Bias-Variance Decomposition

Pramook Khungurn

January 6, 2018

Decomposing the expected error of a machine learning model into bias, variance, and noise terms is an important tool for understanding and designing machine learning models. I learned about the decomposition when I took a machine learning class at Cornell. However, the class only teaches the decomposition as it applies to regression and the squared loss. It was unclear to me how the same concept would apply to classification and other loss functions.

To fill this gap of understanding, this document will address two papers that deal with this question. The first is the 2000 paper by Pedro Domingos [Domingos, 2000], and the second by is the 2003 paper by James [James, 2003].

1 Domingos 2000

1.1 Settings and Notations

- Let the training set be denoted by $D = \{(x_1, t_1), (x_2, t_2), \ldots, (x_n, t_n)\}$.
- Let f be the model produced by training on D. Note that f is a function of D.
- Let $y = f(x)$ denote the prodiction made by the model on input x.
- Let t denote the true value of the prediction of x.
  - In general, t is a non-deterministic function of x.
  - In other words, there is a probability distribution of (x, t) pairs where there are pairs with the same value of x but different values of t.
- Let $L(t, y)$ denote the loss function that measures the cost of predicting y while the true value is t.
- Some commonly used loss functions are:
  - **Squared loss:** $L(t, y) = (t - y)^2$.
  - **Absolute loss:** $L(t, y) = |t - y|$.
  - **Zero-one loss:** $L(t, y) = \begin{cases} 0, & t = y \\ 1, & t \neq y \end{cases}$.
- The *optimal prediction* y_* for an input x is the prediction that minimizes $E_t[L(t, y_*)]$.
- The *optimal model* is the model such that $f(x) = y_*$ for al x.
  - In case of classification, the optimal model is called the *Bayes classifier*.
  - The expected loss of the Bayes classifier is called the *Bayes rate*.

- The model f produced depends on the training set D. As a result, $L(t, y)$ is a function of D. To get a big picture of the loss, though, we should eliminate the dependence on D. We do this by averaging out the loss over all the possible training sets. As a result, we are interested in the *expected loss:*

$$E_{D,t}[L(t, y)]$$

  Here, the expectation is taken with respect to the training set D and the true value t. Note that this is a function of x, so t is conditioned on x.

- The bias-variance decomposition splits the expected loss into three different terms: *the bias*, *the variance*, and *the noise*.

1.2 Main Definitions

- **Definition 1.** *The* main prediction *for a particular input value x is:*

$$y_m = \operatorname*{argmin}_{y'} E_D[L(y, y')].$$

  In other words, the main prediction is the prediction y' whose average loss over all possible predictions is minimal. In other words,

  - The main prediction under the square loss is the mean of all possible predictions.
  - The main prediction under the absolute loss is the median of all possible predictions.
  - The main prediction under the zero-one loss is the mode of all possible predictions.

- **Definition 2.** *The* bias *of the learning algorithm on a particular example x is:*

$$B(x) = L(y_*, y_m).$$

  In other words, it is the loss incurred by the main prediction relative to the optimal prediction.

- **Definition 3.** *The* variance *of the learning algorithm on a particular example x is:*

$$V(x) = E_D[L(y_m, y)].$$

  That is, is the average loss incurrred by all predictions relative to the main prediction.

- Bias and variance may be averaged over all possible input data. The averages are called the *expected bias* $E_x[B(x)]$ and the *expected variance* $E_x[V(x)]$, respectively.

- **Definition 4.** *The* noise *of a particular example x is given by:*

$$N(x) = E_t[L(t, y_*)].$$

  Note that the definition of the noise does not contain any terms that rely on the training set. Hence, it is a feature of the dataset and is independent of the learning algorithm used.

- Nice properties of the above definitions:

  - The main prediction y_m defines the "center" of the possible predictions.
  - The bias measures how the center is off from the optimal prediction. Hence it measure the systematic (deterministic) loss of the learning algorithm.
  - The variance measures how the possible predictions are scattered around the center of prediction. Hence, it measures the stochastic component of the expected loss.

- If the loss function is non-negative, then all the three terms are non-negative.
- The bias is independent of the training set and is zero if the learner always output the optimal prediction.
- The variance is independent of the true prediction value and is zero for the learner that always output the same prediction regardless of the training set.
- All the terms only requires that the expectation be computable to be well defined.

1.3 The Main Result

- It is not true, however, that the expected loss can be written as the sum of the three terms for all loss function L.
- The following, though, is true:

  Claim 5. *For certain loss functions L, we have:*

  $$E_{D,t}[L(y,t)] = c_1 N(x) + B(x) + c_2 V(x) \tag{1}$$

  where c_1 and c_2 are multiplicative factors that will take on different values for different loss functions.

