Bias–variance tradeoff

From Wikipedia, the free encyclopedia

In statistics and machine learning, the **bias–variance tradeoff** (or **dilemma**) is the problem of simultaneously minimizing two sources of error that prevent supervised learning algorithms from generalizing beyond their training set:

- The *bias* is error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- The *variance* is error from sensitivity to small fluctuations in the training set. High variance can cause overfitting: modeling the random noise in the training data, rather than the intended outputs.

The **bias–variance decomposition** is a way of analyzing a learning algorithm's expected generalization error with respect to a particular problem as a sum of three terms, the bias, variance, and a quantity called the *irreducible error*, resulting from noise in the problem itself.

This tradeoff applies to all forms of supervised learning: classification, regression (function fitting),[1][2] and structured output learning. It has also been invoked to explain the effectiveness of heuristics in human learning.

Contents

Motivation

The bias–variance tradeoff is a central problem in supervised learning. Ideally, one wants to choose a model that both accurately captures the regularities in its training data, but also generalizes well to unseen data. Unfortunately, it is typically impossible to do both simultaneously. High-variance learning methods may be able to represent their training set well, but are at risk of overfitting to noisy or unrepresentative training data. In contrast, algorithms with high bias typically produce simpler models that don't tend to overfit, but may *underfit* their training data, failing to capture important regularities.

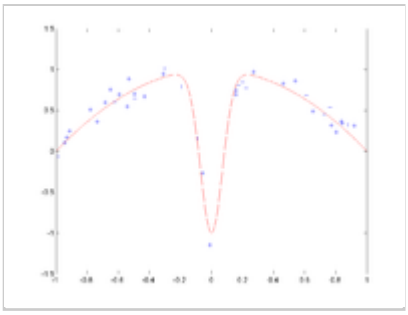
Models with low bias are usually more complex (e.g. higher-order regression polynomials), enabling them to represent the training set more accurately. In the process, however, they may also represent a large noise component in the training set, making their predictions less accurate - despite their added complexity. In contrast, models with higher bias tend to be relatively simple (low-order or even linear regression polynomials), but may produce lower variance predictions when applied beyond the training set.

Bias–variance decomposition of squared error

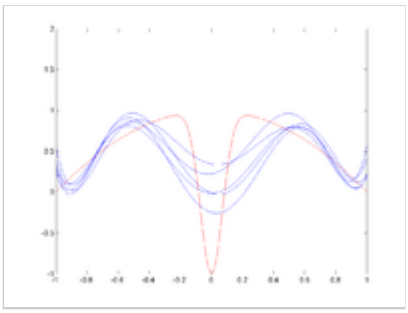
Suppose that we have a training set consisting of a set of points $x_1, \dots, x_n$ and real values y_i associated with each point x_i. We assume that there is a functional, but noisy relation $y_i = f(x_i) + \epsilon$, where the noise, ϵ, has zero mean and variance σ^2.

We want to find a function $\hat{f}(x)$, that approximates the true function $y = f(x)$ as well as possible, by means of some learning algorithm. We make "as well as possible" precise by measuring the mean squared error between y and $\hat{f}(x)$: we want $(y - \hat{f}(x))^2$ to be minimal, both for $x_1, \dots, x_n$ *and for points outside of our sample*. Of course, we cannot hope to do so perfectly, since y contains noise ϵ; this means we must be prepared to accept an *irreducible error* in any function we come up with.

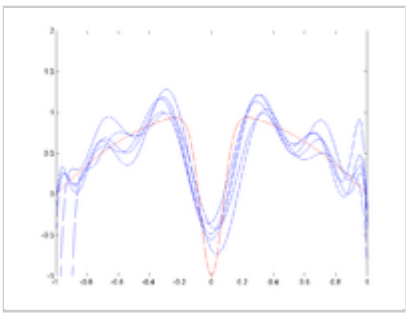
Finding an $\hat{f}$ that generalizes to points outside of the training set can be done with any of the countless algorithms used for supervised learning. It turns out that whichever function $\hat{f}$ we select, we can decompose its expected error on an unseen sample x as follows:[3]:34[4]:223



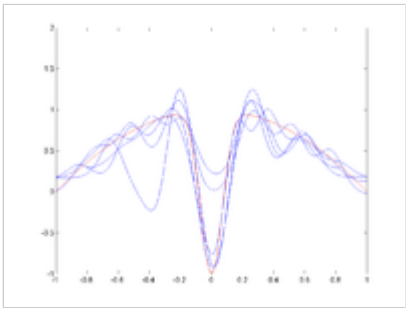
Function and noisy data.



spread=5



spread=1



spread=0.1

A function (red) is approximated using radial basis functions (blue). Several trials are shown in each graph. For each trial, a few noisy data points are provided as training set (top). For a wide spread (image 2) the bias is high: the RBFs cannot fully approximate the function (especially the central dip), but the variance between different trials is

low. As spread decreases (image 3 and 4) the bias decreases: the blue curves more closely approximate the red. However, depending on the noise in different trials the variance between trials increases. In the lowermost image the approximated values for x=0 varies wildly depending on where the data points were located.

