Do not distribute course material

You may not and may not allow others to reproduce or distribute lecture notes and course materials publicly whether or not a fee is charged.

Topic 3 Model Selection

PROF. LINDA SELLIE

Learning objectives

- Understand how to create a more complex model using feature transformation
- Visually identity overfitting and undercutting of a model from a scatterplot
- Understand how overfitting and underfitting affect the in-sample and out of sample errors
- Understand the effect of bias/variance/noise in out of sample error
- Know how to compute generalization bound for classification
- Choose a model based on validation set
- Know how to use training, validation, and test datasets to predict the performance of a classifier on unseen data (without cheating)
- Explain the difference between (1) training error, (2) validation error, (3) cross-validation error, (4) test error, and (5) out of sample error
- Know the effect of L1 and L2 regularization and how to modify the objective function to use L1 or L2 regularization

Outline

- Motivating example: What polynomial degree should a model use?
- Polynomial transformation
- Underfitting and overfitting
- Understanding error: Bias and variance and noise
- Learning curves
- validation and model selection
- Model selection (with limited data)
- K-fold cross validation
- Regularization

How to create a more complex hypothesis

Understanding where the error comes from, and how to estimate $E_{\text{out}}[g(\mathbf{x})]$

If we have many different hypothesis classes to choose from - how can we choose wisely? And how can we estimate $E_{\text{out}}[g(\mathbf{x})]$?

Outline

- Motivating example: What polynomial degree should a …e?
- Polynomial transformation
- Underfitting and overfitting
- Understanding error: Bias and varianc…
- Learning curves
- validation and model selection
- Model selection (with limit…
- K-fold cross validation
- Regularization

Yea!

Uh oh….

How to create a more complex hypothesis

Understanding what went wrong

Understanding where the error comes from, and how to estimate $E_{\text{out}}[g(\mathbf{x})]$

Our strategy

If we have many different hypothesis classes to choose from - how can we choose wisely? And how can we estimate $E_{\text{out}}[g(\mathbf{x})]$?

How do we evaluate our model? Or choose among models (e.g. the which polynomial transformation should we choose?)

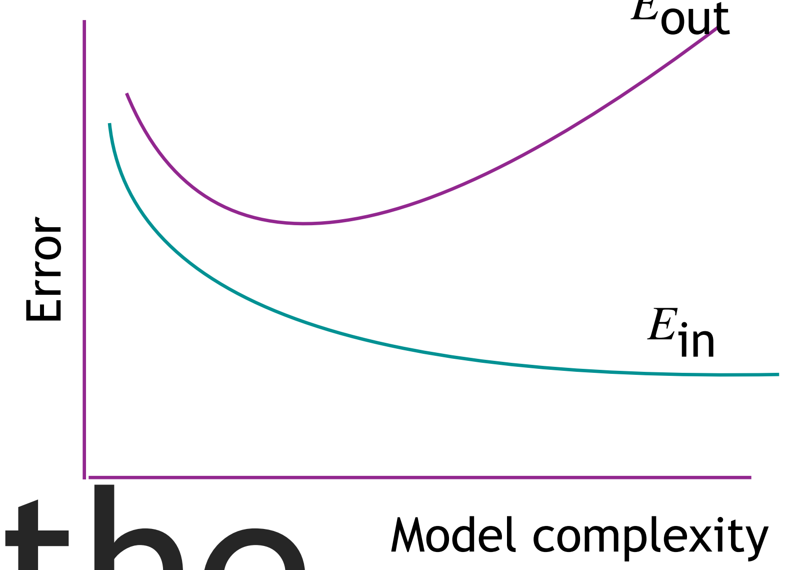
- We can evaluate how well it works by looking at its errors
- We would like the error to be zero on all future data. However
  - The unseen variables means the true model has non-zero error (i.e. the world is a messy place)
  - Our hypothesis probably doesn't contain the underlying true model
  - We don't get enough data to perfectly estimate our model. We only get a finite sample of the data. The more data we receive, the more our sample is representative of underlying data and our estimates should converge

Open discussion

Noise/irreducible error

Bias

Variance



Understanding the errors

THEORETICAL METRIC

Where did the prediction error in our hypothesis come from?

- Regression example: $y = f(\mathbf{x}) + \epsilon$

Deterministic

Noise $\sim N(0,\sigma)$
We are assuming the noise has mean 0 and variance σ^2

This means $E_{\mathbf{x},y}[f(\mathbf{x}) - y] = 0$ and $E_{\mathbf{x},y}[(f(\mathbf{x}) - y)^2] = E_{\mathbf{x}}(\epsilon^2) = \sigma^2$

Best estimate for y given $\mathbf{x}$ is $f(\mathbf{x})$

- Goal is to understand why our *expected* hypothesis (model) does not have zero error

$$E_D\left[E_{\text{out}}(g^{(D)})\right] = E_D\left[E_{\mathbf{x},y}[(g^{(D)}(\mathbf{x}) - y)^2]\right] \neq 0$$

