

# Machine Learning

Dr. Indu Joshi

Assistant Professor at  
Indian Institute of Technology Mandi

27 August 2025

# Bias-Variance Tradeoff

- We are given a dataset  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , drawn i.i.d. from some distribution  $P(X, Y)$ .
- Throughout this lecture we assume a regression setting, i.e.,  $y \in \mathbb{R}$ .
- In this lecture we will decompose the generalization error of a classifier into three rather interpretable terms.
- Before we do that, let us consider that for any given input  $x$  there might not exist a unique label  $y$ .

## Bias-Variance Tradeoff

- For example, if your vector describes features of the house (e.g. no. of bedrooms, square footage, ...) and the label  $y$  its price, you could imagine two houses with identical descriptions selling for different prices.
- So for any given feature vector  $x$ , there is a distribution over possible labels. We therefore define the following, which will come in useful later on.

# Terminology

**Expected Label** (given  $x \in \mathbb{R}$ ):

$$\bar{y}(x) = E_{y|x}[Y] = \int_y yP(y|x)dy$$

- The expected label denotes the label you would expect to obtain, given a feature vector  $x$ .
- Alright, so we draw our training set  $D$ , consisting of  $n$  inputs, i.i.d. from the distribution  $\alpha$ .
- As a second step we typically call some machine learning algorithm  $A$  on this data set to learn a hypothesis
- Formally, we denote this process as  $h_D = A(D)$ .

# Terminology

For a given  $h_D$ , learned on dataset  $D$  with algorithm  $A$ , we can compute the generalization error (as measured in squared loss) as follows:

**Expected Test Error** (given  $h_D$ ):

$$E_{(x,y) \sim \alpha}[(h_D(x) - y)^2] = \int_x \int_y (h_D(x) - y)^2 P(x, y) dy dx$$

- Note that one can use other loss functions.
- We use squared loss because it has nice mathematical properties, and it is also the most common loss function.
- The previous statement is true for a given training set  $D$ .
- However, remember that  $D$  itself is drawn from  $\alpha^n$ , and is therefore a random variable.

# Terminology

$h_D$ , is a function of  $D$ , and is therefore also a random variable.  
And we can of course compute its expectation:

**Expected Classifier** (given  $A$ ):

$$\bar{h} = E_{D \in \alpha^n}[h_D] = \int_D h_D P(D) dD$$

- where  $P(D)$  is the probability of drawing  $D$  from  $\alpha^n$ .
- Here,  $\bar{h}$  is a weighted average over functions.
- We can also use the fact that  $h_D$  is a random variable to compute the expected test error only given  $A$ , taking the expectation also over  $D$ .

# Terminology

**Expected Test Error** (given  $A$ ):

$$E_{(x,y) \sim \alpha, D \sim \alpha^n} [(h_D(x) - y)^2] =$$

$$\int_D \int_x \int_y (h_D(x) - y)^2 \alpha(x, y) \alpha(D) dy dx dD$$

- To be clear,  $D$  is our training points and the  $(x, y)$  pairs are the test points.
- We are interested in exactly this expression, because it evaluates the quality of a machine learning  $A$  algorithm with respect to a data distribution  $\alpha(X, Y)$ .

## Decomposition of Expected Test Error

$$\begin{aligned} E_{x,y,D}[(h_D(x) - y)^2] &= E_{x,y,D}[((h_D(x) - \bar{h}(x)) + (\bar{h}(x) - y))^2] \\ &= E_{x,D}[(h_D(x) - \bar{h}(x))^2] + 2E_{x,y,D}[(h_D(x) - \bar{h}(x))(\bar{h}(x) - y)] \\ &\quad + E_{x,y}[(\bar{h}(x) - y)^2] \end{aligned}$$

The middle term of the above equation is 0 as we show below

$$E_{x,y,D}[(h_D(x) - \bar{h}(x))(\bar{h}(x) - y)] = E_{x,y}[E_D[h_D(x) - \bar{h}(x)](\bar{h}(x) - y)]$$

## Decomposition of Expected Test Error (contd.)

$$\begin{aligned} &= E_{x,y}[(E_D[h_D(x)] - \bar{h}(x))(\bar{h}(x) - y)] \\ &= E_{x,y}[(\bar{h}(x) - \bar{h}(x))(\bar{h}(x) - y)] = E_{x,y}[0] = 0 \end{aligned}$$

