose we wanted to find the value of h that minimizes $\mathbb{E}_{\mathcal{F}}[(h - g(x))^2]$. As this is simply the expected squared difference between h and $g(x)$ over $\mathcal{F}$, the minimum value is given by the expected value of $g(x)$. That is, $h = \mathbb{E}_{\mathcal{F}}[g(x)]$ minimizes the expected squared difference $\mathbb{E}_{\mathcal{F}}[(h - g(x))^2]$. With this example, we see that the deviance-minimizing estimator is a generalization of the property that the mean minimizes expected squared error. In Appendix C we present examples of deviance-minimizing estimators for common distributions.

Theorem 8.2. *(Deviance Decomposition) As demonstrated in Appendix C, with the above definition of the deviance-minimizing estimator we can decompose the expected deviance for a model g at a test value (x, y) as*

$$\mathbb{E}[d(y, g(x))] = \mathbb{E}[d(y, f(x)] + d(f(x), \tilde{g}(x)) + \mathbb{E}[d(\tilde{g}(x), g(x))].$$

Our inspiration for the above decomposition comes from Hansen and Heskes (2000). As compared to Hansen and Heskes (2000), we provide a further refinement of the sum of bias and irreducible error. Additionally we provide a more detailed derivation (Appendix C). The first term is the expected deviance between the observed target value and the true functional value at that point. This is analogous to irreducible error in the MSE bias-variance decomposition. The second term is the deviance of the true functional value relative to the deviance-minimizing estimator, which is a measure of bias. The third term is the expected value of the deviance between the fitted model and the deviance-minimizing estimator of the model, which is a generalization of variance.

The deviance decomposition theorem shows that the principle of bias-variance tradeoff applies, not just for MSE, but in a more generalized setting where deviance is used to evaluate goodness-of-fit.

9. Example illustrating the connection to credibility

Credibility theory, as developed by Bühlmann, introduces a biased estimator that is a blend between the raw estimate and the population average. This blend is chosen such that the estimator has the best expected performance on new data. Using a simple example below, we illustrate the connection between Bühlmann credibility and bias-variance tradeoff.

Let x be a covariate with two levels {u,d} where the target y is normally distributed with mean plus or minus a. Specifically we assume:

$$\mathbb{E}[y|x=u]=a$$

$$\mathbb{E}[y|x=d]=-a$$

As u and d are equally likely, we assume the conditional variance is

$$\mathrm{Var}[y|x]=\sigma^2.$$

As a consequence the unconditional variance of y is

$$\mathrm{Var}[y]=\sigma^2+a^2.$$

This structure may look familiar. In credibility theory, σ^2 is the within variance and a^2 is the between variance.

Before moving further with the connection to credibility, let us consider the two simple models from Examples 6.1 and 6.2. For the fully-parameterized model, suppose we estimate $y|x$ by

$$\hat{y}(x)=g_1(x)=avg(y|x)=\bar{y}_x,$$

that is, the average of the sample for each of the two levels of x. For the intercept model, $y|x$ is estimated by the population average without regard to the levels of x. That is, $\hat{y}(x)=g_2(x)=\bar{y}$.

The expected MSE is calculated for each of the models. As mentioned above, we assume both levels of x are sampled equally on each training set of size n. The MSE of the fully parameterized model is

$$\mathbb{E}[MSE(g_1(x))]=\mathbb{E}[(y-g_1(x))^2]=\sigma^2\left(1+\frac{2}{n}\right).$$

Similarly for the intercept model g_2

$$\mathbb{E}[MSE(g_2(x))]=(\sigma^2+a^2)\left(1+\frac{1}{n}\right).$$

Suppose we had to choose just between these two options. Assuming the covariate has true signal we would generally assume that parameterized models would outperform an intercept model, but are there times when we would prefer the intercept model g_2 over the fully parameterized model g_1? Focusing on the test partition, we consider the conditions under which:

$$\mathbb{E}[MSE(g_2(x))]<\mathbb{E}[MSE(g_1(x))]$$

$$\Leftrightarrow(\sigma^2+a^2)\left(1+\frac{1}{n}\right)<\sigma^2\left(1+\frac{2}{n}\right).$$

Solving for n, we find:

$$n<\frac{\sigma^2}{a^2}-1.$$

We recognize the familiar Bühlmanns $k = \sigma^2/a^2$. It may seem surprising that there are any cases for which the intercept model gives superior test performance over the fully parameterized model. For example, if $\sigma^2 = 10a^2$, then we would need at least 10 observations in order to prefer $\bar{y}_x$ to the entire sample average $\bar{y}$.

To understand this better, we consider the traditional credibility weighting between the two estimates.

$$\hat{y} = g_Z(x) = \bar{y}_x Z + (1 - Z)\bar{y}$$

As shown in Appendix D, the expected MSE can be derived as

$$\mathbb{E}[MSE(g_Z)] = \sigma^2 + a^2(1 - Z)^2 + \frac{1}{n}\left(2\sigma^2 Z + (\sigma^2 + a^2)(1 - Z)^2\right)$$

Notice that the MSE has been written in the form of the bias-variance decomposition, where

- Irreducible error = σ^2
- Squared Bias = $a^2(1 - Z)^2$
- Variance = $\frac{1}{n}(2\sigma^2 Z + (\sigma^2 + a^2)(1 - Z)^2)$

For the fully parameterized model ($Z = 1$) the estimator is unbiased with variance $2\sigma^2/n$. For the intercept model ($Z = 0$) the estimator is biased with:

- Squared Bias = a^2
- Variance = $\frac{1}{n}(\sigma^2 + a^2)$

Generally we prefer unbiased models over biased models, but we must also consider the contribution to the total test error by the variance. While $\bar{y}_x$ may be an unbiased model, it is possible for $\bar{y}_x$ to have higher variance than $\bar{y}$. In fact, that is the case when $a^2 < \sigma^2$. The idea is that when there is a large amount of noise, reducing variance is more important than eliminating bias.

Naturally, the next question to ask is for what Z is the MSE minimized? Taking the derivative of $\mathbb{E}[\text{MSE}(g_Z)]$ and setting equal to 0, we recover a familiar Bühmann-style credibility formula.

$$Z = \frac{n+1}{n+1+\frac{\sigma^2}{a^2}}$$

The key point is that credibility in the above context attempts to find the optimal balance between the error contributed by bias and variance. If we had an infinite sample on which to build our model, then the unbiased model would perform best out of sample. Of course, we are restricted to finite samples that are often much smaller than we would prefer.

10. Practical application

Here we briefly offer some advice on how to apply the concept of bias-variance tradeoff in practice. A complete treatment of these topics would fill volumes, among which there are already excellent resources, such as Goldburd, Khare, and Tevet (2016), James et al. (2013), and Hastie, Tibshirani, and Friedman (2009).

Given the focus of this paper, it may be surprising that it is unlikely that one would even attempt to quantify bias or variance in practice. Decomposing the expected test error into irreducible error, bias, and variance is generally not possible or even desirable. The modeler usually has little knowledge of the true population being sampled, and so it is not possible to ascertain either the expected value of the model estimate nor the true expected target value.