- For the squared loss, $c_1 = c_2 = 1$.
- **Theorem 6.** *In two-class problems, Claim 5 is valid for any real-valued loss function L for which:*

  1. *$L(y,y) = 0$ for all y.*
  2. *$L(y_1, y_2) \neq 0$ for all $y_1 \neq y_2$.*

  For such a function, it is true that:

  $$c_1 = P_D(y = y_*) - \frac{L(y_*, y)}{L(y, y_*)} P_D(y \neq y_*).$$

  Moreover,

  $$c_2 = \begin{cases} 1, & y_* = y_m \\ -L(y_*, y_m)/L(y_m, y_*) & y_* \neq y_m \end{cases}.$$

  Proof. We first show that:

  $$L(t,y) = L(y_*, y) + c_0 L(t, y_*) \tag{2}$$

  where

  $$c_0 = \begin{cases} 1, & y = y_* \\ -L(y_*, y)/L(y, y*), & y \neq y_* \end{cases}.$$

  To see that (2) is true, we consider the two cases where $y = y_*$ and where $y \neq y_*$. The first case is obvious. For the second case, we have that either $t = y$ or $t = y*$ because we have a two-class problem. If $t = y$, then

  $$0 = L(t,y) = L(y_*, y) - \frac{L(y_*, y)}{L(y, y*)} L(y, y_*),$$

which confirms that (2) is true. If $t = y*$, then

$$L(t, y) = L(y*, y) = L(y_*, y) - \frac{L(y_*, y)}{L(y, y*)} L(y^*, y^*),$$

which also confirms that (2) is also true in this case. As a result, we can conclude that (2) is true in all cases.

In a similar manner that we proved (2), we can also show that:

$$L(y_*, y) = L(y_*, y_m) + c_2 L(y_m, y) \tag{3}$$

where c_2 is defined in the Theorem's statement. Again, there are two cases. First, when $y_m = y_*$, we have that (3) becomes:

$$L(y_m, y) = L(y_m, y_m) + 1 \cdot L(y_m, y),$$

which is trivially true. Second, when $y_m \neq y_*$. Then, either $y = y_m$ or $y = y_*$. If $y = y_m$, then (3) reduces to:

$$L(y_*, y_m) = L(y_*, y_m) - \frac{L(y_*, y_m)}{L(y_m, y_*)} L(y_m, y_m),$$

which is true. If $y = y_*$, then (3) reduces to:

$$L(y_*, y_*) = L(y_*, y_m) - \frac{L(y_*, y_m)}{L(y_m, y_*)} L(y_m, y_*),$$

which is also true as well. As a result, we can say that (3) is true in all cases.

Using (2), we have that:

$$\begin{aligned} E_{D,t}[L(t, y)] &= E_D[E_t[L(t, y)]] \\ &= E_D[E_t[L(y_*, y) + c_0 L(t, y_*)]] \\ &= E_D[L(y_*, y) + c_0 E_t[L(t, y_*)]]] \\ &= E_D[L(y_*, y)] + E_D[c_0] E_t[L(t, y_*)]]]. \end{aligned}$$

The next to last line comes from the fact that $L(y_*, y)$ and c_0 does not depend on t. The last line comes from the fact that y_* and t does not depends on the training dataset D.

Substituting (3), we have that:

$$\begin{aligned} E_{D,t}[L(t, y)] &= E_D[L(y_*, y_m) + c_2 L(y_m, y)] + E_D[c_0] E_t[L(t, y_*)]]] \\ &= L(y_*, y_m) + c_2 E_D[L(y_m, y)] + E_D[c_0] E_t[L(t, y_*)] \\ &= B(x) + c_2 V(x) + E_D[c_0] N(x). \end{aligned}$$

Now,

$$E_D[c_0] = P_D(y = y_*) - \frac{L(y_*, y)}{L(y, y*)} P_D(y \neq y_*) = c_1.$$

As a result, we have that:

$$E_{D,t}[L(t, y)] = c_1 N(x) + B(x) + c_2 V(x)$$

as desired. □

- If the loss function is symmetric (for example, the zero-one loss), then we have that:

$$c_1 = 2P_D(y = y_*) - 1,$$
$$c_2 = \begin{cases} 1, & y_* = y_m \\ -1, & y_* \neq y_m \end{cases}.$$