Returning to the earlier expression, we're left with the variance and another term:

$$E_{x,y,D}[(h_D(x) - y)^2] = \underbrace{E_{x,D}[(h_D(x) - \bar{h}(x))^2]}_{\text{Variance}} + E_{x,y}[(\bar{h}(x) - y)^2]$$

We can break down the second term in the above equation as follows:

$$E_{x,y}[(\bar{h}(x) - y)^2] = E_{x,y}[(\bar{h}(x) - \bar{y}(x)) + (\bar{y}(x) - y)^2]$$

## Decomposition of Expected Test Error (contd.)

$$\underbrace{E_{x,y}[(\bar{y}(x) - y)^2]}_{\text{Noise}} + \underbrace{E_x[(\bar{h}(x) - \bar{y}(x))^2]}_{\text{Bias}^2} + 2E_{x,y}[(\bar{h}(x) - \bar{y}(x))(\bar{y}(x) - y)]$$

The third term in the equation above is 0, as we show below:

$$\begin{aligned} E_{x,y}[(\bar{h}(x) - \bar{y}(x))(\bar{y}(x) - y)] &= E_x[E_{y|x}[\bar{y}(x) - y](\bar{h}(x) - \bar{y}(x))] \\ &= E_x[E_{y|x}[\bar{y}(x) - y](\bar{h}(x) - \bar{y}(x))] \\ &= E_x[(\bar{y}(x) - E_{y|x}[y])(\bar{h}(x) - \bar{y}(x))] \\ &= E_x[(\bar{y}(x) - \bar{y}(x))(\bar{h}(x) - \bar{y}(x))] \\ &= E_x[0] = 0 \end{aligned}$$

## Decomposition of Expected Test Error (contd.)

$$\underbrace{E_{x,y,D}[(h_D(x) - y)^2]}_{\text{Expected Test Error}} = \underbrace{E_{x,D}[(h_D(x) - \bar{h}(x))^2]}_{\text{Variance}} + \underbrace{E_{x,y}[(\bar{y}(x) - y)^2]}_{\text{Noise}}$$
$$+ \underbrace{E_{x,y}[(\bar{h}(x) - \bar{y}(x))^2]}_{\text{Bias}^2}$$

## Significance: Variance

- Captures how much your classifier changes if you train on a different training set.
- How "overspecialized" is your classifier to a particular training set (overfitting)?
- If we have the best possible model for our training data, how far off are we from the average classifier?

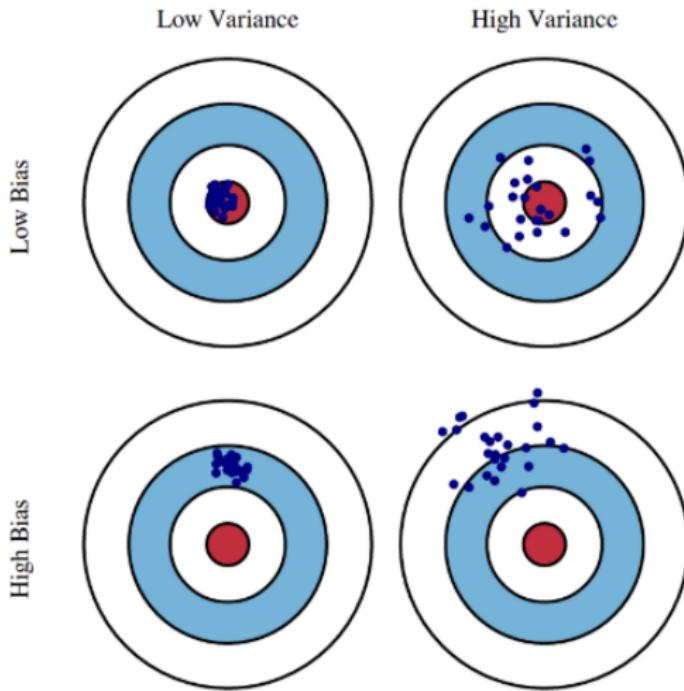
## Significance: Bias

- What is the inherent error that you obtain from your classifier even with infinite training data?
- This is due to your classifier being "biased" to a particular kind of solution (e.g. linear classifier).
- In other words, bias is inherent to your model.