Our goal is to develop a model that has the best predictive power on unseen data. That is, we are concerned with minimizing the out-of-sample test error. How much the irreducible error, bias, and variance contribute to the total error is not important. Why, then, should we be concerned with the bias-variance tradeoff? Most important is that familiarity with the dynamics of bias-variance tradeoff helps build the modeler's intuition regarding the tradeoffs associated with model complexity. This intuition is further developed by considering illustrations such as the simulation in Section 7 and the credibility example in Section 9. These examples help build a modeler's understanding of the techniques and decisions that produce better models.

A single holdout dataset is often employed to help optimize model performance on unseen data, but there are concerns with this approach. If a modeler tests multiple models on a single holdout set, he or she may begin to overfit the holdout data. Furthermore, reserving part of the data for testing/validation reduces the amount of data available for model training, possibly resulting in fewer ascertained patterns and less confidence in the selected structure.

As an alternative to a single holdout set, cross-validation is considered to be a standard method for assessing holdout performance (Hastie, Tibshirani, and Friedman 2009). Briefly, in n-fold cross validation the training set is divided into n parts of approximately equal size. For the ith part the model is fit to the remaining $n-1$ parts collectively, and the resulting prediction error is computed on the ith part, which serves as a partial holdout set. The n partial-prediction-error totals are then aggregated to estimate the total test prediction error. Cross-validation has the important advantage over traditional holdout sets in that all the data is used to build and test the model. In an extension of this technique called *k-times n-fold cross-validation*, the cross-validation procedure is repeated k times using a different random partitioning each time. Within this framework we are simply focused on reducing the MSE as measured through cross-validation. The error contributed by noise, bias, or variance is still unknown.

When parameter estimation is discussed, it is traditionally suggested that it is preferable for estimates to have zero bias. But as noted above, bias is not the only contributor to test error. Specifically, this means bias is not necessarily something to be avoided, as long as the reduction in variance is greater than the increase in bias.

As suggested in Section 2, models with high complexity tend to overfit, and models with low complexity tend to underfit. The following considerations may assist the modeler in finding the "sweet spot" of an optimal fit:

1. Variable Selection: Adding parameters to a model generally increases model complexity, usually reducing the bias while increasing variance. In this category we include such considerations as which covariates enter the model, grouping levels of categorical variables, introducing polynomial powers, binning continuous variables, and introducing interaction terms. As described in Anderson et al. (2007), consistency-testing individual covariates can assist in eliminating those with the most unstable relationships to the dependent variable. This is accomplished by examining each potential covariate's interaction with a "time" variable, or with some other categorical variable that provides a meaningful partition of the book of business.
2. Credibility: Once a covariate is selected for inclusion in a model, it may be determined that the "raw" estimated coefficient is not optimal. Complexity can be adjusted further by choosing a balance between the coefficients fitted by least-squares regression vs. not including the covariate. Aside from traditional credibility approaches, models that implement a coefficient-blending approach are mixed-effects models as discussed in Frees, Derrig, and Meyers (2014) and elastic-net models as discussed in James, Witten, Hastie, and Tibshirani (2013) and Hastie, Tibshirani, and Friedman (2009). From a model-complexity perspective, reducing coefficient credibility correspondingly reduces model complexity. At first glance this may seem odd, as there are still the same number of parameters in the model (i.e., traditional degrees of freedom). But by constraining the "freedom" of the parameters, partial credibility effectively reduces complexity.
3. Bagging (or, bootstrap aggregation): Essentially, the model is built multiple times on subsamples of the dataset. A popular application of bagging is with Random Forests, in which many trees are fit and averaged. Generally, while fitting a single full tree greatly overfits the data, the variance associated with a single tree can be reduced by averaging many trees, where each tree is fit on a subset of the data (sample bagging) and a subset of covariates (feature bagging). By tuning parameters such as the number of trees, the

subsampling percentage of the observations, and the subsamples of covariates, the modeler can control the balance between bias and variance. Interestingly, fitting a greater number of trees actually reduces model complexity. This may seem counterintuitive, since fitting a greater number of trees sounds like we are making the model more complicated. In fact, we are reducing the complexity as the model is more constrained. That is, with more trees being averaged, the model is less able to overfit the data.

4. Boosting: Boosting attacks the ensemble problem from a different perspective as compared to bagging. Instead of taking the average of a large number of fitted models, a large number of "weak learners" are employed serially, each one working off of the residuals of the previous iteration. While each iteration does not learn the data very strongly, the combination of the weak learners may result in a strong learner. The modeler can tune parameters such as the learning rate, the number of iterations, and the subsampling of observations and covariates. In the past few years tree-based boosting algorithms, such as xgboost, have become recognized as some of the most powerful machine-learning models.

References

Anderson, D., S. Feldblum, C. Modlin, D. Schirmacher, E. Schirmacher, and N. Thandi, *A Practitioner's Guide to Generalized Linear Models* (3rd ed.), 2007, https://www.casact.org/pubs/dpp/dpp04/04dpp1.pdf.

Fortmann-Roe, S., "Understanding the Bias-Variance Tradeoff," 2012, http://scott.fortmann-roe.com/docs/BiasVariance.html (accessed July 10, 2017).

Frees, E., R. Derrig, and G. Meyers, eds., *Predictive Modeling Applications in Actuarial Science* (International Series on Actuarial Science), Cambridge: Cambridge University Press, 2014.

Goldburd, M., A. Khare, and D. Tevet, *Generalized Linear Models for Insurance Rating*, 2016, http://www.casact.org/pubs/monographs/papers/05-Goldburd-Khare-Tevet.pdf.

Hansen, J. V., and T. Heskes, "General Bias/Variance Decomposition with Target Independent Variance of Error Functions Derived from the Exponential Family of Distributions" in *Pattern Recognition*, Proceedings. 15th International Conference on (Vol. 2), pp. 207–210, IEEE, 2000.

James, G., D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, New York: Springer, 2013.

Jørgensen, B., *The Theory of Exponential Dispersion Models and Analysis of Deviance* (No. 51). Conselho Nacional de Desenvolvimento Científico e Tecnológico, Instituto de Matemática Pura e Aplicada, 1992.

Hastie, T., R. Tibshirani, and J. Friedman, *Elements of Statistical Learning: Data Mining, Inference and Prediction,* 2nd ed., New York: Springer, 2009.

McCullagh, P., and J. Nelder, *Generalized Linear Models*, 2nd ed., New York: Springer, 1989.

Ohlsson, E., and B. Johansson, *Non-Life Insurance Pricing with Generalized Linear Models*, New York: Springer, 2010.

Stansbury, D., "Model Selection: Underfitting, Overfitting, and the Bias-Variance Tradeoff," https://theclevermachine.wordpress.com/2013/04/21/model-selection-underfitting-overfitting-and-the-bias-variance-tradeoff/ (accessed July 26, 2017).

Vijayakumar, S., "The Bias-Variance Tradeoff," 2007, course material for Machine Learning and Sensorimotor Control class at University of Edinburgh http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf.

Wikipedia contributors, "Bias–variance tradeoff," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/Bias-variance_tradeoff (accessed July 17, 2017).