$E_{\text{out}}(g^{(D)})$

$E_{\mathbf{x},y}[(g^{(D)}(\mathbf{x}) - y)^2]$ expected error for they hypothesis $g^{(D)}(\mathbf{x})$

The expected error of the hypothesis on any future example. The hypothesis was fit using the data set D

We focus on Algorithm Bias

Understanding Error

$$E_{\text{out}}(g^{(D)}(\mathbf{x})) = E_{\mathbf{x},y}[(y - g^{(D)}(\mathbf{x}))^2]$$

Our definitions will be for the squared loss function
You can think of how to substitute other loss functions

Bias-Variance-Noise Decompositio

$E_{\text{out}}(g)$ = bias + variance + noise

This cannot be computed in practice because we do not have access to the target function or the probability distribution

In predictions there are three sources of error.

1. noise - irreducible error
2. bias - error of average hypothesis (estimated from N examples) from the true function
3. variance - how much would the prediction for an example change if the hypothesis was fit on a different set of N points

High Bias ↔ underfitting
High Variance ↔ overfitting

Outline

- Motivating example: What polynomial degree should a ... ?
- Polynomial transformation
- Underfitting and overfitting
- Understanding error: Bias and variance and noise
  - Bias
  - Variance
  - Bias and variance and noise
- Learning curves
- validation
- Model selection
- Cross validation
- Regularization

Yea!

Uh oh....

How to create a more complex hypothesis

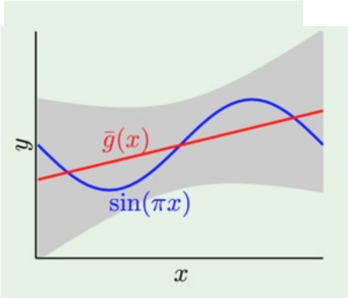
Understanding where the error comes from and how to estimate $E_{\text{out}}[g(\mathbf{x})]$

Understanding what went wrong

Our strategy

If we have many different hypothesis classes to choose from - how can we choose wisely? And how to estimate $E_{\text{out}}[g(\mathbf{x})]$?

Average Hypothesis



- Given: N training examples $D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\}$
- Learn: If I had a different set of N training examples, I would get a different hypothesis (models) $g^{(D)}(\mathbf{x})$
- Expected prediction (averaged over hypothesis): $\bar{g}(\mathbf{x}) = E_D[g^{(D)}(\mathbf{x})]$

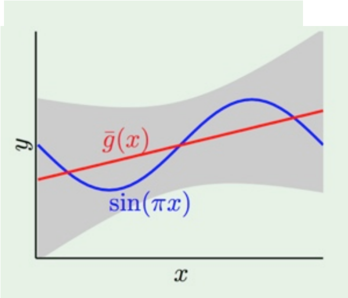
Mean prediction of the algorithm for $\mathbf{x}$

Intuitive approximation:

$$\bar{g}(\mathbf{x}) \approx \frac{1}{k}\sum_{i=1}^{k} g_i^{(D_i)}(\mathbf{x}) \quad D_1, D_2, \ldots, D_k$$

Bias

Bias of the hypothesis class (not an individual hypothesis from the class)



- bias($\mathbf{x}$) = $\left(f(\mathbf{x}) - \overline{g}(\mathbf{x})\right)^2$

Conceptually: squared difference from "average prediction" for $\mathbf{x}$, and expected label $f(\mathbf{x})$

- bias $= E_{\mathbf{x}}\left[(\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2\right]$

$$\approx \frac{1}{N}\sum_{i=1}^{N}(\bar{g}(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(i)}))^2$$

Bias of the hypothesis class

less flexible model then more bias

Occasionally this is called bias^2

When using this model class, measures how well you expect the "average prediction" to represent the true solution
We expect the bias to decreases with a more complex model

Outline

- ❑Motivating example: What polynomial degree should a ?
- ❑Polynomial transformation
- ❑Underfitting and overfitting
- ❑Understanding error: Bias and variance and noise
  - Bias
  - Variance
  - Bias and variance and noise
- ❑Learning curves
- ❑validation
- ❑Model selection
- ❑Cross validation
- ❑Regularization

Yea!

Uh oh....

How to create a more complex hypothesis

Understanding what went wrong

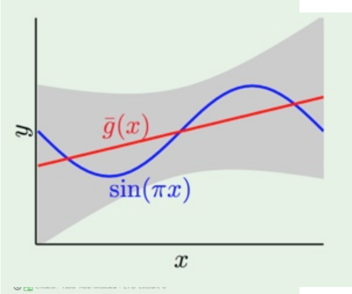
Understanding where the error comes from and how to estimate $E_{\text{out}}[g(\mathbf{x})]$

Our strategy

If we have many different hypothesis classes to choose from - how can we choose wisely? And how to estimate $E_{\text{out}}[g(\mathbf{x})]$?

