The bias and variance tradeoff

Understanding the error

- We want to improve our regressors and classifiers
  - To do this, we need a deeper understanding where the error is coming from!
  - We will try to decompose the error into different components, and then later develop techniques that improve on them.
- Note: throughout this class, we will assume a regression setting.

Training data

$$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$$

We will assume that the training data $\mathcal{D}$ is drawn from some distribution $P(X, Y)$

- As this is probabilistic, there is no unique correct y for a given $\boldsymbol{x}$.
- We can find, however, an **expected label**:

$$\overline{y}(\boldsymbol{x}) = \mathbb{E}_{y|\boldsymbol{x}}[Y] = \int_y y \, Pr(y|\boldsymbol{x}) dy$$

- Now, we have a **machine learning algorithm** $\mathcal{A}$ which takes $\mathcal{D}$ and creates a predictor $f_{\mathcal{D}}(\boldsymbol{x}) \rightarrow \hat{y}$.

$$\mathcal{A}(\mathcal{D}) \rightarrow f_{\mathcal{D}}$$

- One way for $\mathcal{A}$ to accomplish this is to find the parameters $\boldsymbol{\theta}$ of f, but it is not the only one.

Expected test error

- Let us say we learned the regressor function $f_{\mathcal{D}}$
- Now, we need to use it: hopefully also on data drawn from P
- What will be our expected loss (assuming squared loss):

$$\mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(f_{\mathcal{D}}(\boldsymbol{x})-y\right)^2\right]$$

- This depends on the $\mathcal{D}$ we used!
- If we would have had a different $\mathcal{D}'$, still sampled from the same P, and the same learning algorithm $\mathcal{A}$ we would have had a different expected error!

Expected regressor and expected test error

- Let us say we commited to $\mathcal{A}$ but we don't know how our $\mathcal{D}$ looks like
  - "we will use $\mathcal{A} =$ "linear regression with regularization"
  - "$\mathcal{D}$ will be taken from our agency's previous years' sales data"
- **expected regressor**

$$\overline{f} = \mathbb{E}_{\mathcal{D} \sim P^n}[\mathcal{A}(\mathcal{D})]$$

So here we did a weighted average over the possible regressor functions.

- **expected test error**

$$\mathbb{E}_{\mathcal{D} \sim P^n, (\boldsymbol{x}, y) \sim P} \left[(f_{\mathcal{D}}(\boldsymbol{x}) - y)^2 \right]$$

Why do we care about these expressions?

- The expected test error gives us the quality of the machine learning algorithm $\mathcal{A}$ for a given data distribution $P(X, Y)$.
  - It doesn't depend on our training data!
- The data distribution is something like "prices given house features" or "driving actions given car sensors"
- Obviously, some distributions are easier to learn, and some learning algorithms work better for certain distributions
- And some learning algorithms are just better, period.

Breaking down the expected text error:

$$\underbrace{\mathbb{E}_{\mathcal{D}\sim P^n,(\boldsymbol{x},y)\sim P}\left[\left(f_{\mathcal{D}}(\boldsymbol{x})-y\right)^2\right]}_{\text{Expected Test Error}}=$$

$$\underbrace{\mathbb{E}_{\boldsymbol{x},\boldsymbol{\mathcal{D}}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x})-\overline{f}(\boldsymbol{x})\right)^2\right]}_{\text{Variance}}+$$

$$\underbrace{\mathbb{E}_{\boldsymbol{x},y}\left[\left(\overline{y}(\boldsymbol{x})-y\right)^2\right)}_{\text{Noise}}+$$

$$\underbrace{\mathbb{E}_{\boldsymbol{x}}\left[\left(\overline{f}(\boldsymbol{x})-\overline{y}(\boldsymbol{x})\right)^2\right]}_{\text{Bias}}$$

Improving $\mathcal{A}$ requires us to improve these components!

- These are all positive values, the larger they are, the worse our learning algorithm $\mathcal{A}$

Noise

- Measures the ambiguity inherent in the data distribution.
  - For instance, two identical houses were sold for different prices
  - There is nothing we can do about it at the level of the algorithm
  - Possibly: there might be some hidden features (e.g. one of the houses is haunted), which might reduce the noise.