- To rephrase the item above, *variance is additive in unbiased examples and subtractive in biased examples* in binary classification problems with symmetric loss functions.
  - It is easy to understand why variance is beneficial in bias examples: if the main prediction is not optimal, it is better to deviate from it.
  - Consequently, highly unstable learners like decision tree and rule induction algorithms can produce good results in practice because it has a good chance to deviate from biased predictions.
  - Zero-one loss has higher tolerance for variance: variance on unbiased examples is offset by the variance on biased ones.
- Now, consider the c_1 factor:
  - In the squared loss, increasing noise always increases the expected error.
  - However, in the zero-one loss, c_1 can be negative in examples where $P_D(y = y_*) < 0.5$. In this case, increasing the noise decreases the error.
- Loss in general can be asymmetric. (There are cases where false negatives are more costly then false positives and vice versa.) So, the general case (i.e., Theorem 6) is important as well.
- **Theorem 7.** *For multi-class problems under the zero-one loss, Equation* (1) *holds with:*

$$c_1 = P_D(y = y_*) - P_D(y \neq y_*)P_t(y = t|y_* \neq t),$$
$$c_2 = \begin{cases} 1, & y_m = y_* \\ -P_D(y = y_*|y \neq y_m), & y_m \neq y_* \end{cases}.$$

- According to the above theorem, not all variance on biased examples contributes to the reduction of the error. It is only those examples where $y = y_*$ that reduces the error.
  - Of course, only the optimal prediction would reduce error.

1.4 Properties of the Decomposition

1.4.1 Order Correctness

- Breiman observed that bagging reduces zero-one loss and explained the phenomenon with the concept of *order-correct* learner [Breiman, 1996].
- **Definition 8.** *A learner is* order-correct *on an example x if:*

$$\forall_{y \neq y_*}, P_D(y) < P_D(y_*)$$

  In other words, the learner has higher probability to arrive at the optimal prediction than any other predictions.
- Breiman proved that bagging transforms an order-correct learner into a nearly optimal one.
- **Claim 9.** *A learner is order-correct on x if and only if $B(x) = 0$ under the zero-one loss.*

1.4.2 Margin

- Schapire et al.explained why boosting works with the notion of *margin* [Schapire et al., 1997].
- **Definition 10.** *In two-class problems, the margin of a learner on an example x is given by:*

$$M(x) = P_D(y = t) - P_D(y \neq t)$$

The set D here is a generalized version of the sample space of all possible training sets. It now means the set of training sets that the learner is applied. In other words, if boosting is conducted for 100 rounds, then $|D| = 100$. Moreover, different training sets have different weights, which can be made to sum up to 1. This naturallly induces a probability distribution over the training set.

- A large margin corresponds to a high confidence in the prediction.
- Schapire et al. showed that, the lower the probability of a low margin, the lower the bound on generalization error.

1.4.3 Relationship between Order Correctness and Margin

- **Theorem 11.** *The margin of a learner on an example x can be expressed in terms of its zero-one bias and variance as:*

$$M(x) = \pm[2B(x) - 1][2V(x) - 1]$$

with positive sign if $y_* = t$ and negative sign otherwise.

Proof. When $y_* = t$, we have that:

$$M(x) = P_D(y = y_*) - P_D(y \neq y_*) = 2P_D(y = y_*) - 1.$$

If $B(x) = 0$, then $y_m = y_*$. So,

$$M(x) = 2P_D(y = y_m) - 1 = 2(1 - V(x)) - 1 = -(2V(x) - 1).$$

If $B(x) = 1$, then

$$M(x) = 2V(x) - 1.$$

In both case, it is true that

$$M(x) = [2B(x) - 1][2V(x) - 1].$$

The case for when $y_* \neq t$ is similar. □

- It is also possible to express the bias and the variance in terms of the margin:

$$B(x) = \frac{1}{2}\big(1 \pm \text{sgn}(M(x))\big)$$
$$V(x) = \frac{1}{2}\big(1 \pm |M(x)|\big)$$

The $\pm$ is positive if $y_* \neq t$ and negative otherwise.

- Since margin can be written in terms of bias and variance, Schapire et al.'s treatment and Breiman's treatment is actually the same thing.

- Bias and variance explanation is difficult to apply to boosting because they are multipled together. That is, suppose that $y_* = t$ and we want to increase the margin. Then, we would like to increase variance on biased examples and decrease variance on unbiased ones. (If $y_* \neq t$, it's the other way around.)
- All in all, I think Domingos's treatment works because we have a binary classification problem, and the loss function is the zero-one loss. It does not really generalize to other lossses. Moreover, the theorems look kind of ugly.

References

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[Domingos, 2000] Domingos, P. (2000). A unifeid bias-variance decomposition and its applications. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 231–238, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[James, 2003] James, G. M. (2003). Variance and bias for general loss functions. *Machine Learning*, 51(2):115–135.

[Schapire et al., 1997] Schapire, R. E., Freund, Y., Barlett, P., and Lee, W. S. (1997). Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 322–330, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.