Variance

Variance of a hypothesis class (model class)



- Variance: difference between the expected prediction and the prediction from a particular dataset

- $$\text{var}(\mathbf{x}) = E_D[(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2] \approx \frac{1}{L}\sum_{\ell=1}^{L}(\bar{g}(\mathbf{x}) - g_\ell^{(D_\ell)}(\mathbf{x}))^2$$

Conceptually: variance of a prediction for $\mathbf{x}$ from the mean prediction

$$\text{var} = E_{\mathbf{x}}\left[E_D\left[(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2\right]\right] \approx \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L}\sum_{\ell=1}^{L}(\bar{g}(\mathbf{x}^{(i)}) - g_\ell^{(D_\ell)}(\mathbf{x}^{(i)}))^2$$

less flexible model then less variance

Measures how sensitive a hypothesis class (model class) is to a specific dataset
Variance typically decreases with simpler models

Outline

- ❑Motivating example: What polynomial degree should a ?
- ❑Polynomial transformation
- ❑Underfitting and overfitting
- ❑Understanding error: Bias and variance and noise
  - •Bias
  - •Variance
  - •Bias and variance and noise
- ❑Learning curves
- ❑validation
- ❑Model selection
- ❑Cross validation
- ❑Regularization

Yea!

Uh oh….

How to create a more complex hypothesis

Understanding where the error comes from and how to estimate $E_{\text{out}}[g(\mathbf{x})]$

Understanding what went wrong

Our strategy

If we have many different hypothesis classes to choose from - how can we choose wisely? And how to estimate $E_{\text{out}}[g(\mathbf{x})]$?

Generalization error: bias, variance, noise decomposition

The expected error of the hypothesis $g^{(D)}(\mathbf{x})$ on any future example. The model was fit using the data set D

$$E_{\text{out}}(g^{(D)}) = E_{\mathbf{x}}[(g^{(D)}(\mathbf{x}) - y)^2]$$

The expected error of the hypothesis fit on a randomly chosen set of N training examples

$$E_D\left[E_{\text{out}}(g^{(D)})\right] = \underline{E_{\mathbf{x}}\left[E_D\left[\left(g^{(D)}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right]} + \sigma^2$$

Posted slides will show this derivation

$$E_{\mathbf{x}}\left[E_D[(g^{(D)}(\mathbf{x}) - f(\mathbf{x}))^2]\right] = \text{Bias + variance} \qquad \neq 0$$

The next slide was not presented in class

Understanding Error

Bias-Variance-Noise Decomposition

For independent a and c:
E[ac] = E[a]E[c]

The linearity of expectation:
E[a + b] = E[a] + E[b]

$$E_D\left[E_{\text{out}}(g^{(D)})\right] = E_D\left[E_{\mathbf{x}}[(g^{(D)}(\mathbf{x}) - y)^2]\right] = E_{\mathbf{x}}\left[E_D[(g^{(D)}(\mathbf{x}) - y)^2]\right]$$

$$= E_{\mathbf{x}}\left[E_D[(\underbrace{g^{(D)}(\mathbf{x}) - f(\mathbf{x})}_{A} + \underbrace{f(\mathbf{x}) - y}_{B})^2]\right]$$

(A+B)²=(A² + 2AB + B²)

$$= E_{\mathbf{x}}\left[E_D[\left(g^{(D)}(\mathbf{x}) - f(\mathbf{x})\right)^2 + 2\left(g^{(D)}(\mathbf{x}) - f(\mathbf{x})\right)\left(f(\mathbf{x}) - y\right) + \left(f(\mathbf{x}) - y\right)^2]\right]$$

$$= E_{\mathbf{x}}\left[E_D[\left(g^{(D)}(\mathbf{x}) - f(\mathbf{x})\right)^2] + 2E_D\left[\left(g^{(D)}(\mathbf{x}) - f(\mathbf{x})\right)\right]\underbrace{\left(f(\mathbf{x}) - y\right)}_{0} + E_D[\underbrace{\left(f(\mathbf{x}) - y\right)^2}_{\epsilon^2}]\right]$$

$$= E_{\mathbf{x}}\left[E_D[\left(g^{(D)}(\mathbf{x}) - f(\mathbf{x})\right)^2]\right] + \sigma^2$$

Error due to model being too simple, or there was not enough data to learn the model accurately

Proof from: https://ocw.mit.edu/courses/sloan-school-of-management/15-097-prediction-machine-learning-and-statistics-spring-2012/lecture-notes/MIT15_097S12_lec04.pdf