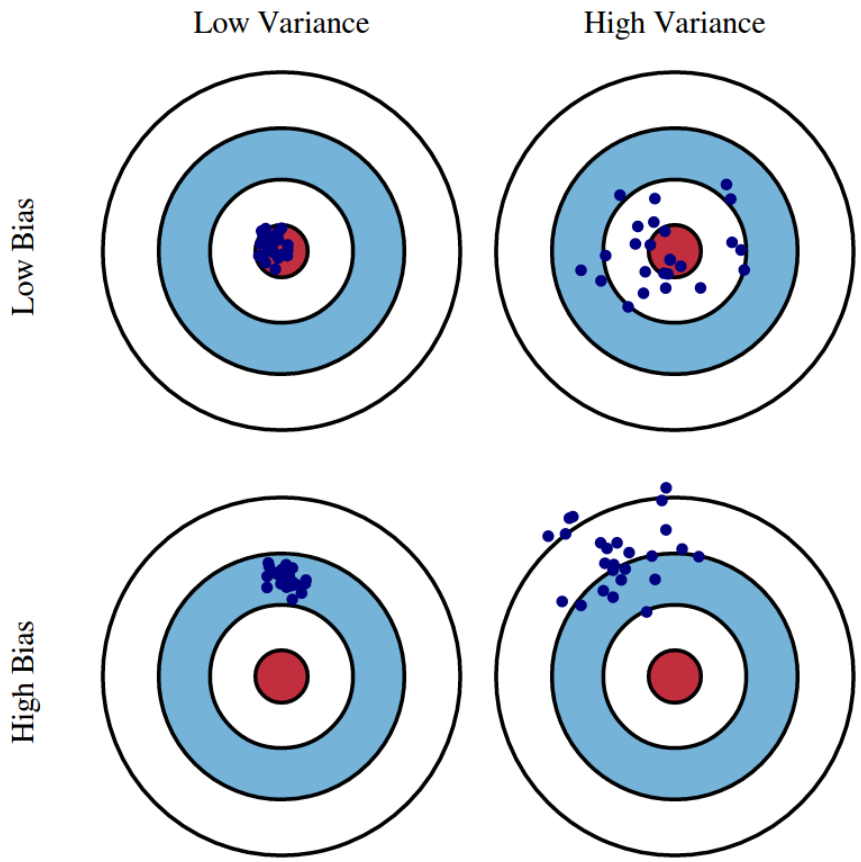
Variance

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

Bias

- How much data you obtain from your classifier even with infinite training data?
- "The classifier is biased toward a certain kind of solution"
  - For instance, linear regression will only find lines
  - It is related to the expressivity of the model
- Bias is a feature of the model

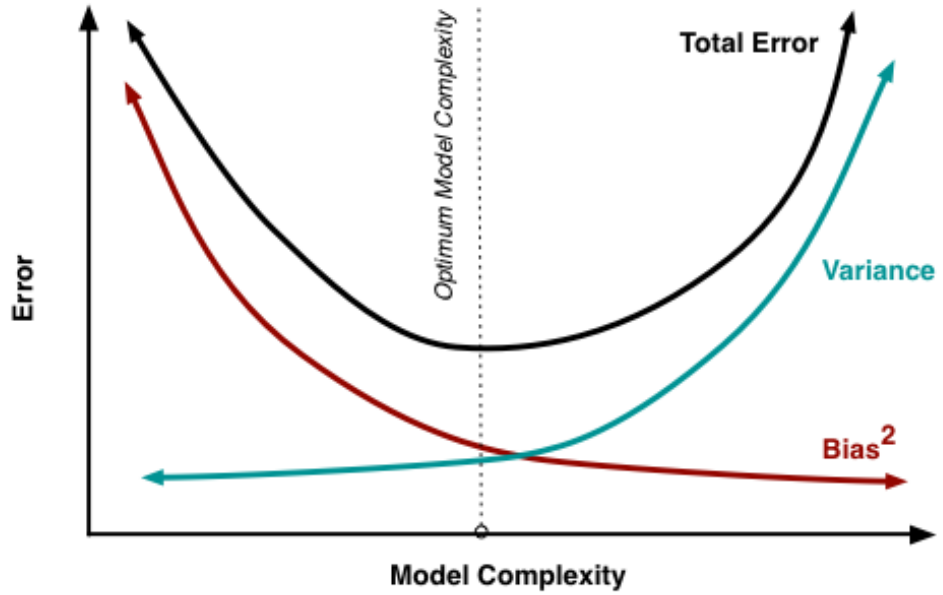
Bias vs variance



Graphical illustration of bias and variance.

