Mohamad Lakkis

February, 2024

1 The Bias Variance Tradeoff

Conditional Expectation

In the following, note that $\hat{f}(x_0)$ is a *R.V.* since it depends on the Training which in its turn is a *R.V.*,as for $f(x_0)$ it is a constant since it doesn't depends on the training data. (note that in the following I am going to use $E \equiv E_{Y|X}$ and $f \equiv f(x_0)$ and $\hat{f} \equiv \hat{f}(x_0)$)

$$E_{Y|X}[(f(x_0) - \hat{f}(x_0))^2]$$

$$= E[(f - E\hat{f} + E\hat{f} - \hat{f})^2]$$

$$= E[(f - E\hat{f})^2] + E[(E\hat{f} - \hat{f})^2] + 2E[(f - E\hat{f})(E\hat{f} - \hat{f})]$$

Now let's work with the last term, notice that$(f - E\hat{f})$ is just a constant so it can get out of the expectation ! And thus,

$$2E[(f - E\hat{f})(E\hat{f} - \hat{f})] = 2(f - E\hat{f})E[E\hat{f} - \hat{f}] = 2(f - E\hat{f})(E\hat{f} - E\hat{f}) = 0$$

Therefore,

$$E_{Y|X}[(f(x_0) - \hat{f}(x_0))^2] = Bias_{Y|X}(\hat{f}(x_0))^2 + Var_{Y|X}(\hat{f}(x_0))$$

Joint Expectation

Now let's dive into the second expectation!
In the following, note that $\hat{f}(x_0)$ is a *R.V.* since it depends on the Training which in its turn is a *R.V.*,as for $f(x_0)$ it is a constant since it doesn't depends on the training data. (note that in the following I am going to use $E \equiv E_{Y,X}$ and $f \equiv f(x_0)$ and $\hat{f} \equiv \hat{f}(x_0)$)

$$E_{Y,X}[(f(x_0) - \hat{f}(x_0))^2]$$

$$= E[(f - E\hat{f} + E\hat{f} - \hat{f})^2]$$

$$= E[(f - E\hat{f})^2] + E[(E\hat{f} - \hat{f})^2] + 2E[(f - E\hat{f})(E\hat{f} - \hat{f})]$$

Now let's work with the last term notice that$(f - E\hat{f})$ is just a constant so it can get out of the expectation ! And thus,

$$2E[(f - E\hat{f})(E\hat{f} - \hat{f})] = 2(f - E\hat{f})E[E\hat{f} - \hat{f}] = 2(f - E\hat{f})(E\hat{f} - E\hat{f}) = 0$$

Therefore,

$$E_{Y,X}[(f(x_0) - \hat{f}(x_0))^2] = Bias_{Y,X}(\hat{f}(x_0))^2 + Var_{Y,X}(\hat{f}(x_0))$$

Conclusion

While ideally one would wish to reduce the estimation error by reducing both the bias and the variance of the predictor $\hat{f}$, this is in general not possible. Predictors that are too simple tend to have higher bias and to underfit, while those that are too complex tend to have high variance and tend to overfit. And this is what we call the Bias-Var Tradeoff ! (quoting the notes)

Additional

KNN: So in other words we need to relate $l_i(x_0, X)$ to the general function of KNN
We know that for KNN,

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_k(x_0)} y_i$$

with x_0 is an observation (might be in training set or test set or actual data that we need to predict), and $N_k(x_0)$ is the set of the K closest points to x_0
And as for the model in this question,

$$\hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0; X) y_i$$

And as a follow up,

$$l_i(x_0, X) = \begin{cases} \frac{1}{K} & \text{, if } x_i \in N_K(x_0), \\ 0 & \text{otherwise.} \end{cases}$$

So, for each x_0, only the K closest points to x_0 will have non-zero weights, and each of these weights will be $\frac{1}{k}$ (which contribute equally to the average)

Linear Model: The estimator is of the form $\hat{f}(x_0) = x_0^T \hat{\beta}$, with $\hat{\beta}$ is a vector(a column of size (p+1)×1 to be precise) of parameters estimated from the data and plugging $\hat{\beta} = (X^T X)^{-1} X^T Y$, and thus,

$$\hat{f}(x_0) = x_0^T (X^T X)^{-1} X^T y$$

And as a follow since $\hat{\beta}$ is linear in y, which is trivial to see, we get that for each y_i, we have,

$$l_i(x_0; X) = (1, x_0)(X^T X)^{-1}(1, x_i)^T$$

Notice that $l_i(x_0; X)$ is in scalar form since the result of the following will result in 1×1, matrix, which is simply one value, which is consistent since we y_i is also one value, and as a follow up of course $\hat{f}(x_0)$ should be one value, and thus each of the y_i is weighted by the correct weights.