Understanding Error
Bias-Variance Decomposition (noise free)

$$\text{bias}(\mathbf{x}) = \left(f(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2$$

$$\text{var}(\mathbf{x}) = E_D[\,(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2]$$

$$\bar{g}(\mathbf{x}) = E_D[g^{(D)}(\mathbf{x})]$$

For any constant c,
E[ac]=cE[a]
E[a+c]=E[a]+c

The linearity of expectation:
E[a + b] = E[a] + E[b]

$$E_{\mathbf{x}}\left[E_D[\left(g^{(D)}(\mathbf{x}) - f(\mathbf{x})\right)^2]\right] = E_{\mathbf{x}}\left[E_D[(\underbrace{g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x})}_{A} + \underbrace{\bar{g}(\mathbf{x}) - f(\mathbf{x})}_{B})^2]\right]$$

$$= E_{\mathbf{x}}\left[E_D[\left(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2 + 2\left(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right) + \left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2]\right]$$

$$= E_{\mathbf{x}}\left[E_D[\left(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2] + 2E_D[\left(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)] + E_D[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2]\right]$$

$$= E_{\mathbf{x}}\left[\underbrace{E_D[\left(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2]}_{\text{variance}(\mathbf{x})} + 2E_D[\underbrace{\left(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)}_{0}]\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right) + \underbrace{E_D[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2]}_{\text{bias}(\mathbf{x})}\right]$$

$$= E_{\mathbf{x}}[\text{bias}] + E_{\mathbf{x}}[\text{variance}]$$

$$= \text{bias} + \text{variance}$$

Notice that

$$E_D[\left(g^{(D)}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)]$$

$$= E_D[g^{(D)}(\mathbf{x})] - \bar{g}(\mathbf{x})$$

Proof from:https://ocw.mit.edu/courses/sloan-school-of-management/15-097-prediction-machine-learning-and-statistics-spring-2012/lecture-notes/MIT15_097S12_lec04.pdf

Understanding Error
Bias-Variance-Noise Decomposition

The expected error of the hypothesis fit on a randomly chosen set of N training examples

$$E_{\text{out}}(g) \quad = \text{bias} + \text{variance} + \text{noise}$$

Noise in the training set contributes to variance

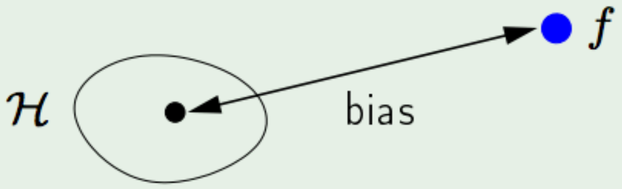
Noise in the test set contributes to irreducible error

Based on averages over what is expected for a training set D

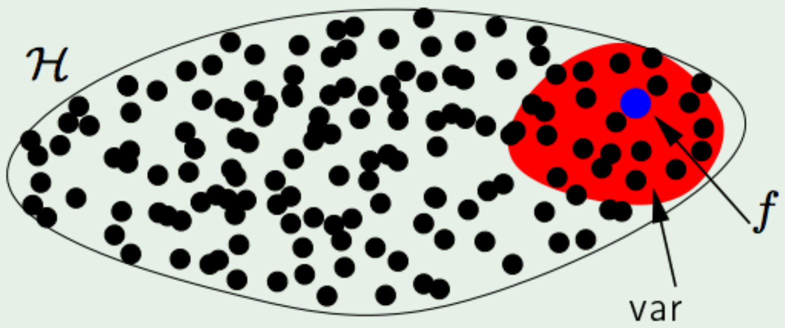
- can we lower variance without increasing too much the bias?
- can we lower bias without increasing too much the variance?

The tradeoff

$$\text{bias} = \mathbb{E}_{\mathbf{x}}\left[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \qquad \text{var} = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]\right]$$



bias = $E_{\mathbf{x}}$



$\downarrow \qquad \mathcal{H} \uparrow \qquad \uparrow$

Example: sine target

$f : [-1, 1] \to \mathbb{R} \qquad f(x) = \sin(\pi x)$

Only two training examples! $\quad N = 2$

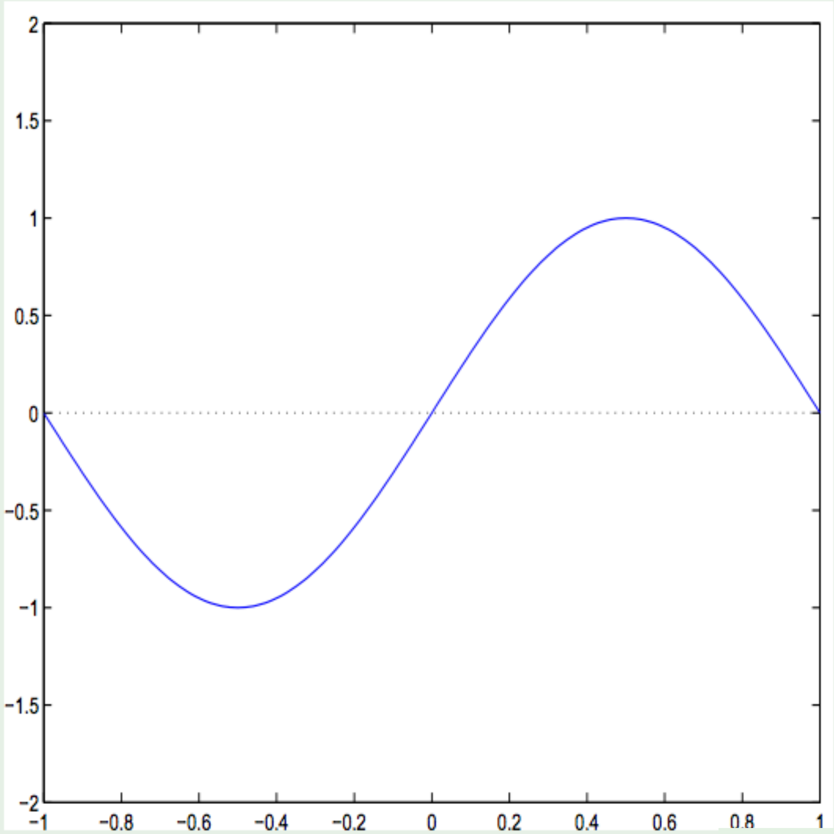
Two models used for learning:

