CSC311: Introduction to Machine Learning

Tutorial 4

Bias Variance Decomposition

1 Towering Property of Expectation

To understand the calculations leading to bias-variance decomposition we need a property of expectations called the towering property of expectation:
Suppose X, Y are random variables(or vectors) with joint distribution $p(x, y)$, then the following holds

$$\mathbb{E}_X[\mathbb{E}_Y[A(X,Y)|X]] = E_{X,Y}[A(X,Y)]. \tag{1.1}$$

Proof. First we use the density function to calculate the simple expectation in left-hand side.

$$E_{X,Y}[A(X,Y)] = \int\int A(x,y)p(X=x, Y=y)dydx$$

To calculate the right-hand side we start with the expression inside the expectation.

$$\mathbb{E}_Y[A(X,Y)|X] = \int A(x,y)p(Y=y|X=x)dy$$

Now taking expectation with respect to X:

$$\begin{aligned}\mathbb{E}_X[\mathbb{E}_Y[A(X,Y)|X]] = \mathbb{E}_X[\int A(x,y)p(Y=y|X=x)dy] = \int(\int A(x,y)p(Y=y|X=x)dy)p(X=x)dx \\ = \int\int A(x,y)p(Y=y|X=x)p(X=x)dydx = \int\int A(x,y)p(X=x,Y=y)dydx\end{aligned}$$

Since both sides of 1.1 are simplified to the same integral, the equality is proved. □

2 Deterministic Setting

Suppose x comes from the distribution p and our target value y, is determined by x i.e. $y = f(x)$ and we have data-set $\mathcal{D} = \{(x_i, f(x_i)\}$ such that x_i's are i.i.d samples from p. Using $\mathcal{D}$, we train a model $h_\mathcal{D}$ to estimate f. Consider the following expression:

$$\mathbb{E}_{x,\mathcal{D}}[(h_\mathcal{D}(x) - f(x))^2]$$

Notice that both data-set and x are random. This expression calculates the expected error with respect to both x and $\mathcal{D}$. Using the towering property we can rewrite it as:

$$\mathbb{E}_\mathcal{D}[\mathbb{E}_x[(h_\mathcal{D}(x) - f(x))^2|\mathcal{D}]]$$

We can interpret this equation more intuitively. The expression inside the expectation

$$\mathbb{E}_x[(h_\mathcal{D}(x) - f(x))^2|\mathcal{D}]$$

measures the average error when a data-set is fixed. The outer expectation, averages this error over all of the data-sets.
Another way of towering the expectations is the following:

$$\mathbb{E}_x\big[\mathbb{E}_{\mathcal{D}}[(h_{\mathcal{D}}(x)-f(x))^2|x]\big]$$

This is a little harder to interpret but the same quantity. First it fixes one x and calculates the error for that specific x over models trained with different data-sets. Then, averages these errors for all of the x. This is the towering we will use for proving decomposition.
The trick to decompose this error (and the error in other cases) is to add and subtract a well-chosen value and show that the cross term is zero(using towering property). In here we need to add and subtract $\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|X]$.

$$\mathbb{E}_x\big[\mathbb{E}_{\mathcal{D}}[(h_{\mathcal{D}}(x)-f(x))^2|x]\big] = \mathbb{E}_x\big[\mathbb{E}_{\mathcal{D}}\big[(h_{\mathcal{D}}(x)-\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]+\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))^2|x\big]\big]$$
$$= \mathbb{E}_x\Bigg[\mathbb{E}_{\mathcal{D}}\big[(h_{\mathcal{D}}(x)-\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x])^2|x\big]+\mathbb{E}_{\mathcal{D}}\big[(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))^2|x\big]+$$
$$\mathbb{E}_{\mathcal{D}}[2(h_{\mathcal{D}}(x)-\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x])(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))|x]\Bigg].$$

Since $(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))$ does not depend on $\mathcal{D}$ (because of expectation) we can write:

$$\mathbb{E}_{\mathcal{D}}[2(h_{\mathcal{D}}(x)-\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x])(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))|x]] = 2\big(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x)\big)\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)-\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]|x].$$

But $\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)-\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]|x]=0$. So:

$$\mathbb{E}_{\mathcal{D}}[2(h_{\mathcal{D}}(x)-\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x])(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))|x]=0.$$

