Combining Models

Part 3: Bias and Variance

Training versus Testing Regularization

- Modern datasets are usually high-dimensional:
  - Documents in unigram, bigram, trigram, or even higher order model
  - High resolution images stored pixel-by-pixel
- If the dimensionality of the data (denoted as *p*) is higher than the number of observations (denoted as *n*), the model is under-identified.
  - That is, we cannot find a unique combination of *p* coefficients, such that the model is optimal.
  - Consequently, the prediction will not be accurate.
- Regularization concerns building a model by reducing the dimensionality of the data (i.e. Using a subset of «predictors»)

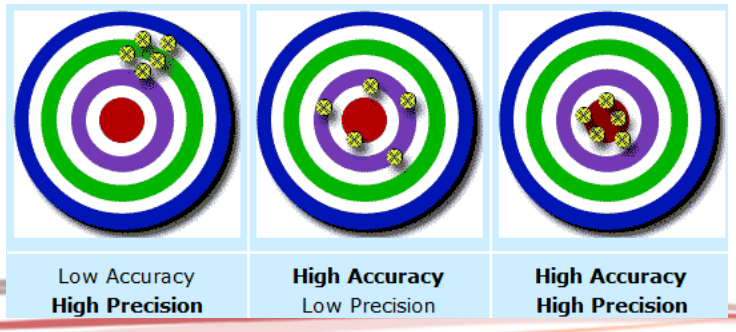
Training versus Testing
Regularization

- Regularization allows complex models to be trained on datasets of limited size without severe over-fitting, essentially by limiting the effective model complexity.

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \underbrace{\frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q}_{\text{Regularization term}}$$

Terminology

- Precision (reproducibility, repeatability): The precision of a measurement system, is the degree to which repeated measurements under unchanged conditions show the same results.
  - Example: new invented weighing machine: does it show the same value in all the experiments?
  - A measurement system can be accurate but not precise, precise but not accurate, neither, or both.
  - For example, if an experiment contains a systematic error, then increasing the sample size generally increases precision but does not improve accuracy. Eliminating the systematic error improves accuracy but does not change precision.



Model Complexity
Bias-Variance Tradeoff

- A popular choice of loss function is the squared loss function, for which the optimal prediction is given by the conditional expectation (conditional average of the target data), which we denote by *h(x)* and which is given by

$$h(x) = E[t|x] = \int tp(t|x)dt$$

p(t|x) is the conditional density of the target variable *t* conditioned on the input vector *x*.

In other words, we have a function *h(x)* which generated *x* and *t* variables. But we do not know this function. We want to approximate to this function.

Model Complexity
Bias-Variance Tradeoff

- Suppose we had a large number of datasets (***k*** number of datasets) each of size *N* and each drawn independently from the distribution $p(t, \mathbf{x})$.
- For any given data set *D*, we can run our learning algorithm and obtain a prediction function $y(\mathbf{x};D)$.
- Different data sets from the ensemble will give different functions and consequently different values of the squared loss. The performance of a particular learning algorithm is then assessed by taking the average over this ensemble of datasets.

$$\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2\right]$$
$$= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}$$

You have *k* number of models built on *k* number of datasets. You obtain the average prediction values. You subtract the target values from these predicted values to obtain the overall squared error.
In other words, you built an ensemble of *k* models to produce an average estimate which is closest to the h(x).

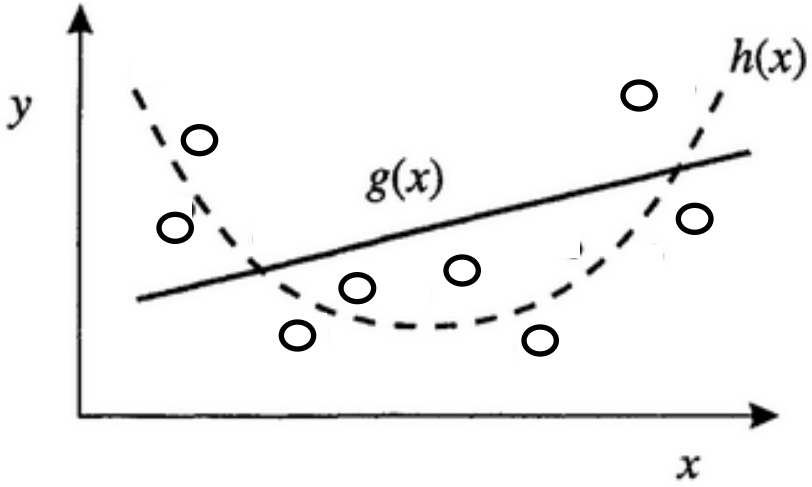
You compute the average predicted values from *k* models. You subtract each model's predicted values from the average values. Then find the square of the result.
In other words, you measure the variance of the estimates.