$$\mathcal{H}_0: \quad h(x) = w_0$$

$$\mathcal{H}_1: \quad h(x) = w_0 + w_1 x$$

Which is better, $\mathcal{H}_0$ or $\mathcal{H}_1$?

f



Example: sine target

$f : [-1, 1] \rightarrow \mathbb{R}$ $\qquad f(x) = \sin(\pi x)$

Only two training examples! $\qquad N = 2$

Two models used for learning:

$$\mathcal{H}_0: \quad h(x) = w_0$$

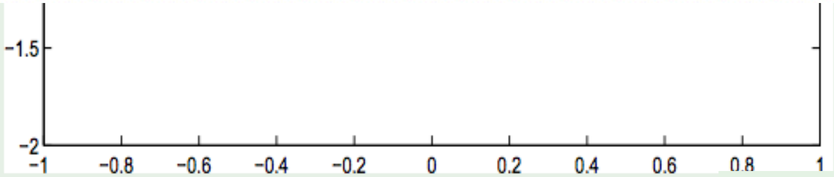
$$\mathcal{H}_1: \quad h(x) = w_0 + w_1 x$$

Which is better, $\mathcal{H}_0$ or $\mathcal{H}_1$?

f

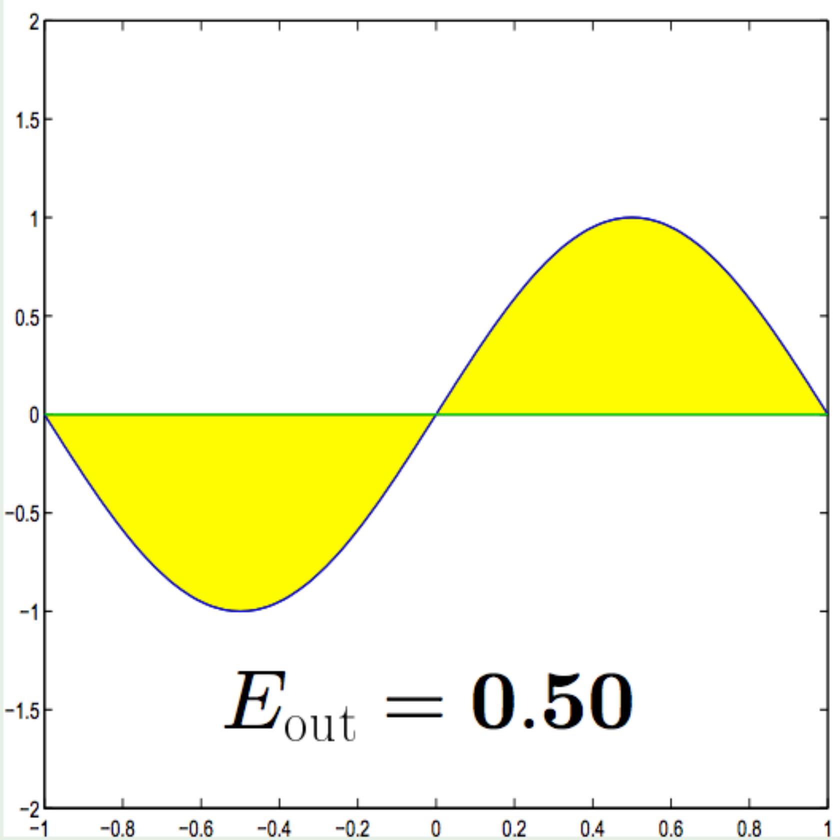
1. Which hypothesis would have a smaller generalization error?

