Noname manuscript No.
(will be inserted by the editor)

Bias and Variance in Software Effort Estimation

An Investigation of Bias-Variance Trade-Off Subject to Different Testing Strategies

the date of receipt and acceptance should be inserted later

A typical dataset consists of a a matrix X and a vector Y. The input variables (a.k.a. features) are stored in X, where each row corresponds to an observation and each column corresponds to a particular variable. Similarly, the dependent variable is stored in a vector Y, where for each observation in X there exists a response value.

Now assume that a prediction model represented by $\hat{f}(x)$ has been learned from a training dataset τ. So as to measure the errors between the actual values in Y and the predictions given by $\hat{f}(x)$, we can make use of an error function represented by $L(Y, \hat{f}(x))$. Some examples of error functions are squared loss (given in Equation 1) or absolute loss (given in Equation 2).

$$L(Y, \hat{f}(x)) = \left(Y - \hat{f}(x)\right)^2 \tag{1}$$

$$L(Y, \hat{f}(x)) = |Y - \hat{f}(x)| \tag{2}$$

Given the assumptions that the underlying model is $Y = f(X) + \epsilon$ where $E(\epsilon) = 0$ and $Var(\epsilon) = \sigma_\epsilon^2$, then we can come up with a derivation of the squared-error loss for $\hat{f}(X)$ [1]. The error for a point $X = x_0$ is:

$$\begin{aligned} Error(x_0) &= E\left[\left(Y - \hat{f}(x_0)\right)^2 | X = x_0\right] \\ &= \sigma_\epsilon^2 + \left(E[\hat{f}(x_0) - f(x_0)]\right)^2 + E\left[\hat{f}(x_0) - E[\hat{f}(x_0)]\right] \\ &= \sigma_\epsilon^2 + Bias^2(\hat{f}(x_0)) + Var(\hat{f}(x_0)) \\ &= \underbrace{IrreducableError}_{1^{st}Term} + \underbrace{Bias^2}_{2^{nd}Term} + \underbrace{Variance}_{3^{rd}Term} \end{aligned}$$

In the above derivation, the explanations of the 1^{st}, 2^{nd} and 3^{rd} terms are as follows:

- The $1^{st}Term$ is the so called *"irreducable error"*, i.e. the variance of the actual model around its true mean. This variance is inevitable regardless of how well we model $f(x_0)$, only exception to that is when the actual variance is zero (when $\sigma_\epsilon^2 = 0$).

Address(es) of author(s) should be given

– The $2^{nd}Term$ is the square of the bias, which is the measure of how different the model estimates are fromt the *true* mean of the underlying model.
– The $3^{rd}Term$ is the variance of the estimated model. It is the expectation of the squared deviation of the estimated model from its own mean.

Furthermore, the above derivation is for an individual instance. The bias and variance values associated with an algorithm $\hat{f}(X)$ is the mean of all individual values.

Then the question becomes how the bias and variance (from now on $B\&V$) relate to different choices of the training size (*K*), i.e. the relation to cross-validation method (CV). Here we will consider two cases of CV: leave-one-out (LOO) and 3-Way. Ideally when training size is equal to the dataset size (*K=N*), we expect CV to be approximately unbiased and to have high variance, because N training sets are so similar to one another. On the other hand, for small values of *K*, say *K=N/3* as in 3-Way, we expect lower variance and a higher bias [1]. Naively put, the relationship is:

– LOO : Higher variance, lower bias
– 3-Way : Lower variance, higher bias

In an ideal case, when we plot $B\&V$ values of each individual test instances on x and y axes respectively, we expect 2 clusters:

– Upper Left: Low bias, high variance; i.e. LOO results.
– Lower right: High bias, low variance; i.e. 3Way results.

Just for the sake of clarity, a very *simple* but *ideal* case would look like Figure 1. In that figure, 30 hypothetical algorithms subject to both LOO and 3-Way are represented.