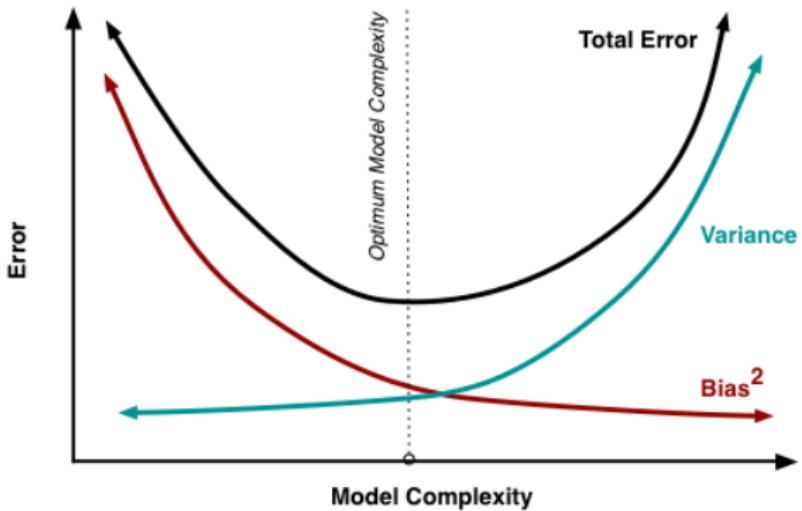
## Significance: Noise

- How big is the data-intrinsic noise?
- This error measures ambiguity due to your data distribution and feature representation.
- You can never beat this, it is an aspect of the data.

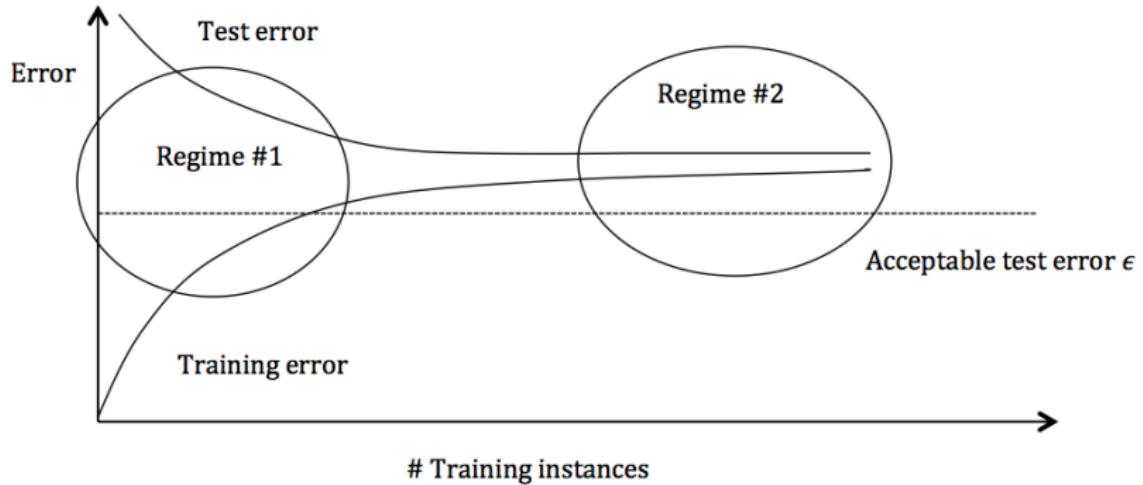
# Bias-Variance Tradeoff



# Bias-Variance Tradeoff



# Detecting High Bias and High Variance



## Detecting High Bias and High Variance

- If a classifier is under-performing (e.g. if the test or training error is too high), there are several ways to improve performance.
- The graph above plots the training error and the test error and can be divided into two overarching regimes.
- In the first regime (on the left side of the graph), training error is below the desired error threshold (denoted by ), but test error is significantly higher.
- In the second regime (on the right side of the graph), test error is remarkably close to training error, but both are above the desired tolerance of .

## Regime 1: High Variance

In the first regime, the cause of the poor performance is high variance.

### Symptoms:

- Training error is much lower than test error
- Training error is lower than  $\epsilon$
- Test error is above  $\epsilon$

### Remedies:

- Add more training data
- Reduce model complexity – complex models are prone to high variance
- Bagging

## Regime 2: High Bias

Unlike the first regime, the second regime indicates high bias: the model being used is not robust enough to produce an accurate prediction.

### Symptoms:

- Training error is higher than  $\epsilon$

### Remedies:

- Use more complex model (e.g. kernelize, use non-linear models)
- Add features
- Boosting

# Thank You

**Contact: [indujoshi@iitmandi.ac.in](mailto:indujoshi@iitmandi.ac.in)**