- ○ h(x)=b
- ○ h(x)=ax+b
- ○ They would have the same generalization error

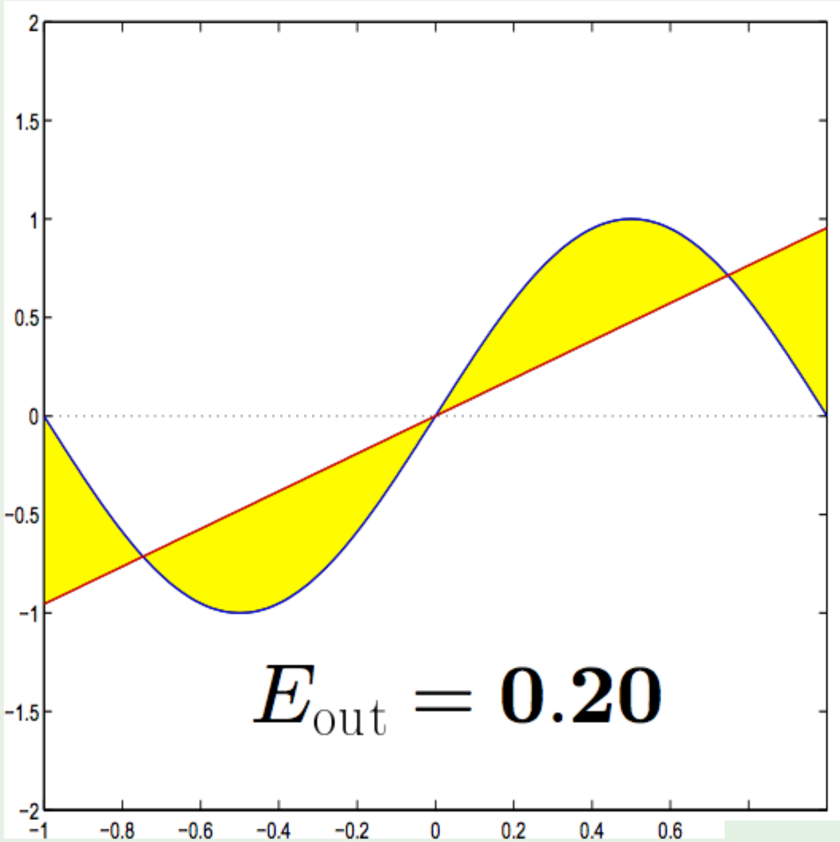


Approximation - $\mathcal{H}_0$ versus $\mathcal{H}_1$