$$\mathrm{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \mathrm{Bias}\left[\hat{f}(x)\right]^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$

Where:

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

and

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

The expectation ranges over different choices of the training set $x_1, \ldots, x_n, y_1, \ldots, y_n$, all sampled from the same distribution. The three terms represent:

- the square of the *bias* of the learning method, which can be thought of the error caused by the simplifying assumptions built into the method. E.g., when approximating a non-linear function $f(x)$ using a learning method for linear models, there will be error in the estimates $\hat{f}(x)$ due to this assumption;
- the *variance* of the learning method, or, intuitively, how much the learning method $\hat{f}(x)$ will move around its mean;
- the irreducible error σ^2. Since all three terms are non-negative, this forms a lower bound on the expected error on unseen samples.[3]:34

The more complex the model $\hat{f}(x)$ is, the more data points it will capture, and the lower the bias will be. However, complexity will make the model "move" more to capture the data points, and hence its variance will be larger.

Derivation

The derivation of the bias–variance decomposition for squared error proceeds as follows.[5][6] For notational convenience, abbreviate $f = f(x)$ and $\hat{f} = \hat{f}(x)$. First, note that for any random variable X, we have

$$\begin{aligned}\mathrm{E}[X^2] &= \mathrm{E}[X^2] - \mathrm{E}[2X\mathrm{E}[X]] + \mathrm{E}[\mathrm{E}[X]^2] + \mathrm{E}[2X\mathrm{E}[X]] - \mathrm{E}[\mathrm{E}[X]^2] \\ &= \mathrm{E}[X^2 - 2X\mathrm{E}[X] + \mathrm{E}[X]^2] + 2\mathrm{E}[X]^2 - \mathrm{E}[X]^2 \\ &= \mathrm{E}[(X - \mathrm{E}[X])^2] + \mathrm{E}[X]^2 \\ &= \mathrm{Var}[X] + \mathrm{E}[X]^2\end{aligned}$$

Since f is deterministic

$$0 = \mathrm{Var}[f] = \mathrm{E}[(f - \mathrm{E}[f])^2] \Rightarrow f - \mathrm{E}[f] = 0 \Rightarrow \mathrm{E}[f] = f.$$

This, given $y = f + \epsilon$ and $\mathrm{E}[\epsilon] = 0$, implies $\mathrm{E}[y] = \mathrm{E}[f + \epsilon] = \mathrm{E}[f] = f$.

Also, since $\mathrm{Var}[\epsilon] = \sigma^2$

$$\mathrm{Var}[y] = \mathrm{E}[(y - \mathrm{E}[y])^2] = \mathrm{E}[(y - f)^2] = \mathrm{E}[(f + \epsilon - f)^2] = \mathrm{E}[\epsilon^2] = \mathrm{Var}[\epsilon] +$$

Thus, since ϵ and $\hat{f}$ are independent, we can write

$$\begin{aligned}\mathrm{E}\left[(y - \hat{f})^2\right] &= \mathrm{E}[y^2 + \hat{f}^2 - 2y\hat{f}] \\ &= \mathrm{E}[y^2] + \mathrm{E}[\hat{f}^2] - \mathrm{E}[2y\hat{f}] \\ &= \mathrm{Var}[y] + \mathrm{E}[y]^2 + \mathrm{Var}[\hat{f}] + \mathrm{E}[\hat{f}]^2 - 2f\mathrm{E}[\hat{f}] \\ &= \mathrm{Var}[y] + \mathrm{Var}[\hat{f}] + (f - \mathrm{E}[\hat{f}])^2 \\ &= \sigma^2 + \mathrm{Var}[\hat{f}] + \mathrm{Bias}[\hat{f}]^2\end{aligned}$$

Q.E.D.

Application to classification

The bias–variance decomposition was originally formulated for least-squares regression. For the case of classification under the 0-1 loss (misclassification rate), it's possible to find a similar decomposition.[7][8] Alternatively, if the classification problem can be phrased as probabilistic classification, then the expected squared error of the predicted probabilities with respect to the true probabilities can be decomposed as before.[9]

Approaches

Dimensionality reduction and feature selection can decrease variance by simplifying models. Similarly, a larger training set tends to decrease variance. Adding features (predictors) tends to decrease bias, at the expense of introducing additional variance. Learning algorithms typically have some tunable parameters that control bias and variance, e.g.:

- (Generalized) linear models can be regularized to increase their bias.
- In artificial neural networks, the variance increases and the bias decreases with the number of hidden units.[1] Like in GLMs, regularization is typically applied.

- In k-nearest neighbor models, a high value of k leads to high bias and low variance (see below).
- In Instance-based learning, regularization can be achieved varying the mixture of prototypes and exemplars.[10]
- In decision trees, the depth of the tree determines the variance. Decision trees are commonly pruned to control variance.[3]:307

One way of resolving the trade-off is to use mixture models and ensemble learning.[11][12] For example, boosting combines many "weak" (high bias) models in an ensemble that has greater variance than the individual models, while bagging combines "strong" learners in a way that reduces their variance.