Appendices

A. Decomposition of test prediction error

As noted in Section 6, test prediction error can be decomposed into its irreducible error, squared bias and variance components as follows:

$$\mathbb{E}\left[(y-g(x))^2\right] = \underbrace{\mathbb{E}\left[(y-f(x))^2\right]}_{\text{Irreducible Error}} + \underbrace{(\mathbb{E}[g(x)]-f(x))^2}_{\text{Squared Bias}}$$

$$+ \underbrace{\mathbb{E}\left[(g(x)-\mathbb{E}[g(x)])^2\right]}_{\text{Variance}}$$

The following derivation mirrors that of Vijayakumar (2007). For simplicity, $g(x)$ is denoted as g, and $f(x)$ is denoted as f. The proof hinges on the following identities:

By definition we have

$$\mathbb{E}[y] = f.$$

As the distribution of y is independent of the sampling distribution $\mathcal{F}$

$$\mathbb{E}[yg] = \mathbb{E}[y]\mathbb{E}[g] = f\mathbb{E}[g].$$

First we expand the total test prediction error:

$$\begin{aligned}\mathbb{E}[(y-g)^2] &= \mathbb{E}[((y-f)+(f-g))^2] \\ &= \mathbb{E}[(y-f)^2]+\mathbb{E}[(f-g)^2] \\ &\quad + 2\mathbb{E}[(f-g)(y-f)] \\ &= \mathbb{E}[(y-f)^2]+\mathbb{E}[(f-g)^2] \\ &\quad + 2\big(\mathbb{E}[fy]-\mathbb{E}[f^2]-\mathbb{E}[gy]+\mathbb{E}[gf]\big)\end{aligned}$$

Noting that f is deterministic, the last term is found to be 0 as

$$\begin{aligned}&\big(\mathbb{E}[fy]-\mathbb{E}[f^2]-\mathbb{E}[gy]+\mathbb{E}[gf]\big) \\ &= f^2 - f^2 - \bar{g}f + \bar{g}f = 0\end{aligned}$$

Thus, we find

$$\mathbb{E}[(y-g)^2] = \mathbb{E}[(y-f)^2]+\mathbb{E}[(f-g)^2],$$

which is the familiar breakdown of test prediction error into irreducible error and explainable error. The second term (the MSE between the true function and the model-estimator) can be further expanded:

$$\begin{aligned}\mathbb{E}[(f-g)^2] &= \mathbb{E}[(f-\mathbb{E}[g]+\mathbb{E}[g]-g)^2] \\ &= \mathbb{E}[(f-\mathbb{E}[g])^2]+\mathbb{E}[(\mathbb{E}[g]-g)^2] \\ &\quad + 2\mathbb{E}[(\mathbb{E}[g]-g)(f-\mathbb{E}[g])]\end{aligned}$$

The first term, $\mathbb{E}[(f-\mathbb{E}[g])^2]$, is squared bias, and the second term, $\mathbb{E}[(\mathbb{E}[g]-g)^2]$, is the variance of the estimator. The cross-product term's reduction to zero is apparent in its expansion:

$$\begin{aligned}&2\mathbb{E}[(\mathbb{E}[g]-g)(f-\mathbb{E}[g])] \\ &= 2\big(\mathbb{E}[fE[g]]-\mathbb{E}[g]^2-\mathbb{E}[fg]+\mathbb{E}[gE[g]]\big) \\ &= 2\big(fE[g]-\mathbb{E}[g]^2-f\mathbb{E}[g]+\mathbb{E}[g]^2\big) \\ &= 0\end{aligned}$$

Combining these identities we arrive at the stated decomposition.

B. Graphs of simulated polynomial models

This appendix contains the graphs of polynomial fits of order one through ten from the simulation discussed in Section 7.

Figure 9.

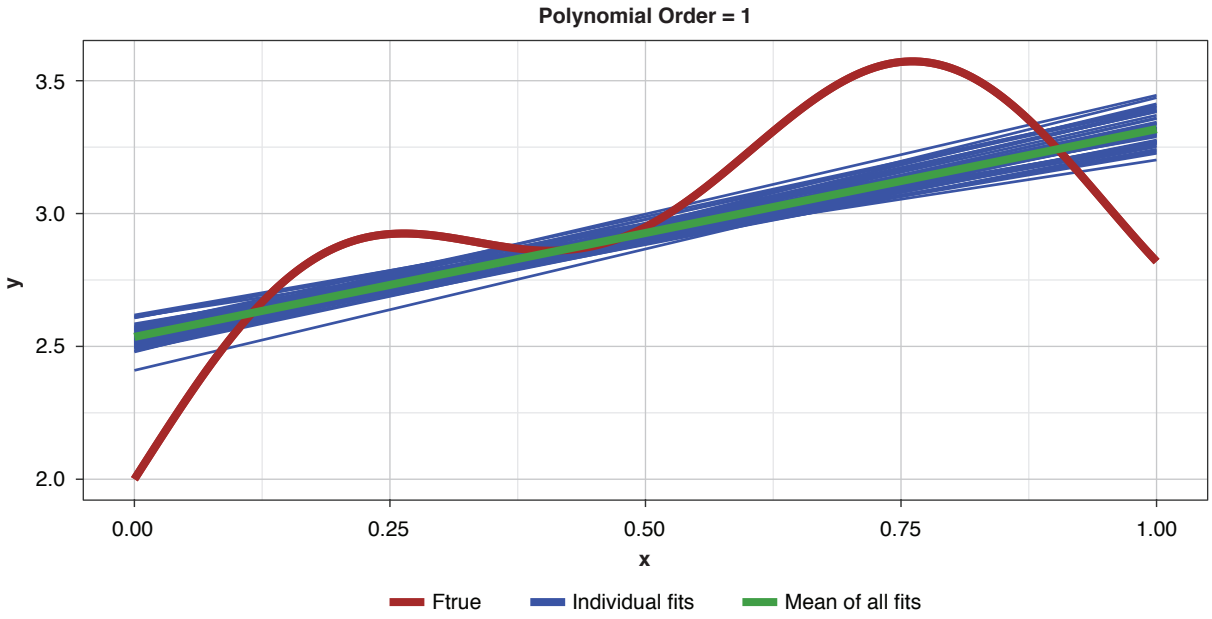


Figure 10.