$\mathcal{H}_0$

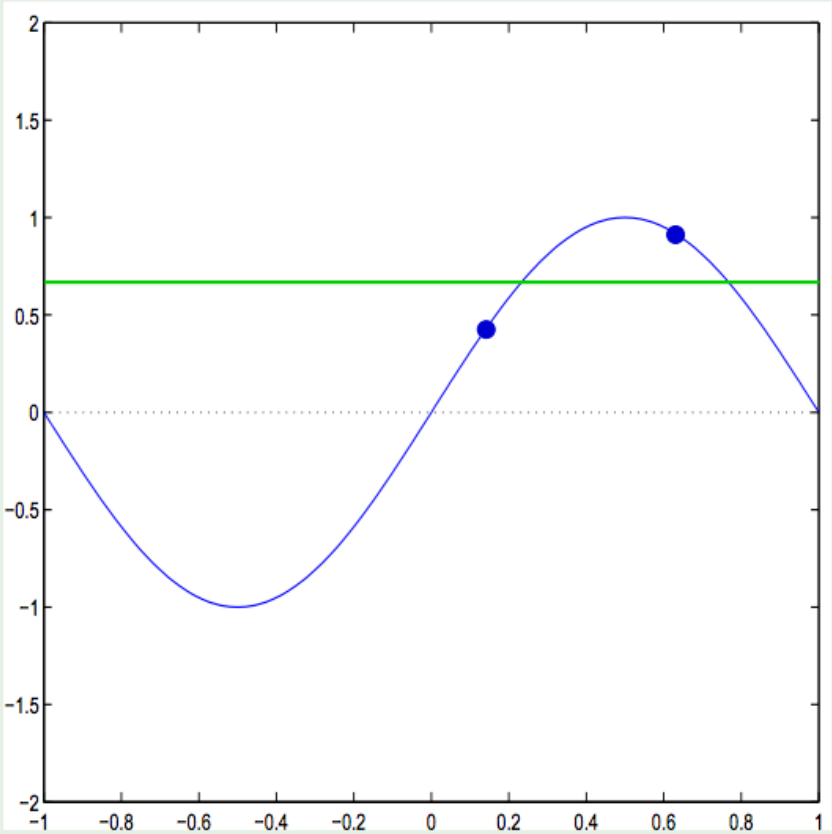


$\mathcal{H}_1$

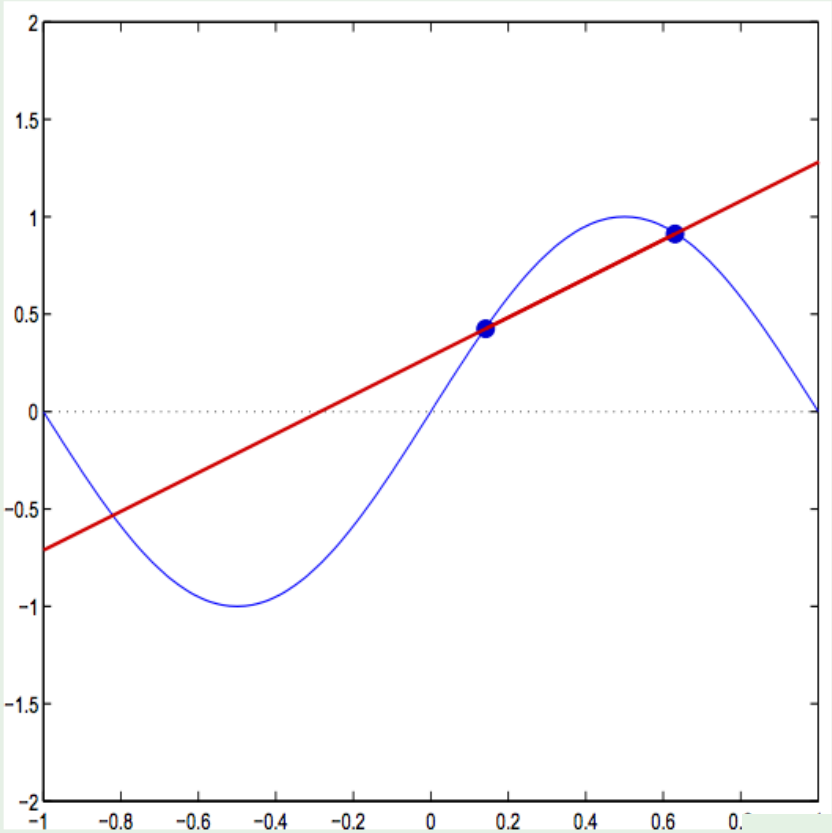


Learning - $\mathcal{H}_0$ versus $\mathcal{H}_1$