Again, Since $(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))$ does not depend on $\mathcal{D}$:

$$\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))^2|x] = (\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))^2$$

Plugging all of these back into original decomposition we get:

$$\mathbb{E}_{x,\mathcal{D}}[(h_{\mathcal{D}}(x)-f(x))^2] = \mathbb{E}_{x,\mathcal{D}}[(h_{\mathcal{D}}(x)-\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x])^2] + \mathbb{E}_x[(\mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(x)|x]-f(x))^2] \tag{2.1}$$

The first term in right-hand side is called the variance and it measures given different data-set how much the prediction varies. The second term is called the bias and measures on average how much the prediction is off from the real target.

3 Bayes Error

In previous section we made the assumption $y=f(x)$. In other words, we assumed that x determines y. The more realistic setting is to take y to be random, even if x is given. Before going into learning setting, let's see what is the best that can be done given that $y|x$ is not deterministic anymore. Suppose we have a function h, used for prediction, again we measure the mean squared error.

$$\mathbb{E}_{x,y}[(h(x)-y)^2] = \mathbb{E}_x[\mathbb{E}_y[(h(x)-y)^2|x]]$$

Again adding and subtracting helps:

$$\mathbb{E}_x[\mathbb{E}_y[(h(x)-y)^2|x]] = \mathbb{E}_x[\mathbb{E}_y[(h(x)-\mathbb{E}_y[y|x]+\mathbb{E}_y[y|x]-y)^2|x]]$$
$$= \mathbb{E}_x[\mathbb{E}_y[(h(x)-\mathbb{E}_y[y|x])^2|x]+\mathbb{E}_y[2(h(x)-\mathbb{E}_y[y|x])(\mathbb{E}_y[y|x]-y)|x]+\mathbb{E}_y[(\mathbb{E}_y[y|x]-y)^2|x]]$$

Like previous part we use two facts to show that cross term is zero. (i) $h(x) - \mathbb{E}_y[y|x]$ does not depend on y. (ii) $\mathbb{E}_y[(\mathbb{E}_y[y|x] - y)|x] = 0$. So we get:

$$\mathbb{E}_x[\mathbb{E}_y[(h(x) - y)^2|x]] = \mathbb{E}_x[(h(x) - \mathbb{E}_y[y|x])^2] + \mathbb{E}_{x,y}[(\mathbb{E}_y[y|x] - y)^2] \tag{3.1}$$

The right-hand side is sum of two positive numbers. The second term does not depend on h, So it can not be cancelled. This term is called the Bayes error and it is due to the unpredictability of target values. The first term, however, depend on h and if we set $h(x) = \mathbb{E}_y[y|x]$, this term becomes zero. This is the best any learning algorithm can do. An algorithm that achieves this is called Bayes optimal.

4 Non-deterministic learning setting

Now to combine the results from previous sections, suppose we have a distribution that generates (x, y) and we are trying to train an algorithm to predict y from x but this time $y|x$ is not deterministic. Notice it is not always possible to find Bayes-optimal solution because we do not have access to the distribution (only samples) so we do not know $\mathbb{E}_y[y|x]$. Again we have a data-set $\mathcal{D} = \{(x_i, y_i)\}$ such that (x_i, y_i)'s are i.i.d samples from distribution. Once again we measure the mean squared error.

$$\mathbb{E}_{x,y,\mathcal{D}}[(h_\mathcal{D}(x) - y)^2] = \mathbb{E}_\mathcal{D}[\mathbb{E}_{x,y}[(h_\mathcal{D}(x) - y)^2|\mathcal{D}]]$$

Consider the expression inside the expectation. Since given $\mathcal{D}$, $h_\mathcal{D}$ is determined we can treat the expression inside the expectation, like 3.1, which gives

$$\mathbb{E}_\mathcal{D}[\mathbb{E}_{x,y}[(h_\mathcal{D}(x) - y)^2|\mathcal{D}]] = \mathbb{E}_\mathcal{D}[\mathbb{E}_x[(h_\mathcal{D}(x) - \mathbb{E}_y[y|x])^2|\mathcal{D}] + \mathbb{E}_{x,y}[(\mathbb{E}_y[y|x] - y)^2|\mathcal{D}]]$$