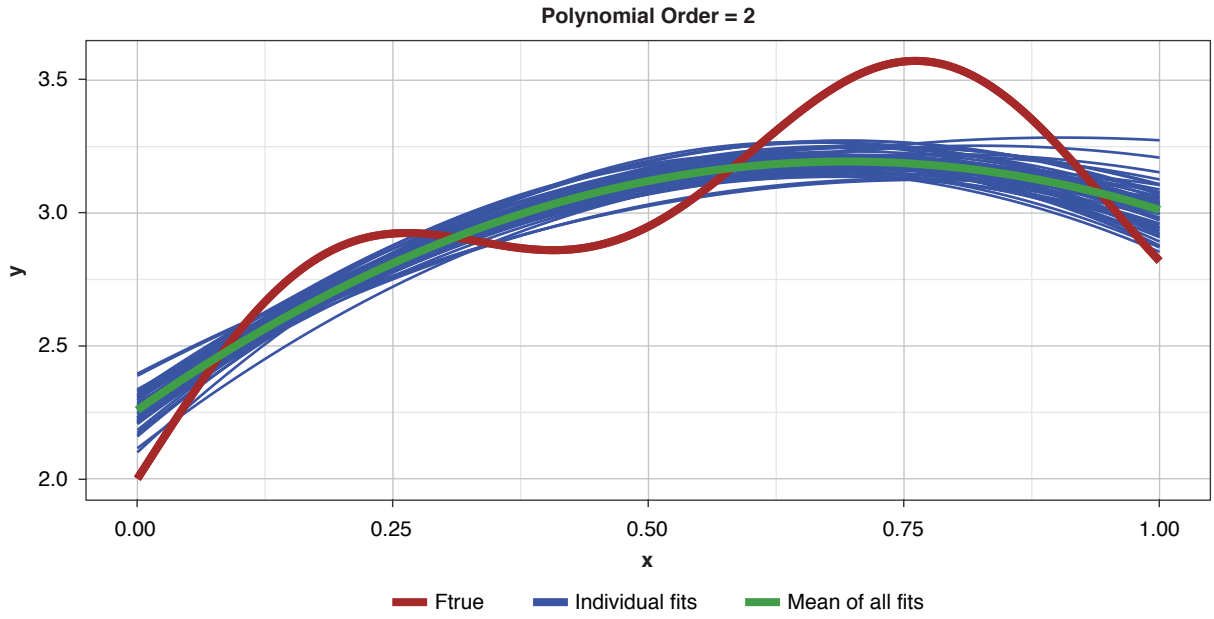


Figure 11.

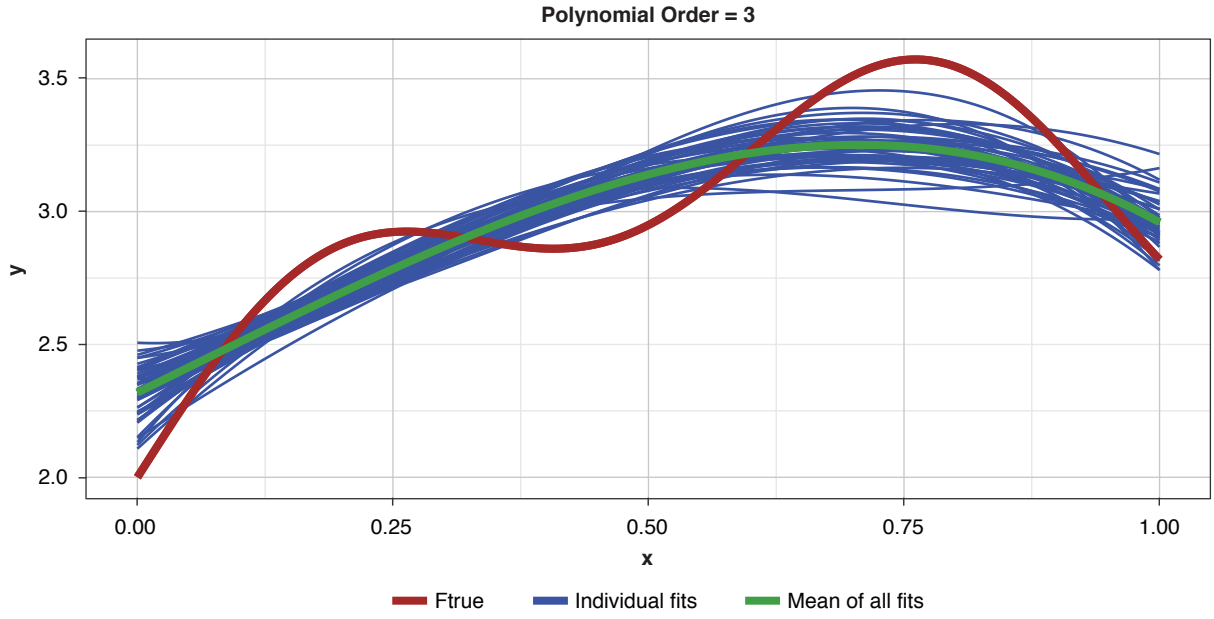


Figure 12.

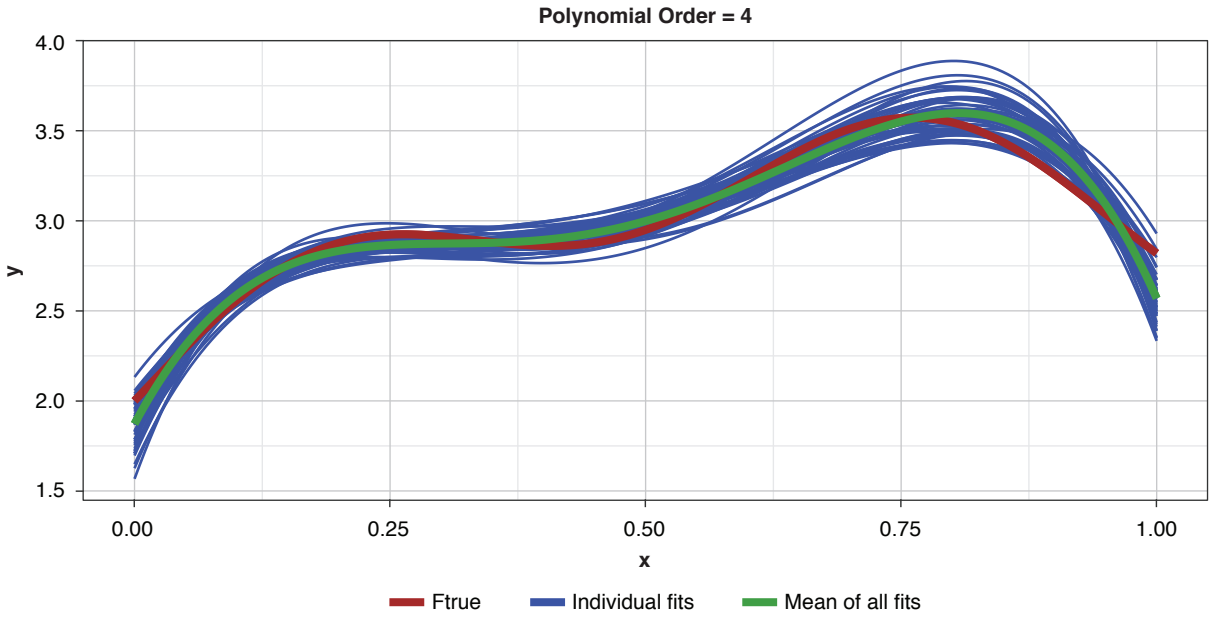


Figure 13.