Model Complexity
Bias-Variance Tradeoff

- The first term, called the squared *bias*, represents the extent to which the average prediction over all data sets differs from the desired regression function.
- The second term, called the *variance*, measures the extent to which the solutions for individual data sets vary around their average, and hence this measures the extent to which the function $y(\mathbf{x};D)$ is sensitive to the particular choice of data set.

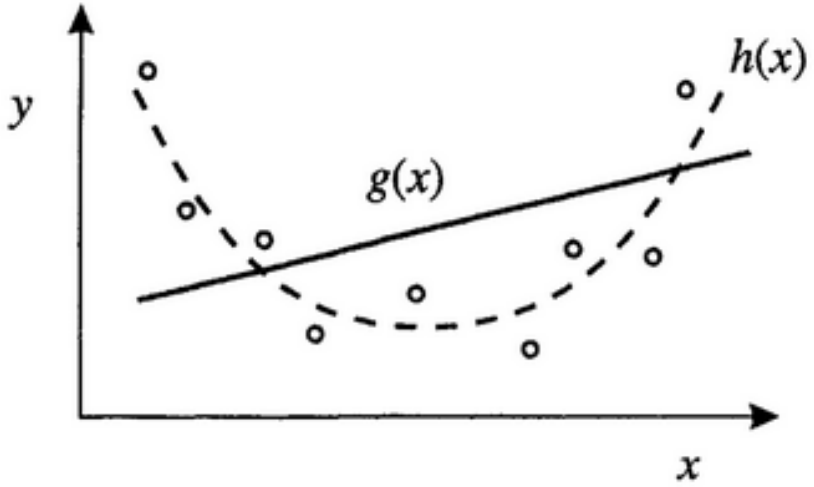
$$\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2\right]$$
$$= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}$$

Model Complexity
Bias-Variance Tradeoff



- Assume that the target data for training is generated from a smooth function *h(x)* to which zero mean random noise epsilon is added.
- A schematic illustration of the meaning of bias and variance. Circles denote a set of data points which have been generated from an underlying function *h(x)* with the addition of noise. The goal is to try to approximate *h(x)* as closely as possible. If we try to model the data by a fixed function *g(x)* (*y(x;D)* in the formula), then the bias will be high while the variance will be zero.

Model Complexity
Bias-Variance Tradeoff



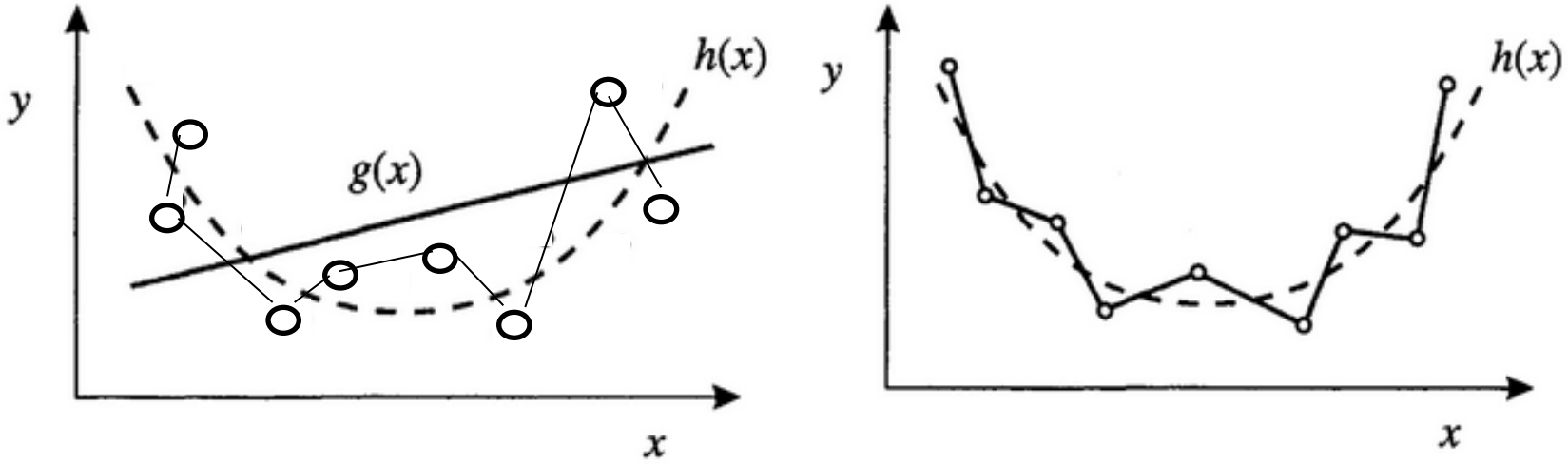
As the model is linear, it will generate the same *y* for all *x*. So this difference will be zero.