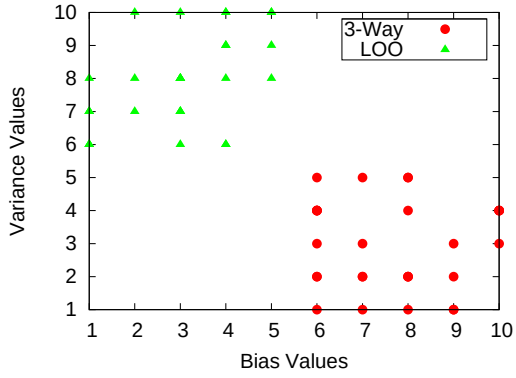

Fig. 1 A simple simulation for the ideal case of $B\&V$ relation to testing strategies.

When we calculated the $B\&V$ values for 90 algorithms (the algorithms in Comba paper) on various datasets, we were unable to observe the behavior of Figure 1, i.e. we did not observe two distinct clusters at predicted $B\&V$ zones. On the contrary, we observed that both $B\&V$ values are close to one another for LOO and 3Way, i.e. the two clusters mostly overlap. Also, the *ideal* or *predicted* lowness and highness for $B\&V$ values were not visible too. The actual $B\&V$ values were both high, regardless of the testing strategy. In Figure 2, Figure 3, Figure 4 the $B\&V$ plots of 90 algorithms (i.e. 90 circles for 3-Way and 90 triangles

for LOO) for Nasa93, Cocomo81 and Desharnais datasets are to be seen. All the values reported in these figures are logged. Also note that the axes in these figures are not scaled, because the differences are so small that scaling the axes makes it difficult to observe the behavior of $B\&V$. See in these figures, how the *ideal* behavior of $B\&V$ differs from the *actual* case for software effort datasets. We have conducted these experiments on many more datasets and the results are pretty much the same: 1) No ideal behavior for 3-Way and LOO; 2) 3-Way and LOO $B\&V$ values overlap.

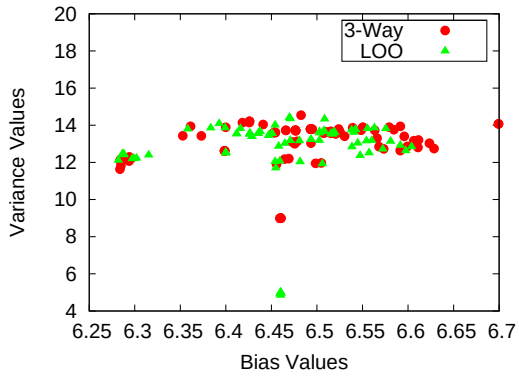

Fig. 2 $B\&V$ values for Nasa93.

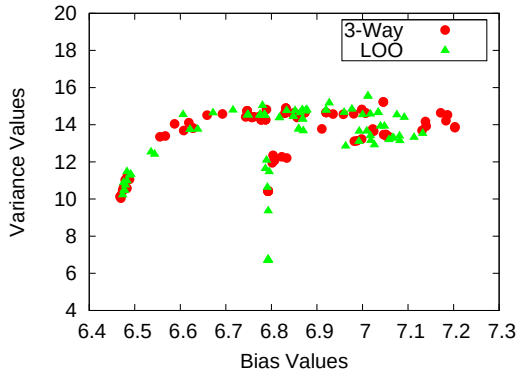

Fig. 3 $B\&V$ values for Cocomo81.

References

1. *The Elements of Statistical Learning*. Springer, July 2003.

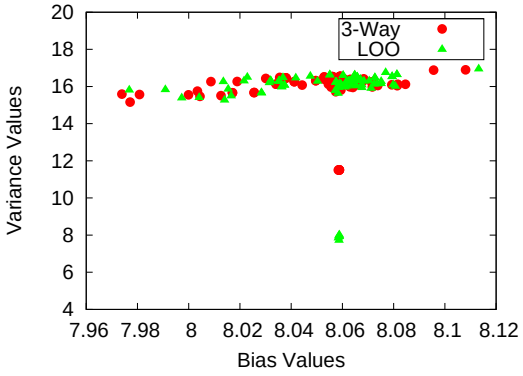

Fig. 4 $B\&V$ values for Desharnais.