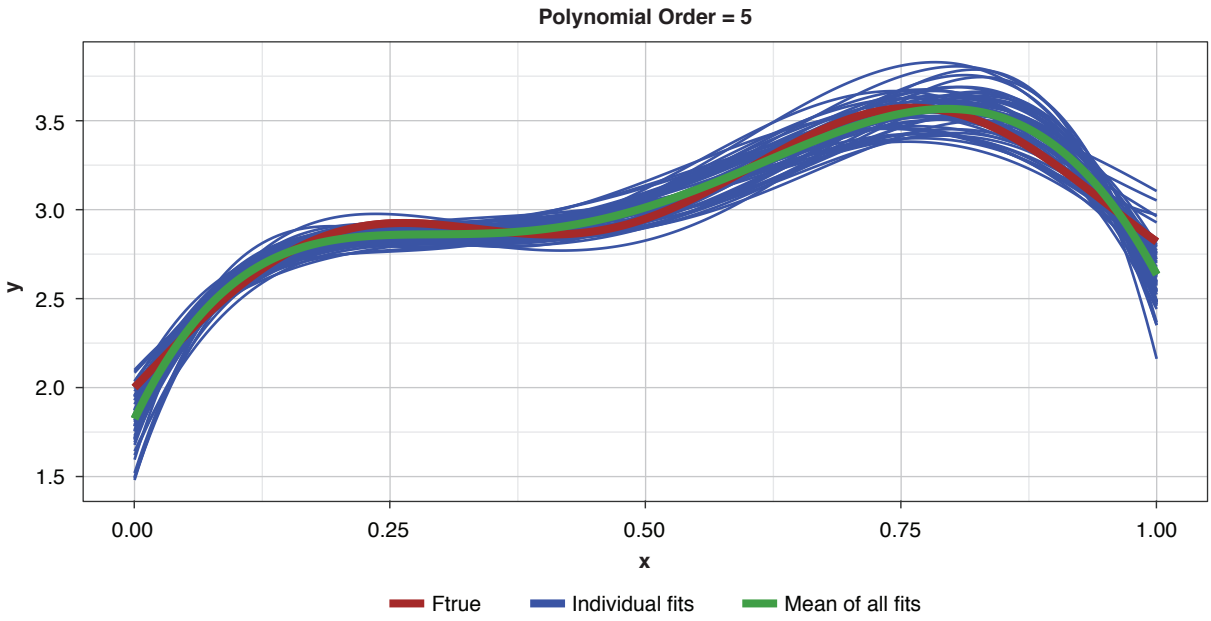


Figure 14.

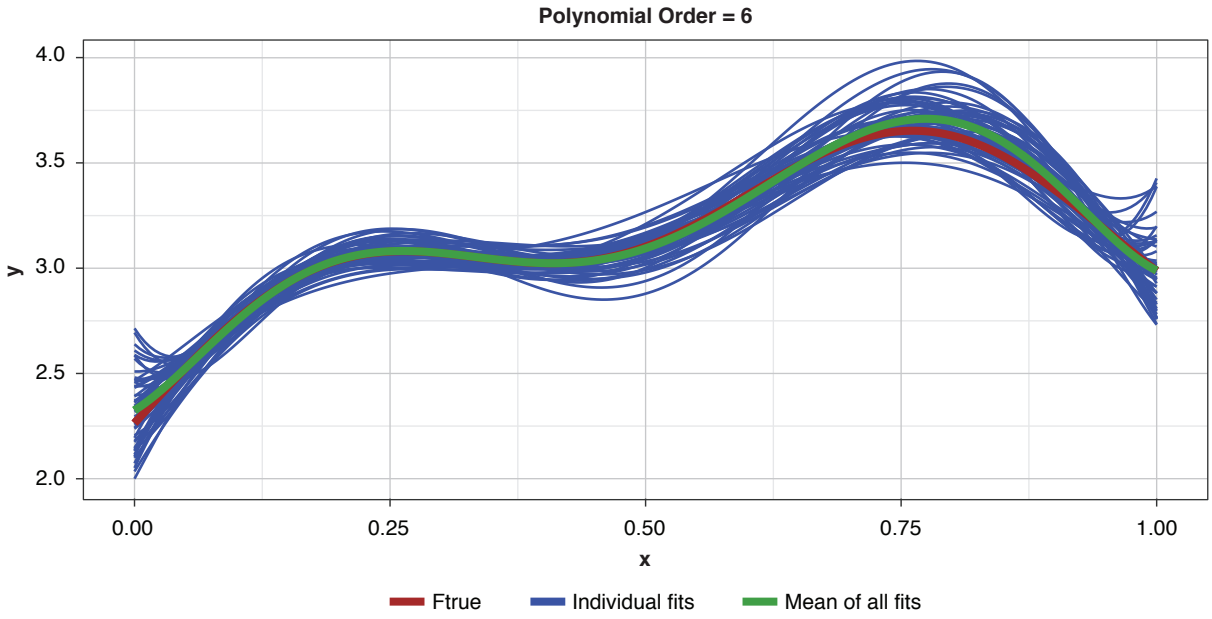


Figure 15.

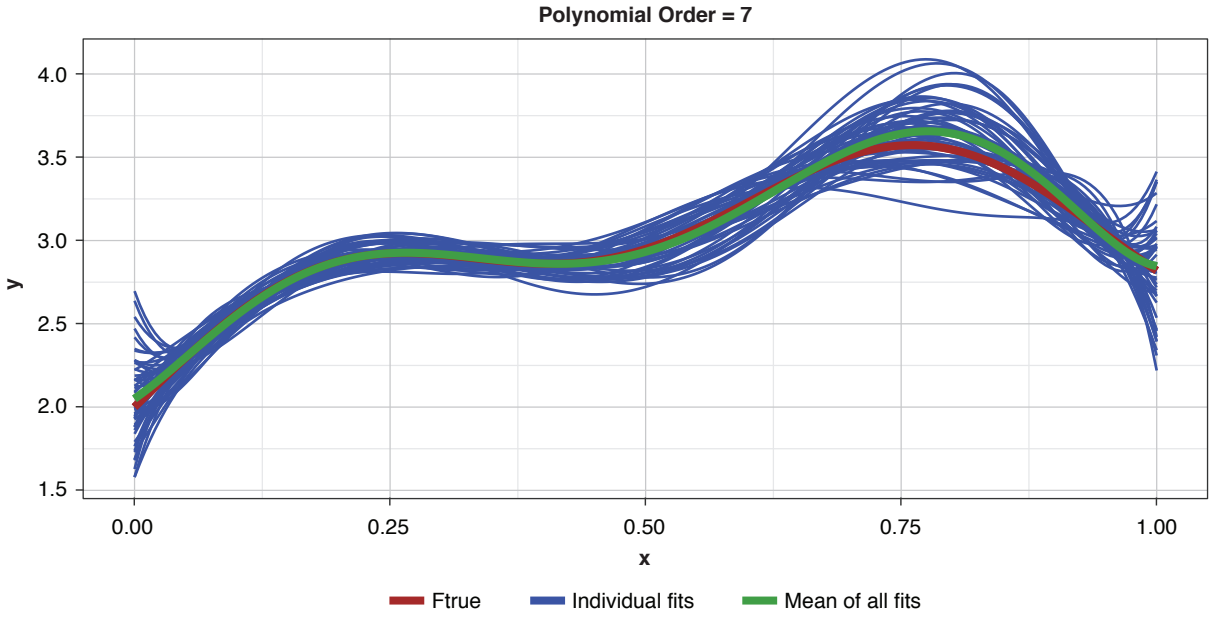


Figure 16.