$$\underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}$$

Variance term will be vanished since $E_D[y(x)] = g(x)=y(x)$

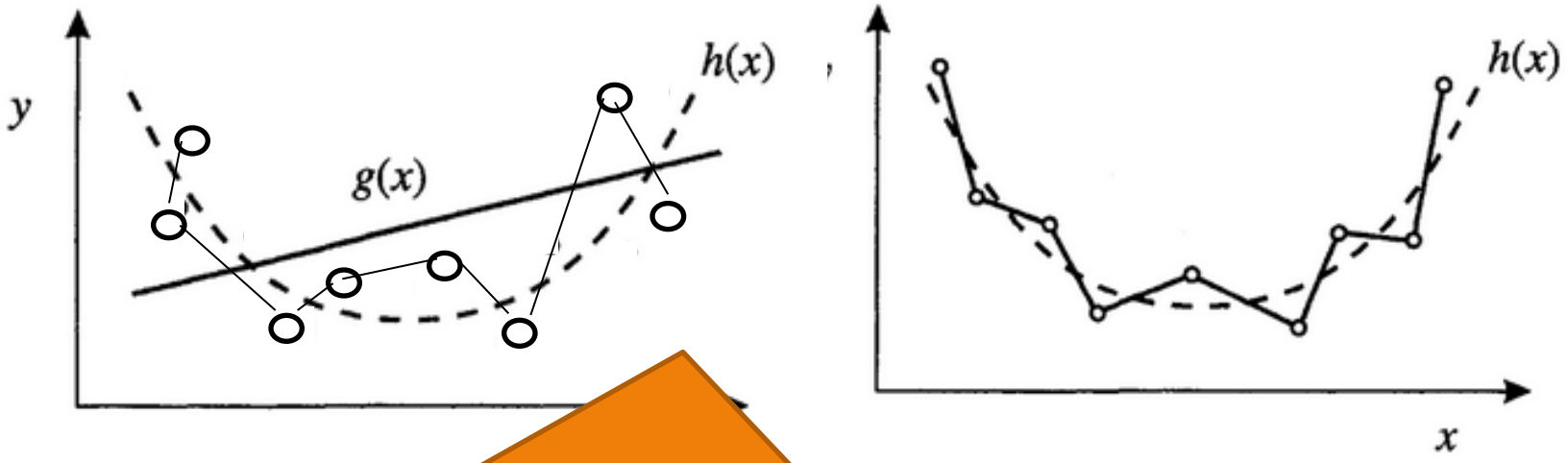
- Assume that the target data for training is generated from a smooth function *h(x)* to which zero mean random noise epsilon is added.
- A schematic illustration of the meaning of bias and variance. Circles denote a set of data points which have been generated from an underlying function h(x) with the addition of noise. The goal is to try to approximate *h(x)* as closely as possible. If we try to model the data by a fixed function *g(x)* (*y(x;D)* in the formula), then the bias will be high while the variance will be zero.

Model Complexity
Bias-Variance Tradeoff



- The opposite extreme is to take a function which fits the training data perfectly, such as the simple exact interpolant like in this figure.
- Here bias is low but the variance is high.