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

Bias vs variance



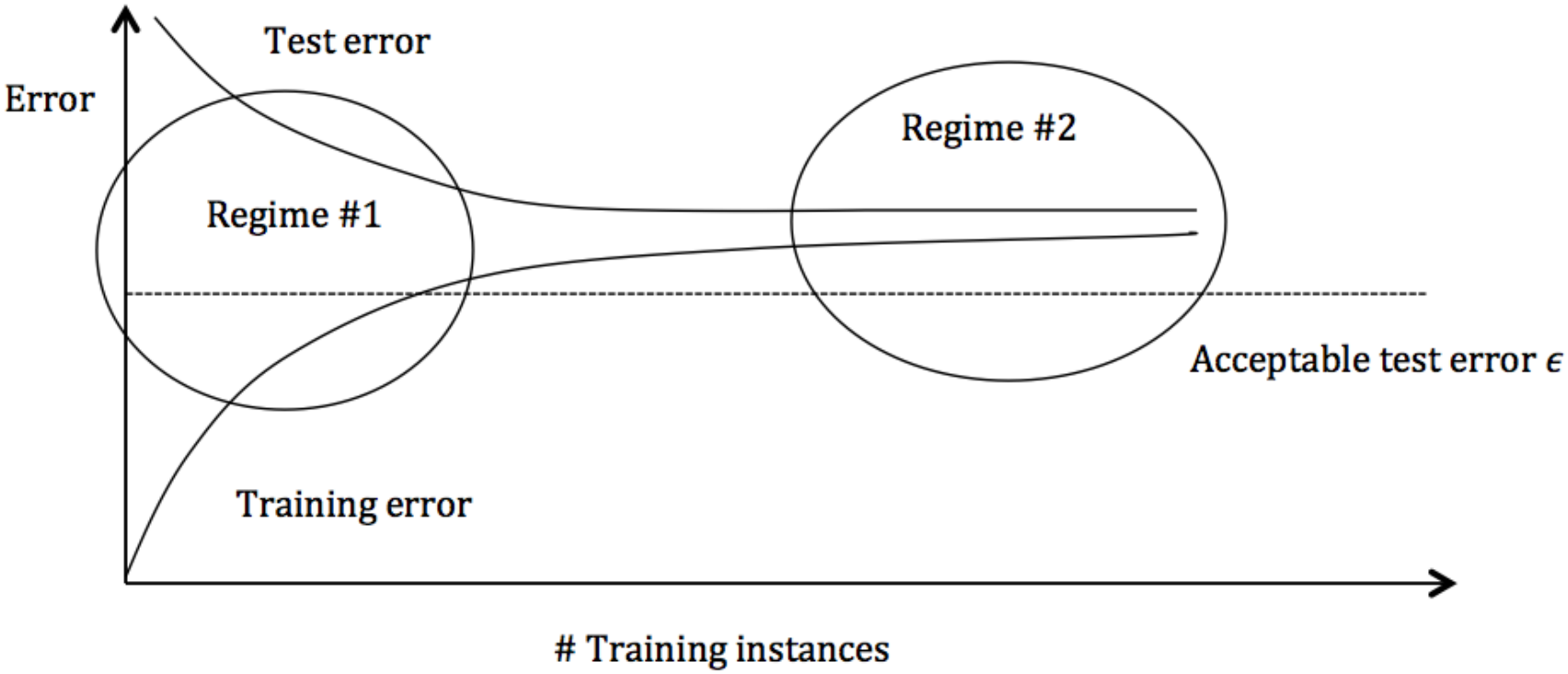
The variation of Bias and Variance with the model complexity. This is similar to the concept of overfitting and underfitting. More complex models overfit while the simplest models underfit.

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

So, how do we use this to improve our regression model (or classifier)?

- Let us say that our regression model has:
  - The training error is too high (and obviously the test error will be not good, either)
  - The training error is ok, but the test error is too high.

Training and testing regimes



Test and training error as the number of training instances increases. The learning algorithm is kept the same, and the training is always assumed to start from scratch.

High variance regime (1)

- **Symptoms**
  - Training error is much lower than test error
  - Training error is lower than acceptable value ϵ
  - Test error is above ϵ
- **Diagnosis**: the cause of the poor performance is high variance!
- **Remedies**:
  - Add more training data
  - Reduce model complexity
  - Bagging

High bias regime (2)

- **Symptoms**
  - Training error is higher than ϵ
- **Diagnosis**: high bias, the model is not expressive enough to produce an accurate prediction
- **Remedies**:
  - Increase the complexity of the model
  - Add features
  - Boosting