K-nearest neighbors

In the case of k-nearest neighbors regression, a closed-form expression exists that relates the bias–variance decomposition to the parameter k:[4]:37, 223

$$\mathrm{E}[(y - \hat{f}(x))^2] = \left(f(x) - \frac{1}{k}\sum_{i=1}^{k} f(N_i(x))\right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

where $N_1(x), \ldots, N_k(x)$ are the k nearest neighbors of x in the training set. The bias (first term) is a monotone rising function of k, while the variance (second term) drops off as k is increased. In fact, under "reasonable assumptions" the bias of the first-nearest neighbor (1-NN) estimator vanishes entirely as the size of the training set approaches infinity.[1]

Application to human learning

While widely discussed in the context of machine learning, the bias-variance dilemma has been examined in the context of human cognition, most notably by Gerd Gigerenzer and co-workers in the context of learned heuristics. They have argued (see references below) that the human brain resolves the dilemma in the case of the typically sparse, poorly-characterised training-sets provided by experience by adopting high-bias/low variance heuristics. This reflects the fact that a zero-bias approach has poor generalisability to new situations, and also unreasonably presumes precise knowledge of the true state of the world. The resulting heuristics are relatively simple, but produce better inferences in a wider variety of situations.[13]

Geman et al.[1] argue that that the bias-variance dilemma implies that abilities such as generic object recognition cannot be learned from scratch, but require a certain degree of “hard wiring” that is later tuned by experience. This is because model-free approaches to inference require impractically large training sets if they are to avoid high variance.

See also

- Bias of an estimator
- Gauss–Markov theorem
- Hyperparameter optimization

- Minimum-variance unbiased estimator
- Model selection
- Regression model validation
- Supervised learning

References

1. ^ *a b c d* Geman, Stuart; E. Bienenstock; R. Doursat (1992). "Neural networks and the bias/variance dilemma" (http://web.mit.edu/6.435/www/Geman92.pdf). *Neural Computation* **4**: 1–58. doi:10.1162/neco.1992.4.1.1 (https://dx.doi.org/10.1162%2Fneco.1992.4.1.1).
2. **^** Bias–variance decomposition, In Encyclopedia of Machine Learning. Eds. Claude Sammut, Geoffrey I. Webb. Springer 2011. pp. 100-101
3. ^ *a b c* Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). *An Introduction to Statistical Learning* (http://www-bcf.usc.edu/~gareth/ISL/). Springer.
4. ^ *a b* Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2009). *The Elements of Statistical Learning* (http://statweb.stanford.edu/~tibs/ElemStatLearn/).
5. **^** Vijayakumar, Sethu (2007). "The Bias–Variance Tradeoff" (http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf). University Edinburgh. Retrieved 19 August 2014.
6. **^** Shakhnarovich, Greg (2011). "Notes on derivation of bias-variance decomposition in linear regression" (https://web.archive.org/web/20140821063842/http://ttic.uchicago.edu/~gregory/courses/wis-ml2012/lectures/biasVarDecom.pdf). Archived from the original (http://ttic.uchicago.edu/~gregory/courses/wis-ml2012/lectures/biasVarDecom.pdf) on 21 August 2014. Retrieved 20 August 2014.
7. **^** Domingos, Pedro (2000). *A unified bias-variance decomposition* (http://homes.cs.washington.edu/~pedrod/bvd.pdf). ICML.
8. **^** Valentini, Giorgio; Dietterich, Thomas G. (2004). "Bias–variance analysis of support vector machines for the development of SVM-based ensemble methods". *JMLR* **5**: 725–775.
9. **^** Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich (2008). *Introduction to Information Retrieval* (http://nlp.stanford.edu/IR-book/). Cambridge University Press. pp. 308–314.
10. **^** Gagliardi, F. (2011) "Instance-based classifiers applied to medical databases: diagnosis and knowledge extraction". Artificial Intelligence in Medicine. Volume 52, Issue 3 , Pages 123-139. http://dx.doi.org/10.1016/j.artmed.2011.04.002
11. **^** Jo-Anne Ting, Sethu Vijaykumar, Stefan Schaal, Locally Weighted Regression for Control. In Encyclopedia of Machine Learning. Eds. Claude Sammut, Geoffrey I. Webb. Springer 2011. p. 615
12. **^** Scott Fortmann-Roe. Understanding the Bias–Variance Tradeoff. 2012. http://scott.fortmann-roe.com/docs/BiasVariance.html
13. **^** Gigerenzer, Gerd; Brighton, Henry (2009). "Homo Heuristicus: Why Biased Minds Make Better Inferences". *Topics in Cognitive Science* **1**: 107–143. doi:10.1111/j.1756-8765.2008.01006.x (https://dx.doi.org/10.1111%2Fj.1756-8765.2008.01006.x). PMID 25164802 (https://www.ncbi.nlm.nih.gov/pubmed/25164802).

External links

- http://scott.fortmann-roe.com/docs/BiasVariance.html

Categories: Dilemmas | Model selection | Machine learning | Statistical classification

---