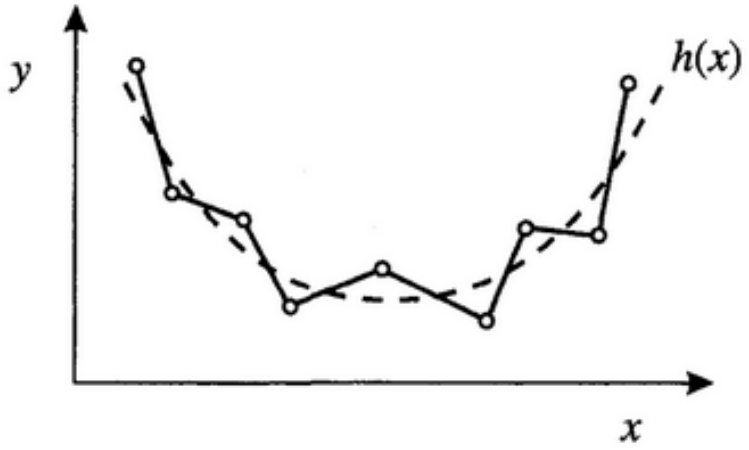
Model Complexity
Bias-Variance Tradeoff



- The o
- traini
- like in
- Here

Variance will be high since in each model (built on the resampled data set) will generate different y for the same x (due to the noise we added to h(x)). But since we predict each correctly with zero error, the bias will be zero.

Model Complexity
Bias-Variance Tradeoff



$$\underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2}.$$

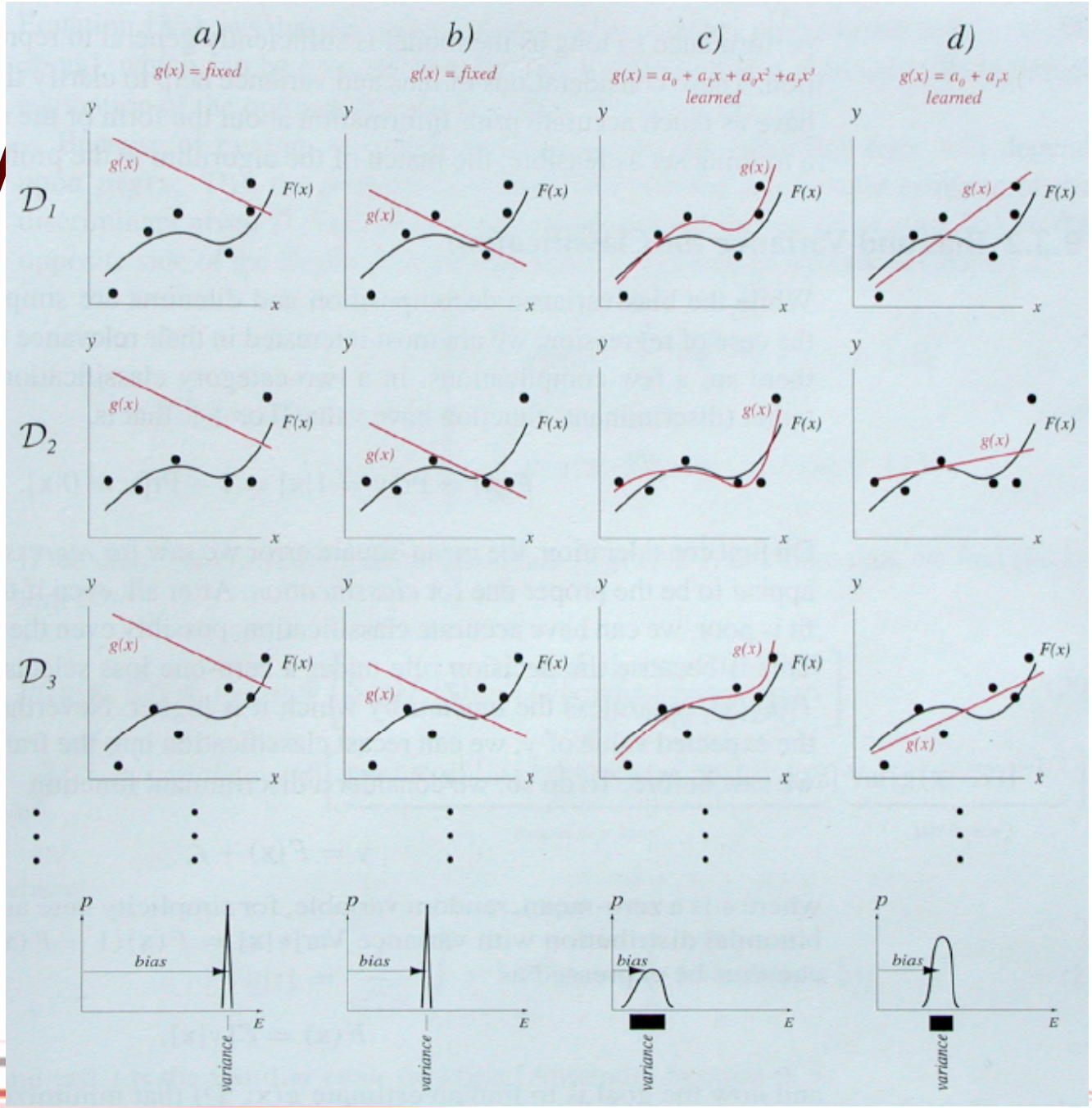
The bias term vanishes at the data points themselves since $E_D[y(x)]=E_D[h(x)+\epsilon]=h(x)=<t|x>$

