The Bias-Variance Tradeoff, Cross-Validation, and Overfitting in Prediction

Prof. George S. Easton

Introduction

1. Suppose that

$$Y = f(X) + \varepsilon$$

2. Also suppose that we have randomly divided our data into training and validation samples (and hopefully a final test sample as well).
3. Further, suppose that we have calculated, using the training data, an estimate of $f(\mathrm{X})$ which we will denote by $\hat{f}(\mathrm{X})$.

Example

- In linear regression $\hat{f}(X) = \hat{\alpha} + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$
- In this case, the fitted coefficients depends on the *Y*-values and *X*-values in the training sample.
- So $\hat{f}(X)$ depends on the training data.
- A more complete notation would be

$$\hat{f}(X; Y_{\text{train}}, X_{\text{train}})$$

Reminder: Bias and Variance

- Suppose S is a statistic which estimates θ.
  - e.g., the fitted regression coefficient $\hat{\beta}$ estimates the true coefficient β .
- The bias of S is:

$$\text{Bias}(S) = E(S) - \theta$$

- The variance of S is:

$$\text{Var}(S) = \text{E}\left[(S - E(S))^2 \right]$$

Reminder (con't)

- So, the bias of $\hat{f}(X)$ is

$$\text{Bias}(\hat{f}(X)) = E\left[\hat{f}(X)\right] - f(X)$$

- The variance of $\hat{f}(X)$ is:

$$\text{Var}(\hat{f}(X)) = E\left[\left(\hat{f}(X) - E\left[\hat{f}(X)\right]\right)^2\right]$$

Bias-Variance Trade-Off

Then it can be shown that

$$E\left[(Y - \hat{f}(X))^2\right] =$$

$$E\left[(\hat{f}(X) - \mathrm{E}[\hat{f}(X)])^2\right] +$$

$$\left(E[\hat{f}(X)] - f(X)\right)^2 +$$

$$\sigma^2$$

Expected mean-squared error (MSE) on the validation sample

Variance of the fit

Squared bias

Variance of the error

Note: Y is a new observation. $\hat{f}(X)$ is calculated from previous Y's so they are uncorrelated.

Bias-Variance Tradeoff (con't)

- Very, very important idea:

$$\text{Prediction MSE} = \left[\text{Bias of the Fit}\right]^2 + \text{Variance of the Fit} + \text{Variance of the Error}$$

- Sometimes said as MSE=Bias+Variance (which, of course, is not quite right).

Training Sample MSE

- On the training sample, the MSE always goes down as the model $\hat{f}(X)$ becomes more complex (has more parameters).
- This is intuitive: the more "flexibility" you have in fitting the data, the closer you should be able to come to a perfect fit.

Validation Sample MSE

- On the validation sample, however, the MSE will usually go down as the model complexity increases only up to a point.
- Then it will start to go up.
- It goes up because over-fitting of the training data begins to occur when the model complexity (think # of parameters) gets sufficiently high.

Very Important Idea!

- The idea that the MSE will decrease but then increase on the validation data as model complexity increases even though it will continue to decrease on the training data is a VERY important idea.

Notes:

- "Statistics" (the field) has developed "goodness-of-fit" methods that try to prevent over-fitting on the training data (all done "within sample").
- An exampled I expect you know about is the adjusted R^2 in regression.
- The adjusted R^2 can be used to choose between regression models with different sets of X-variables.

Notes (con't)

- Another measure of fit that you may have seen is the AIC (smaller AIC is better). There is a "corrected" version called the AICc which works better in small samples.
- Like the adjusted R^2, the AIC can be used to choose between models with different sets of *X*-variables.

Notes (con't)

- Having a randomly drawn validation sample, however, is a direct test of over-fitting.
- When you have enough data, using a validation sample is better.
- But this does NOT mean that measures such as the adjusted R^2, the AIC, or the AICc are not useful.

Overfitting

- When a model is fit to a dataset that has sufficient flexibility ("complexity") that it begins to fit random features in the data, it is said to have over-fit the data.
- The bias in the model will typically have gone down, but the variance will have increased because the model is fitting the "noise" in the data.

Overfitting (con't)

- So in terms of the bias vs. variance tradeoff, the bias will be low, but the variance will be high.
- Because the random errors in

$$Y = f(X) + \varepsilon$$

are being fit, the residuals will be unrealistically small.

Cross-Validation

- In statistics, methods that split the data into training and validation data sets go under the general name of "cross-validation."
- So, just splitting the data into a training set and a validation set (as we discussed before) is a form of cross validation.
- This is the simplest kind of cross-validation used in data science.

Reminder

- In Data Science, if we have enough data, we generally want to split our data into training, validation, and test samples.
- Typical split ratios would be 50%, 25%, and 25% or 60%, 20%, 20%.
- If you only have a relatively small amount of data (e.g., 300 observations), you might forgo the test sample and use a 2/3 to 1/3 split.

Cross-Validation (con't)

- The training data is used to estimate or fit a (relatively small) set of models.
  - For example, these models may be several linear regressions with different sets of *X*-variables or a collection of regression trees with different number of end nodes. Or you might compare trees to linear regression.
- Fitting means estimating the parameters of the models (e.g., the regression coefficients).

Cross-Validation (con't)

- The validation data set is used to pick which model is best (based on how it does predicting the *Y*-values for validation data).
- That is, the validation data set is used to determine the right level of model complexity (flexibility; e.g., number of *X*-variables in the model) or the right kind of model structure (tree vs. linear regression).

Cross-Validation (con't)

- Cross-validation generally prevents overfitting!
- Why? Random noise or random features in the training data set that are fit by a too-flexible model are not likely to also be the same in the validation data set.
- So an over-fitting model will generally perform badly for the validation data.

Cross-Validation (con't)

- Once the model has been selected, it can be re-fit to training + validation data.
- The test data set is used as a final check.

Summary: How to Think!

- The training data set is used to discover and fit a reasonably small number of models.
- These models should span a reasonable range of flexibility (complexity).
- The “deliverable” from the training data set is a collection of models (fit on the training data set) to test on the validation data set.

Summary: How to Think (con't)

- These models then "compete" by predicting the *Y*-variable on the validation data set (from the validation data set *X*-variables).
- Fitting (estimation of coefficients, etc.) is not done on the validation data – just application of the models.
- The validation data set picks the right model complexity (and structure).

Conclusion

- It is hard to overstate the importance of using cross-validation.
- The problem of overfitting is generally under-appreciated.
- Cross-validation is easy enough to do when you have enough data.

Conclusion (con't)

- There are other ways to do cross-validation (e.g., leave-one-out CV or k-fold CV).