Since $\mathbb{E}_y[y|x] - y$ does not depend on $\mathcal{D}$ and $h_\mathcal{D}(x) - \mathbb{E}_y[y|x]$ does not depend on y, we can write:

$$\mathbb{E}_{x,y,\mathcal{D}}[(h_\mathcal{D}(x) - y)^2] = \mathbb{E}_{x,\mathcal{D}}[(h_\mathcal{D}(x) - \mathbb{E}_y[y|x])^2] + \mathbb{E}_{x,y}[(\mathbb{E}_y[y|x] - y)^2]$$

The last term is Bayes error. In the first term we have $\mathbb{E}_y[y|x]$ and it depends on x deterministically, so just like 2.1 we can decompose it.

$$\mathbb{E}_{x,\mathcal{D}}[(h_\mathcal{D}(x) - \mathbb{E}_y[y|x])^2] = \mathbb{E}_{x,\mathcal{D}}[(h_\mathcal{D}(x) - \mathbb{E}_\mathcal{D}[h_\mathcal{D}(x)|x])^2] + \mathbb{E}_x[(\mathbb{E}_\mathcal{D}[h_\mathcal{D}(x)|x] - \mathbb{E}_y[y|x])^2]$$

Putting all of these terms together we get:

$$\mathbb{E}_{x,y,\mathcal{D}}[(h_\mathcal{D}(x) - y)^2] = \mathbb{E}_{x,\mathcal{D}}[(h_\mathcal{D}(x) - \mathbb{E}_\mathcal{D}[h_\mathcal{D}(x)|x])^2] + \mathbb{E}_x[(\mathbb{E}_\mathcal{D}[h_\mathcal{D}(x)|x] - \mathbb{E}_y[y|x])^2] + \mathbb{E}_{x,y}[(\mathbb{E}_y[y|x] - y)^2] \tag{4.1}$$

The terms on the right-hand side are called Variance, Bias and Bayes error respectively.

5 Effects of Bagging on Error

Suppose we have m independent data-sets $\mathcal{D}_i$. We train a predictor h_i using each data-set $\mathcal{D}_i$. Then at test time we report $h(x) = \frac{1}{m}\sum h_i(x)$ as our prediction. This procedure is called bagging. Although in real life we do not have access to independent data-sets so for each $\mathcal{D}_i$ we sample from original training set. Using the fact that $\mathcal{D}_i$'s are i.i.d, lets compute the average and variance of our predictor.

$$\mathbb{E}[h(x)] = \mathbb{E}[\frac{1}{m}\sum h_i(x)] = \frac{1}{m}\sum \mathbb{E}[h_i(x)] = \mathbb{E}[h_0(x)]$$

So the average does not change.

$$\mathrm{Var}(h(X)) = \mathrm{Var}\left(\frac{1}{m}\sum h_i(x)\right) = \frac{1}{m^2}\sum \mathrm{Var}(h_i(x)) = \frac{1}{m}\mathrm{Var}(h_0(x))$$

So the variance reduces.
In the decomposition 4.1, The Bayes error does not depend on the model. Since the average does not change the bias stays the same. So, the only part that is changed with bagging is the variance part, which is divided to m and this is the part that will reduce the error of the model.

6 Methods for Splitting in Decision Tree

As you have seen finding the smallest tree is NP-complete so we use a greedy approach. We take a function like f, that takes all the samples in a node and gives a number that show how "bad" that node is. Using f we determine which attribute is "best" for splitting.
Suppose we have n samples in node A and we split based on some attribute a, and get two nodes B and C such that n_B samples go into node B and n_C samples go into node C. We compute the following value that shows how much we gained in terms of f

$$S(a) = f(A) - (\frac{n_B}{n} f(B) + \frac{n_C}{n} f(C))$$

We do this for all of the splittings, meaning compute $S(a)$ for all attributes, and then select the splitting with the maximum value.
For example if we set f to be the entropy of the samples in the node, then $S(a)$ is information gain.
Decision trees can be used to predict continuous variables as well, in that case the predicted values for leaves are typically the average of samples in each leaf (While for discrete case it was majority of the leaf). The criterion for splitting is MSE i.e. we set $f = \frac{1}{n}\sum(y_i - \bar{y})^2$ where $\bar{y} = \frac{1}{n}\sum y_i$, and y_i's are the target values for the samples in node.