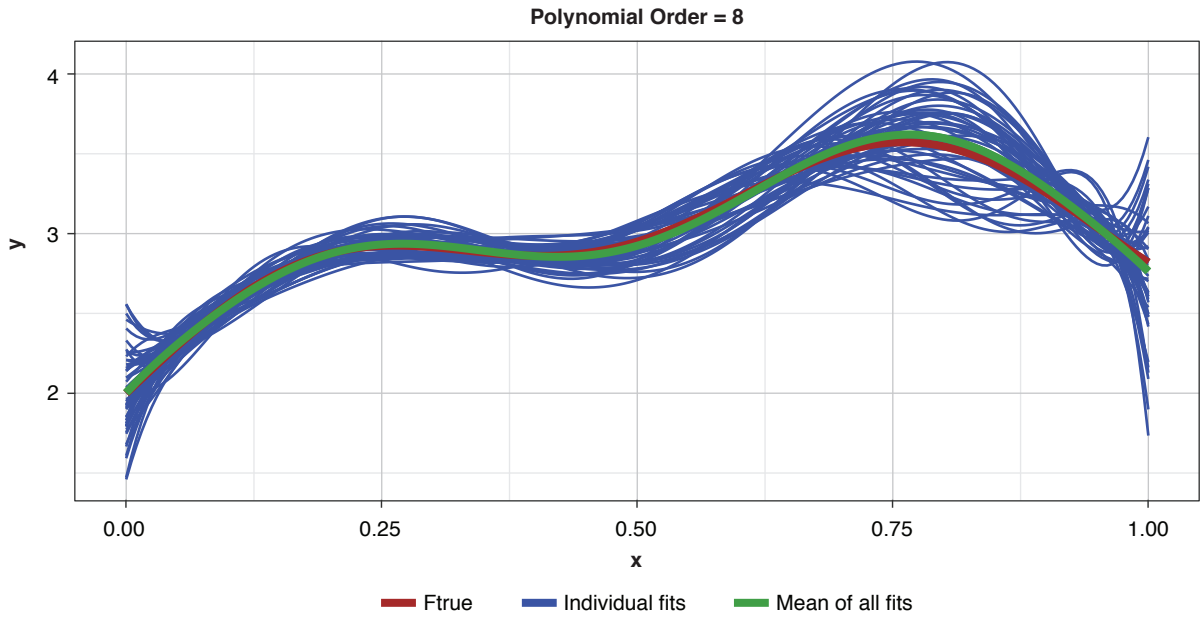


Figure 17.

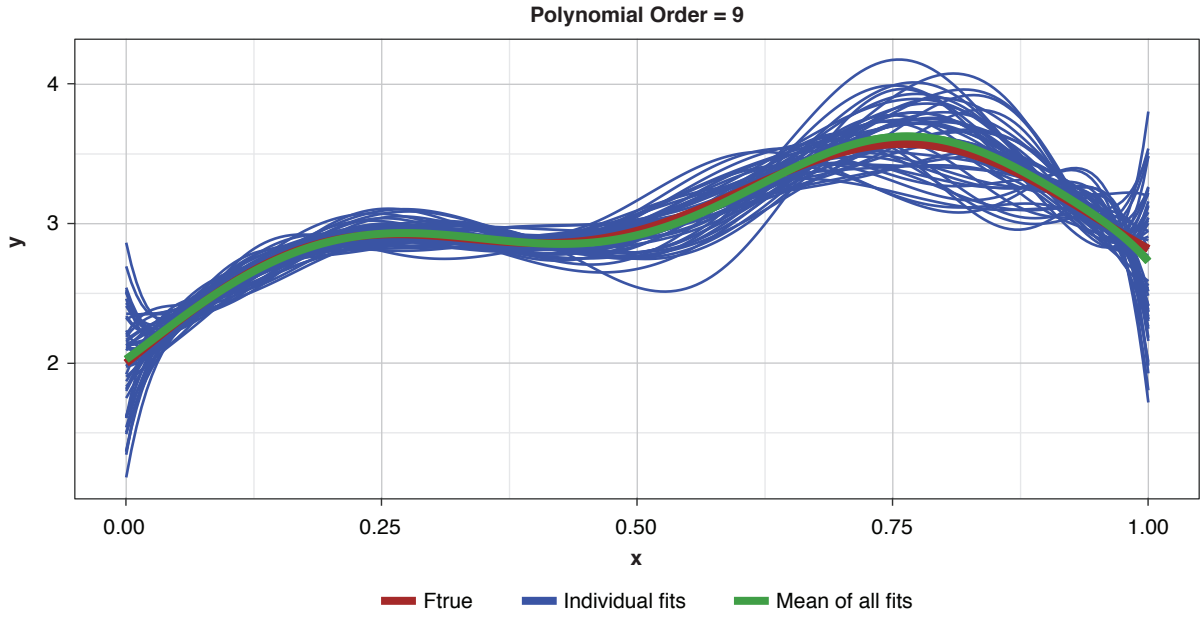
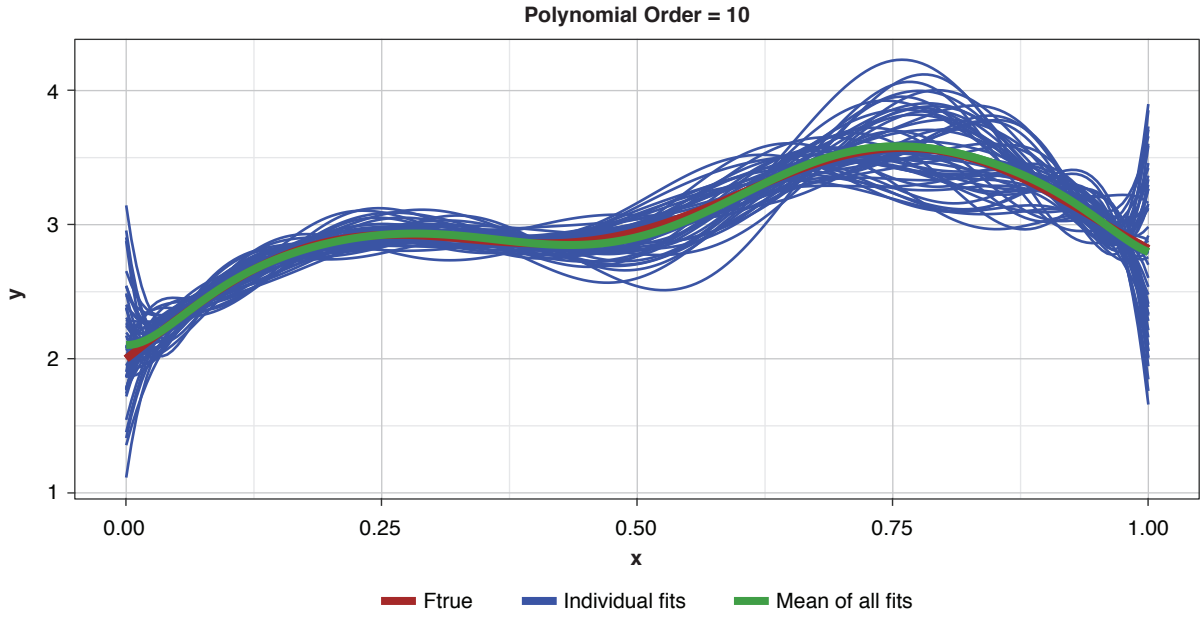


Figure 18.