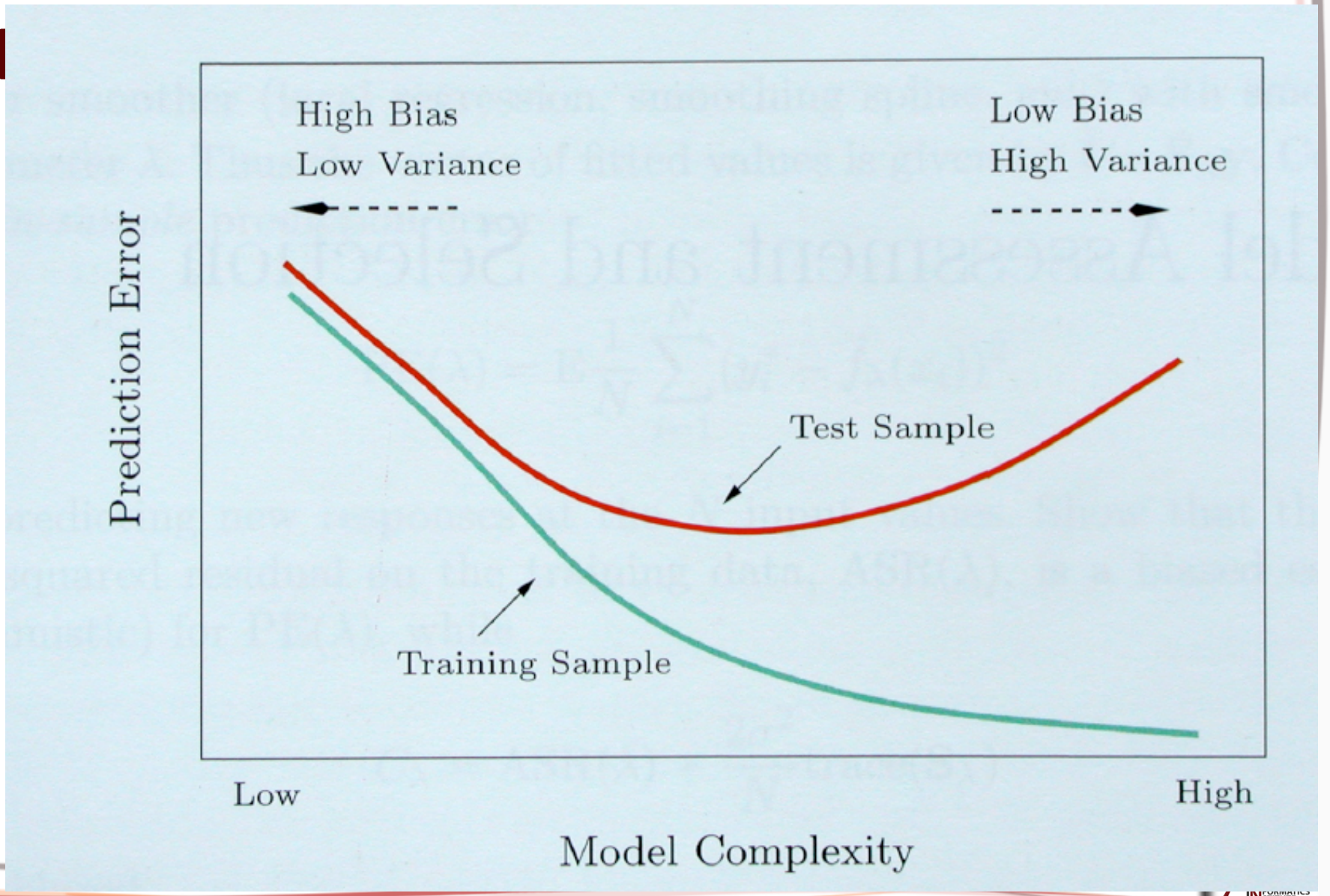
- The opposite extreme is to take a function which fits the training data perfectly, such as the simple exact interpolant like in this figure.
- Here bias is low but the variance is high.

