Overfitting

Clearing the Confusion: Understanding overfitting–bias and variance of WHAT?

N. Matloff, UC Davis

Goal

Explanations of overfitting in machine learning tend to be frustratingly vague. We for instance hear "An overfitted model predicts poorly on new examples," which is a definition, not an explanation. Or we hear that overfitting results from "fitting the training data too well" or that we are "fitting the noise," again rather unsatisfying.

Somewhat better are the explanations based on the famous Bias-Variance Tradeoff:

> Mean squared prediction error is the sum of squared bias and variance. Bias and variance are at odds with each other. The lower the bias, the more the variance.

OK, but the bias and variance of WHAT? Our treatment here is based on that notion, **but with more emphasis on what it is that we are really talking about** in discussing bias and variance.

Required Background

Very basic statistical concepts + patience (document is a bit long).

Setting

Let Y denote our outcome variable, and X represent the vector of our predictors/features. For instance, Y might be human weight, with X being (height, age). We include binary classification problems, with Y = 1 or 0. (In multiclass problems, Y is a vectors of 1s and 0s, with only one component being 1 at a time.)

The "True" Relation between Y and X

The *regression function of Y on X*, (t), is defined to be the mean Y for the given X values. In the weight/height/age example, (68.2,32.9) is the mean weight of

all people of height 68.2 and age 32.9.

is the best predictor of weight based on height and age, the ideal. Note the phrase "based on," though. If we had a third predictor variable/feature available, say waist size, there would be another for that 3-predictor setting, better than the 2-predictor one.

Note that is a population entity, which we estimate from our sample data. Denote the estimated (t) by r(t). Say we are studying diabetic people. (t) gives us the relation of weight vs. height and age in the entire population of diabetics; if our study involves 100 patients, that is considered a sample from the population, and our estimate r(t) is based on that sample.

(Note: "Sample" means our entire dataset; we say, "A sample of 100 peoplee," not "We have 100 samples.")

Also, though some books use the term "regression" to mean a linear model, the actual definition is unrestricted; it applies just as much to, say, random forests as to linear models.

In a binary classification problem, this function reduces to the probability of class 1.

Bias

Parametric methods such as linear and logistic models have a bias, in the sense of the inaccuracy of the model itself. Say we use a linear model to predict weight from height, i.e. our model for (height) is

(weight) = mean weight = 0 + 1 height

(This is called a "parametric" model, in that it expresses r(t) in terms of some parameters, in this case the i.)

Our sample-based estimate is

r(weight) = mean weight = b0 + b1 height

The bi are the sample estimates of the i.

Though the true population relation may be somewhat linear, the linear model cannot be perfectly correct. So, no matter how much data we have, our estimated r(t) will not converge to the true (t) as the sample size grows; r(t) WILL converge to something, but that something will be the best-fitting LINEAR FUNCTION for the population, not the best-fitting FUNCTION.

The bias here is the difference between the true (t) and the limiting value of r(t) as the sample size grows to infinity. It will be different at each t, probably larger at larger t. E.g. the bias for the linear model may be larger for taller people.

For model-free, i.e. nonparametric, methods, the bias is subtler. Consider k-Nearest Neighbors (k-NN), say again predicting weight from height. To calcu-

late, say, r(68.2), we find the k points in our data closest to 68.2, then take as r(68.2) the average weight among those people. Bias arises as follows: Say we wish to predict the weight for a person of height 64, which is on the shorter end of the height range. Then the neighboring data points are likely be predominantly taller than 64, and since our prediction will consist of the mean weight among the neighbors, this 64-inch tall person's weight will likely be overestimated.

So again, there is a bias, again dependent on the value of t of interest, just as in the case of parametric methods. However, for this very reason, we let the size of the neighborhood get smaller as our dataset size grows, causing the bias to shrink. There is no such (direct) remedy for a linear model. There are analogs of this point for random forests, neural nets etc. On the other hand, a linear model will have a smaller variance, our next topic:

Variance

This refers to sampling variance, which measures the degree of instability of one's estimated r(t) from one sample to another, e.g. the variation of the bi from one sample to another in our linear model above.

The same holds for nonparametric models. In the k-NN example above, say we take too small a value of k, even k = 1. So we estimate r(68.2) to be the weight of whichever person in our sample is closest to 68.2. Clearly this value varies a lot from one sample to another, hence a large variance.

The relevance of variance is difficult for many nonspecialists in statistics/machine learning to accept. "But we only have one sample," some might object to the analysis in the preceding paragraphs. True, but we are interested in the probability that that our sample is representative of the population. In a gambling casino, you may play a game just once, but you still would like to know the probability of winning. The same is true in data science, and that in turn means that sampling variance is key.

A U-Shaped Curve

We stated above that a linear model cannot be exactly correct, even with an unlimited amount of data. So, why not a quadratic model?

$$\text{(weight)} = \text{mean weight} = \beta_0 + \beta_1 \text{ height} + \beta_2 \text{ height}^2$$

This includes the linear model as a special case ($\beta_2 = 0$), but also is more general. In the population, the best-fitting quadratic model will be more accurate than the best-fitting linear one. How about one of degree 3? With higher and higher degree polynomials, we can reduce the bias to smaller and smaller amounts–if we have infinitely many data points.

But in finite samples, the higher the degree of the polynomial, the larger the variance. (We have more parameters to estimate, and I like to think in terms of them "sharing" the data, less data available to each one.)

So, we have a battle between bias and variance! As we increase the degree, first

1, then 2, then 3 and so on, the bias initially shrinks a lot while the variance grows only a little. But eventually, inceasing the degree by 1 will reduce bias only a little, while variance increases a lot.