C. Bias-variance decomposition of deviance

Following McCullagh and Nelder (1989) we assume that the distribution y (given x) is from the exponential family

$$f(y|\theta,\phi)=\exp\left\{\frac{y\theta-b(\theta)}{\phi}+c(y,\phi)\right\}.$$

As shown in Jørgensen (1992), every distribution in this exponential family is fully determined by the variance relationship linking the variance of y to a function, v, of the mean. That is,

$$\text{Var}[y]=\phi v(\mu),$$

where $\mu=\mathbb{E}[y]$.

We can write the deviance in terms of the distribution's variance relationship as

$$d(y,\mu)=2\int_{\mu}^{y}\frac{y-t}{v(t)}\text{d}t.$$

Note that we are working with the unscaled deviance. The scaled deviance is given by $d^*=d/\phi$.

Most resources on GLMs do not rely on the above representation of deviance. More common is the likelihood-based definition of deviance. For completeness, we briefly show equivalence here. In Anderson et al. (2007), it is stated that the forms are equivalent.

As shown in Ohlsson and Johansson (2010), the parameters θ and function b are related to the mean and variance function through the following identities:

$$b'(\theta)=\mu\Rightarrow\theta=b'^{-1}(\mu)$$

$$b''(b'^{-1}(\mu))=\frac{Var[y]}{\phi}=v(\mu)$$

Denote the two forms of deviance as follows. The integral deviance:

$$d_1(y,\mu)=\frac{2}{\phi}\int_{\mu}^{y}\frac{y-t}{v(t)}\text{d}t.$$

The likelihood deviance, which is twice the difference of the saturated model and the fitted model:

$$\begin{aligned}d_2(y,\mu)&=2[loglik(y,y)-loglik(y,\mu)]\\&=\frac{2}{\phi}[y\theta(y)-b(\theta(y))-y\theta(\mu)+b(\theta(\mu))]\\&=\frac{2}{\phi}[yb'^{-1}(y)]-\frac{2}{\phi}[b(b'^{-1}(y))]\\&\quad-\frac{2}{\phi}[yb'^{-1}(\mu)]+\frac{2}{\phi}[b(b'^{-1}(\mu))]\end{aligned}$$

Next we differentiate d_1 and d_2 with respect to parameter μ.

$$\frac{\mathrm{d}}{\mathrm{d}\mu} d_1(y,\mu) = -\frac{2}{\phi}\frac{y-\mu}{v(\mu)}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mu} d_2(y,\mu) = -\frac{2}{\phi}\left[\frac{y}{b''(b'^{-1}(\mu))} - \frac{b'(\theta)}{b''(b'^{-1}(\mu))}\right]$$

$$= -\frac{2}{\phi}\frac{y-\mu}{v(\mu)}$$

As the derivatives are equal, the two forms differ by an additive constant. Evaluating both deviances at $\mu = y$, we find that the constant is zero, showing equivalence.

Lemma 13.1. *Define the deviance-minimizing estimator of model g at point x as*

$$\tilde{g}(x) = \underset{h}{\operatorname{argmin}}\, \mathbb{E}[d(h, g(x))].$$

Then,

$$\mathbb{E}\left[\int_{g(x)}^{\tilde{g}(x)} \frac{1}{v(t)}\,dt\right] = 0.$$

Proof. We drop x for simplicity. As $\tilde{g}$ is the minimizer of $\mathbb{E}[d(h, g)]$ we have

$$\frac{\mathrm{d}}{\mathrm{d}\tilde{g}}\mathbb{E}\left[\int_{g}^{\tilde{g}} \frac{\tilde{g}-t}{v(t)}\,\mathrm{d}t\right] = 0.$$

We assume that the distribution associated with the sampling space is regular enough to interchange the derivative and expectation. Assuming $v(t)$ is continuous we can apply the Leibniz integral rule to find

$$0 = \frac{\mathrm{d}}{\mathrm{d}\tilde{g}}\mathbb{E}\left[\int_{g}^{\tilde{g}} \frac{\tilde{g}-t}{v(t)}\,\mathrm{d}t\right] = \frac{\tilde{g}-\tilde{g}}{v(\tilde{g})} + \mathbb{E}\left[\int_{g}^{\tilde{g}} \frac{1}{v(t)}\,\mathrm{d}t\right]$$

$$= \mathbb{E}\left[\int_{g}^{\tilde{g}} \frac{1}{v(t)}\,\mathrm{d}t\right]$$

While Lemma 13.1 is a necessary technical result, we also gain a method in which to calculate the deviance-minimizing estimator. Assume $v(t) = t^p$. Applying the conclusion the Lemma 13.1, we obtain the following results:

$$\tilde{g} = \begin{cases} \mathbb{E}[g] & p = 0 \\ \exp\{\mathbb{E}[\ln(g)]\} & p = 1 \\ \mathbb{E}\left[g^{1-p}\right]^{\frac{1}{1-p}} & p > 1 \end{cases}.$$

Consider the case of the canonical link. As we have used g to denote the fitted model in this paper, we let q denote the link function for our GLM. That is, $g = q^{-1}(\eta)$, where η is the linear predictor. We have the following intuitive result that the deviance-minimizing estimator is equal to the inverse link of the expected value of the linear predictor

$$\tilde{g} = \begin{cases} \mathbb{E}[\eta] & p = 0 \\ \exp\{\mathbb{E}[\eta]\} & p = 1 \\ \mathbb{E}[\eta]^{\frac{1}{1-p}} & p > 1 \end{cases}.$$

In fact this relationship holds in general for all distributions.

Corollary 13.2. *Let q represent the link for the fitted GLM. For the canonical link q, the deviance-minimizing estimator equals*

$$\tilde{g} = q^{-1}(\mathbb{E}[\eta]).$$

Proof. The key is that the canonical link satisfies $q'(\mu) = \dfrac{1}{v(\mu)}$. Substituting q' into the result of Lemma 13.1 and applying the fundamental theorem of calculus, we naturally arrive at the relationship for $\tilde{g}$.

We can think of the deviance-minimizing estimator is an average on the space transformed by the variance function v and link q.

We now state and prove the main theorem of this appendix.

Theorem 13.3 *(Deviance Decomposition) The expected deviance for estimator g at a test point* (x, y) *can be decomposed as follows*

$$\mathbb{E}[d(g(x), y)] = \mathbb{E}[d(y, f(x))] + d(f(x), \tilde{g}(x)) + \mathbb{E}[d(\tilde{g}(x), g(x))],$$

where $\tilde{g}(x) = argmin_h \mathbb{E}[d(h, g(x))]$ is the deviance-minimizing estimator.