Bias/Variance Tradeoff

- (bias+variance) is what counts for prediction
- Often:
  - low bias => high variance
  - low variance => high bias
- Tradeoff:
  - Bias vs. variance

Bias/


a)
g(x) = fixed
b)
g(x) = fixed
c)
g(x) = a0 + a1x + a2x2 + a3x3
learned
d)
g(x) = a0 + a1x
learned
D1
D2
D3
y
x
g(x)
F(x)
p
bias
variance
E


High Bias
Low Variance
Low Bias
High Variance
Prediction Error
Test Sample
Training Sample
Low
High
Model Complexity

Understanding the Error
Bias and Variance Tradeoff

- Low bias
  - linear regression applied to linear data
  - 2nd degree polynomial applied to quadratic data
  - Neural networks with many hidden units trained to completion
- High bias
  - constant function
  - linear regression applied to non-linear data
  - Neural Networks with few hidden units applied to non-linear data

Understanding the Error
Bias and Variance Tradeoff

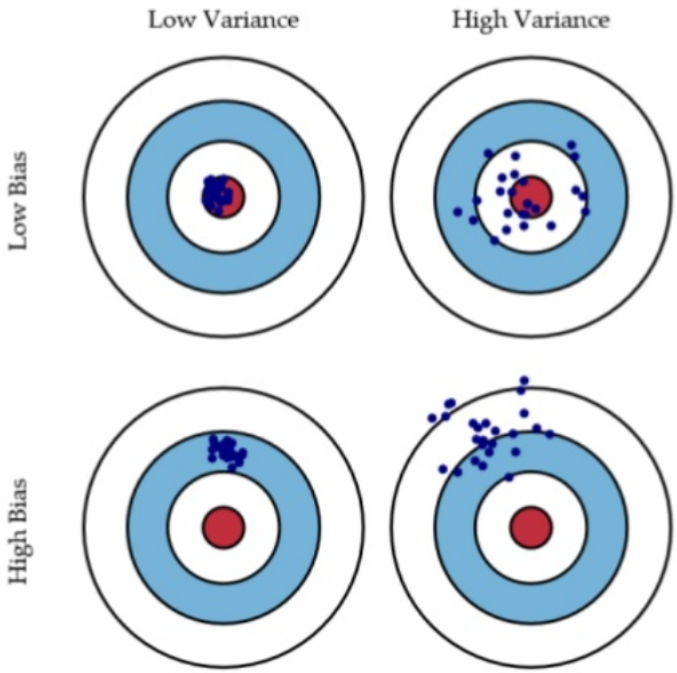
- Low variance
  - constant function
  - model independent of training data
  - model depends on stable measures of data
    - mean
    - median
- High variance
  - high degree polynomial
  - ANN with many hidden units trained to completion

Source of Variance in Supervised Learning

- noise in targets or input attributes
- bias (model mismatch)
- training sample
- randomness in learning algorithm
  - neural net weight initialization
- randomized subsetting of train set:
  - cross validation, train and early stopping set

Terminology Summary

- Error due to Bias: The error due to bias is taken as the difference between the expected (or average) prediction of our model and the correct value which we are trying to predict.
- Error due to Variance: The error due to variance is taken as the variability of a model prediction for a given data point.
- Imagine you can repeat the entire model building process multiple times. The variance is how much the predictions for a given point vary between different realizations of the model.



Reduce Variance Without Increasing Bias

- Averaging reduces variance:

$$Var(\bar{X}) = \frac{Var(X)}{N}$$

- Average models to reduce model variance
- One problem:
  - only one train set
  - where do multiple models come from?

Bagging

- Best case:

$$Var(Bagging(L(x,D))) = \frac{Variance(L(x,D))}{N}$$

- In practice:
  - models are correlated, so reduction is smaller than 1/ N
  - variance of models trained on fewer training cases usually somewhat larger
  - stable learning methods have low variance to begin with, so bagging may not help much

Reduce Bias^2 and Decrease Variance?

- Bagging reduces variance by averaging
- Bagging has little effect on bias
- Can we average *and* reduce bias?
- Yes: Boosting