Result:

> If we plot prediction error vs. degree, we typically will get a U-shaped curve. Bias reduction dominates variance increase initially, but eventually variance overpowers bias. Once we pass the low point of the curve, we we overfitting.

The same is true for k-NN, plotting prediction error against k. As noted, we can achieve smaller bias with smaller k. The latter situation means smaller neighborhoods, making the problem of "using tall people to predict a short person's weight" less likely. But smaller k means we are taking an average over a small set of people, which will vary a lot from one sample to another.

Or, consider the number of predictors/features, say in the Million Song Dataset. The goal is to predict the year of release of a song, based on various audio characteristics of the song, 91 features in all. Let p denote the number of features used in our analysis. We might use just the first feature (needn't be the first, just an example), p = 1, or the first two, p = 2 and so on. The more features we use, the smaller the bias, but the larger the variance.

Empirical Illustration

I used the Million Song Dataset To reduce the amount of computation, I used only a random subset of 10,000 songs, and only two audio variables.

As noted, mean squared prediction error is a sum of bias squared and variance. I calculated mean absolute prediction error (MAPE), but the principle is the same; it still combines bias and variance.

My goal was to show that:

1. As degree increases, MAPE at first falls but later rises, i.e. the "U-shape" discussed above.
2. But as degree increases, variance *always* increases.

Combining these two points, we can observe the battle between bias and variance in their impact on MAPE. At first the reduction in bias dominates the increase in variance, but for larger degrees, the opposite it true.

Remember, variance is *always* increasing as degree increases here. But how can we measure variance to show this numerically? Most readers here know that R reports standard errors of estimates, but for different degrees with have different numbers of estimated parameters, thus with noncomparable standard errors.

Instead, I took the first song in the dataset, and let x = the audio characterstics of that particular song. I then found the variance of r(x) (a matrix quadratic

form in x and the estimated covariance matrix of the vector of coefficients of the polynomial).

I fit degrees 1 through 12. (Again, to save on computation, I used interaction terms only of degree 2.) For cross-validated MAPE values, I used 20 random holdout sets of size 1000. Here is the output:

```
           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
[1,] 0.02440784 0.02718803 0.03184419 0.04229401 0.04945168 0.05450264
[2,] 7.86296820 7.74149419 7.74222432 7.66320622 7.70638321 7.69907202
           [,7]       [,8]     [,9]        [,10]    [,11]    [,12]
[1,] 0.06572677 0.07980077 0.114887 -0.008906266       NA       NA
[2,] 7.73367546 7.70928618 7.733230  7.793813681 9.980302 17.36951
```

In the top row, one can see variance steadily increasing through degree 9. Starting with degree 10, there are serious numerical issues, arising from exact or nearly exact collinearity among the various terms in the polynomial.

The code is here.

MAPE is shown in the second row. There is a general "U" trend, albeit a shallow one. We seem to have *numerical* overfitting starting at degree 10, but *statisical* overfitting starting at degree 5 or so.

Double Descent

In recent years, the phenomenon of "double descent'' has caused quite a stir in the statistics and machine learning communities. It turns out that one can actually have *two* U-shapes, the second one coming immediately after the first. Here is an example, also using the Million Song dataset, but in this case using more and more columns of the dataset rather than using polymials of increasing degree.

Many people have theorized as to the cause of this.

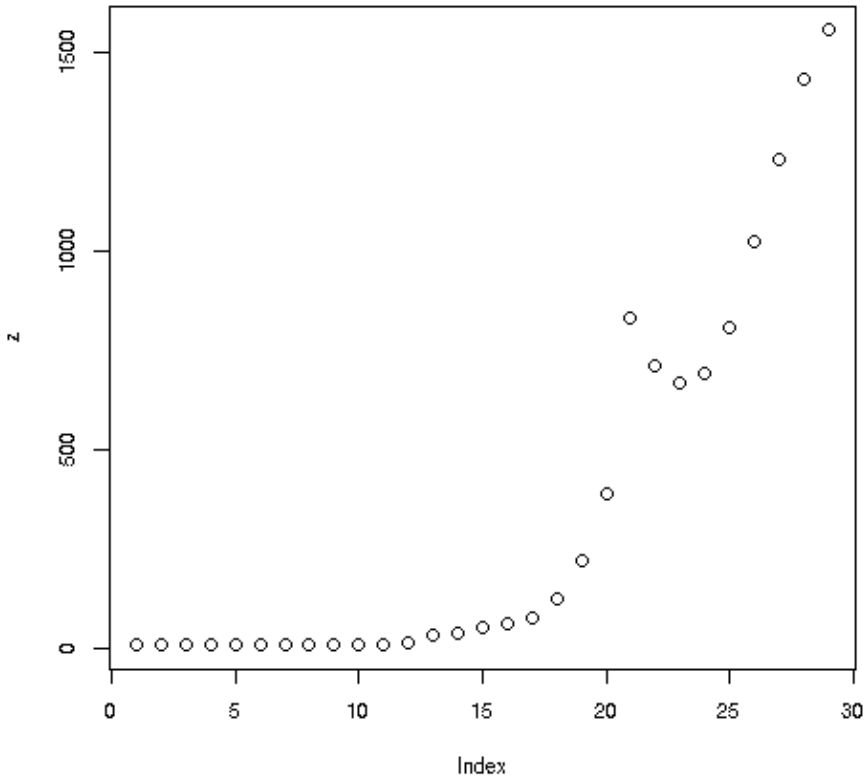

Figure 1: alt text