Proof. We omit the multiplicative 2 and x for notational simplicity. Using basic properties of integrals, we write

$$\begin{aligned}\mathbb{E}[d(g(x), y)] &= \mathbb{E}\left[\int_g^y \frac{y-t}{v(t)}\,dt\right] \\ &= \mathbb{E}\left[\int_f^y \frac{y-t}{v(t)}\,dt\right] + \mathbb{E}\left[\int_{\bar{g}}^f \frac{y-t}{v(t)}\,dt\right] \\ &\quad + \mathbb{E}\left[\int_g^{\bar{g}} \frac{y-t}{v(t)}\,dt\right].\end{aligned}$$

The first term is simply

$$\mathbb{E}\left[\int_f^y \frac{y-t}{v(t)}\,dt\right] = \mathbb{E}[d(y, f)].$$

For the second component we note that $\mathbb{E}[y] = f(x)$ and importantly that the expectation relative to sampling distribution is independent of f, y, and $\bar{g}$. Thus,

$$\begin{aligned}\mathbb{E}\left[\int_{\bar{g}}^f \frac{y-t}{v(t)}\,dt\right] &= \int_{\bar{g}}^f \frac{\mathbb{E}[y]-t}{v(t)}\,dt = \int_{\bar{g}}^f \frac{f-t}{v(t)}\,dt \\ &= d(f, \bar{g}).\end{aligned}$$

Finally, for the third component:

$$\begin{aligned}\mathbb{E}\left[\int_g^{\bar{g}} \frac{y-t}{v(t)}\,dt\right] &= \mathbb{E}\left[\int_g^{\bar{g}} \frac{y-\bar{g}+\bar{g}-t}{v(t)}\,dt\right] \\ &= \mathbb{E}\left[\int_g^{\bar{g}} \frac{y-\bar{g}}{v(t)}\,dt\right] + \mathbb{E}\left[\int_g^{\bar{g}} \frac{\bar{g}-t}{v(t)}\,dt\right] \\ &= \mathbb{E}\left[\int_g^{\bar{g}} \frac{y-\bar{g}}{v(t)}\,dt\right] + \mathbb{E}[d(\bar{g}, g)].\end{aligned}$$

As y and the sampling distribution, $\mathcal{F}$, are distributed independently, we can reduce the first term above to zero using Lemma 13.1.

$$\mathbb{E}\left[\int_g^{\bar{g}} \frac{y-\bar{g}}{v(t)}\,dt\right] = (f-\bar{g})\mathbb{E}\left[\int_g^{\bar{g}} \frac{1}{v(t)}\,dt\right] = 0.$$

Final comments on the deviance decomposition:

The above decomposition relied on the unscaled deviance, which does not include the dispersion parameter ϕ. If ϕ is assumed fixed and known, then the above decomposition extends to the scaled deviance. This may not seem like a reasonable assumption to make, but we note two important points. First the traditional squared error, $(y - g(x))^2$, for normal regression implicitly ignores the dispersion parameter σ^2. Thus, when we consider the bias-variance tradeoff in the context of normally distributed data, we are implicitly assuming the dispersion parameter σ^2 is fixed and constant. Further, one can form a quasi-likelihood as discussed in McCullagh and Nelder (1989). With a quasi-likelihood, a dispersion parameter only needs to be estimated in order to produce inferential statistics. If the modeler is simply interested in goodness of fit relative to the prescribed deviance function, then the dispersion parameter is unnecessary.

D. Example illustrating the connection to credibility—derivation of the expected mean squared error

Given the distribution described in Section 9, calculate the expected mean squared error for

$$g_Z(x) = \bar{y}_x Z + (1-Z)\bar{y}.$$

We compute the expected test error at test point (x, y) where $x = u$. From the symmetry of the random variable y it follows that the expected test error is the same for $x = d$. We assume each training set is of size n and that we sample u and d equally.

$$\begin{aligned}\mathbb{E}[(y-g_Z)^2|x=u] &= \mathrm{Var}[y-g_Z|x=u] \\ &\quad + \mathbb{E}[y-g_Z|x=u]^2 \\ &= \mathrm{Var}[y|x=u] + \mathrm{Var}[g_Z|x=u] \\ &\quad - 2Cov(y, g_Z|x=u) \\ &\quad + \mathbb{E}[y-g_Z|x=u]^2 .\end{aligned}$$

We further simplify the above equation using the following identities.

From the problem statement

$$\mathrm{Var}[y|x=u] = \sigma^2$$

$$\mathbb{E}[y|x=u] = a$$

$$\mathbb{E}[\bar{y}] = 0$$

The unconditional variance of y can be derived as follows. Recall that we assume a 50/50 sampling between u and d.

$$\begin{aligned}\mathrm{Var}[y] &= \mathbb{E}[\mathrm{Var}[y|x]] + \mathrm{Var}[\mathbb{E}[y|x]] \\ &= \mathbb{E}[\sigma^2] + \mathrm{Var}[+a, -a] \\ &= \sigma^2 + a^2\end{aligned}$$

As the sampling distribution is independent of the test distribution, $Cov(y, g_Z|x = u) = 0$. Taking the expectation of g_Z

$$\mathbb{E}[g_Z] = Z\mathbb{E}[\bar{y}_x|x=u] + (1-Z)\mathbb{E}[\bar{y}] = Za.$$

And so

$$\begin{aligned}\mathbb{E}[y-g_Z|x=u]^2 &= (\mathbb{E}[y|x=u] - \mathbb{E}[g_Z|x=u])^2 \\ &= (a-aZ)^2 \\ &= a^2(1-Z)^2 .\end{aligned}$$

Next we compute $Var[g_Z|x = u]$:

$$\begin{aligned}\mathrm{Var}[g_Z|x=u] &= Z^2\mathrm{Var}[\bar{y}_x|x=u] + (1-Z)^2\,\mathrm{Var}[\bar{y}] \\ &\quad + 2Z(1-Z)Cov(\bar{y}_x, \bar{y}|x=u)\end{aligned}$$

We investigate each of these parts in detail:

$$\begin{aligned}\mathrm{Var}[\bar{y}_x|x=u] &= \frac{2}{n}\mathrm{Var}[y|x=u] \\ &= \frac{2}{n}\sigma^2.\end{aligned}$$

$$\begin{aligned}Var[\bar{y}] &= \frac{1}{n}Var[y] \\ &= \frac{1}{n}(\sigma^2 + a^2).\end{aligned}$$

$$\begin{aligned}\mathrm{Cov}(\bar{y}_x, \bar{y}|x=u) &= Cov\left(\bar{y}_x, \frac{1}{2}\bar{y}_x + \frac{1}{2}\bar{y}_y \middle| x=u\right) \\ &= Cov\left(\bar{y}_x, \frac{1}{2}\bar{y}_x \middle| x=u\right) \\ &\quad + Cov\left(\bar{y}_x, \frac{1}{2}\bar{y}_y \middle| x=u\right) \\ &= \frac{1}{2}\mathrm{Var}[\bar{y}_x|x=u] + 0.\end{aligned}$$

Putting all the pieces together, we observe:

$$\begin{aligned}\mathbb{E}[(y-g_Z)^2] &= \sigma^2 + \frac{2}{n}Z^2\sigma^2 + \frac{1}{n}(1-Z)^2(\sigma^2 + a^2) \\ &\quad + \frac{2}{n}Z(1-Z)\sigma^2 + a^2(1-Z^2)\end{aligned}$$

Rearranging and simplifying, we arrive at the desired form

$$\begin{aligned}\mathbb{E}[(y-g_Z)^2] &= \sigma^2 + a^2(1-Z)^2 \\ &\quad + \frac{1}{n}(2Z\sigma^2 + (1-Z)^2(\sigma^2 + a^2)).\end{aligned}$$