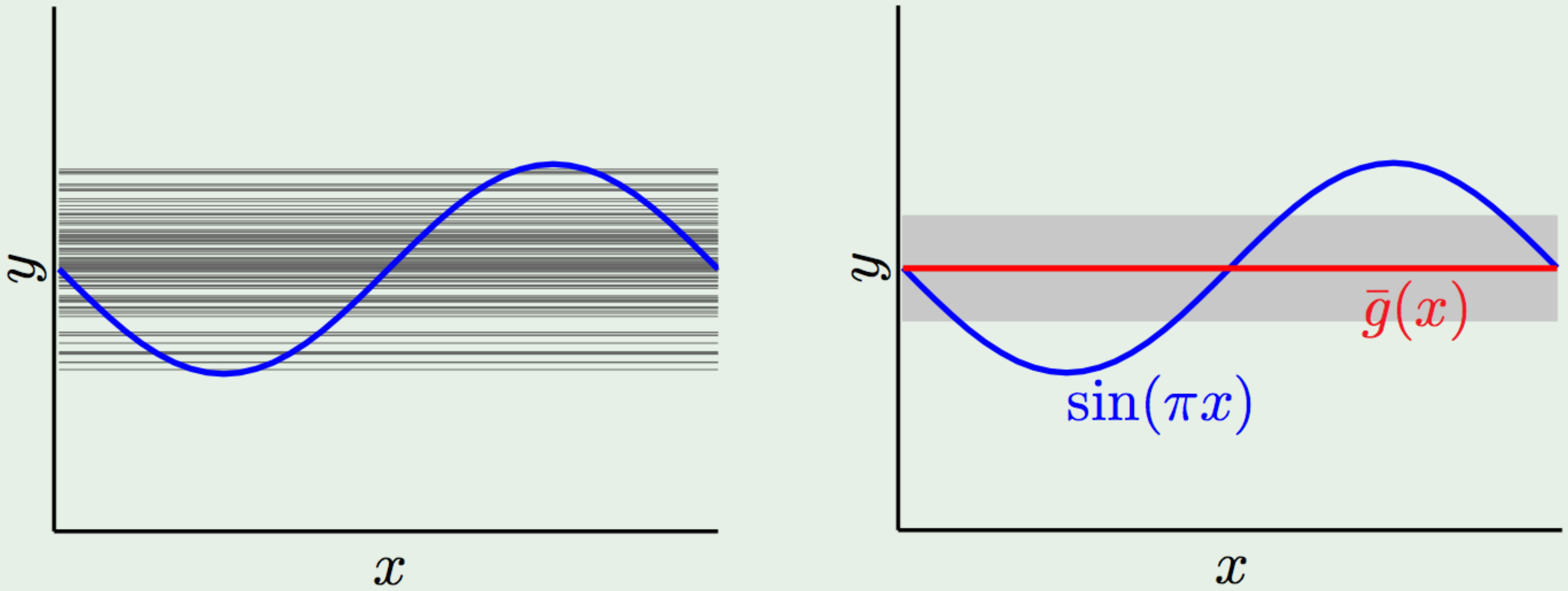
$\mathcal{H}_0$



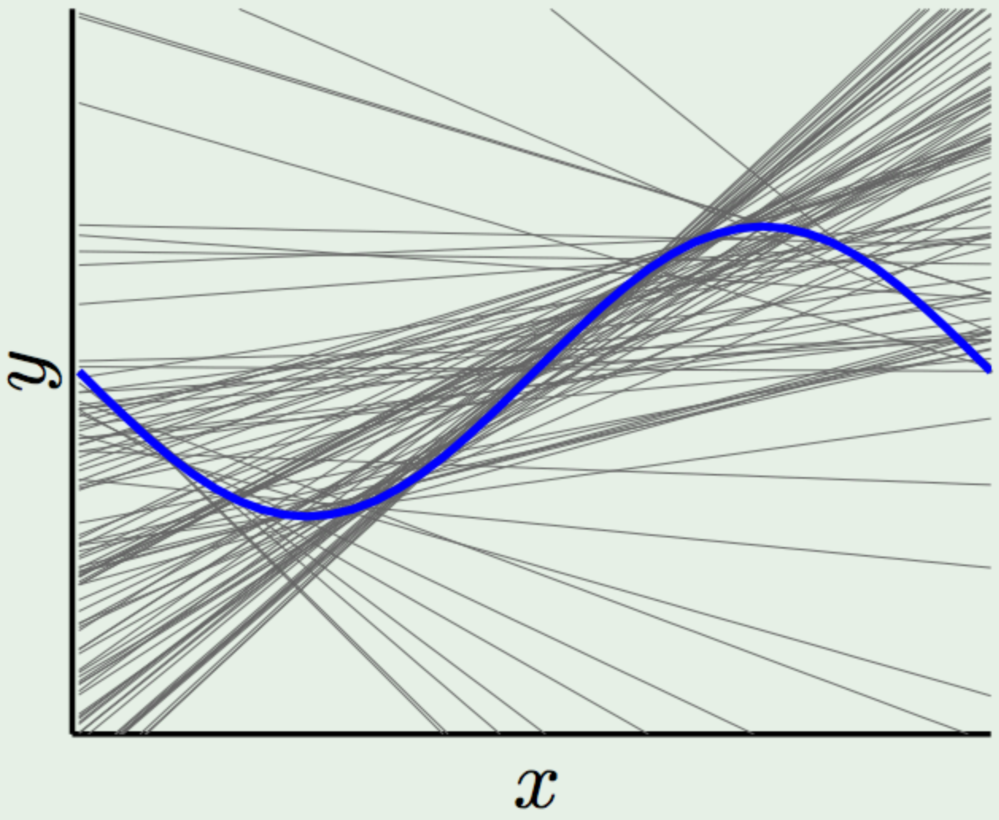
$\mathcal{H}_1$

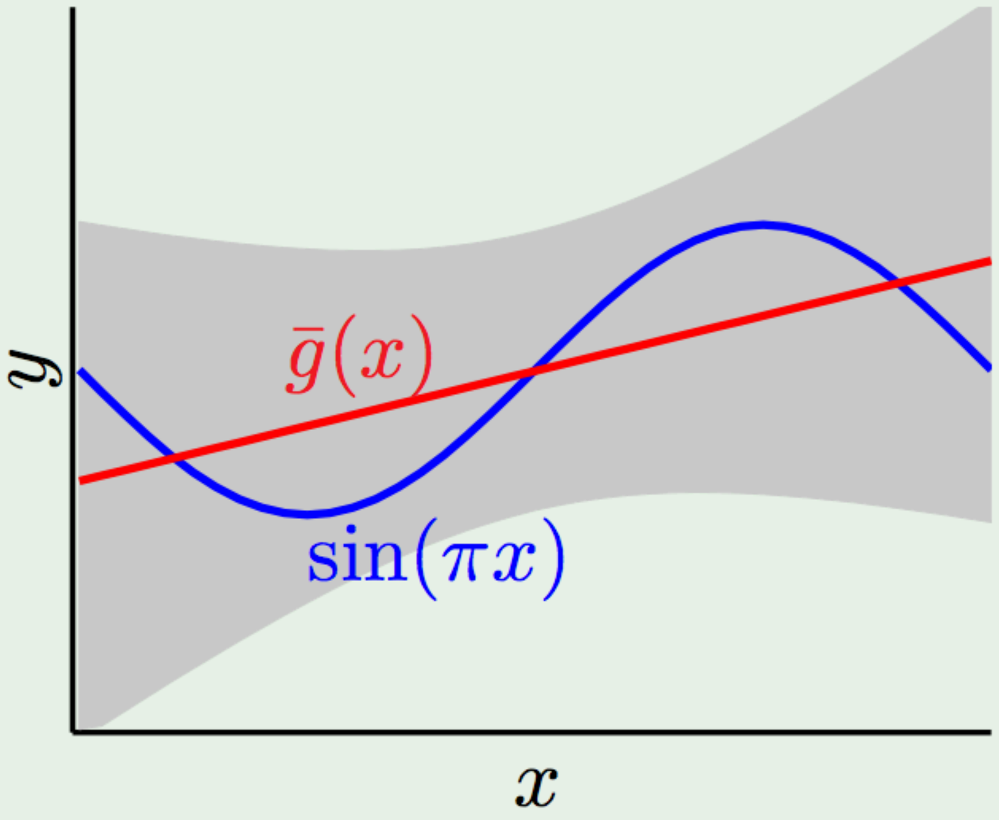


Bias and variance - $\mathcal{H}_0$



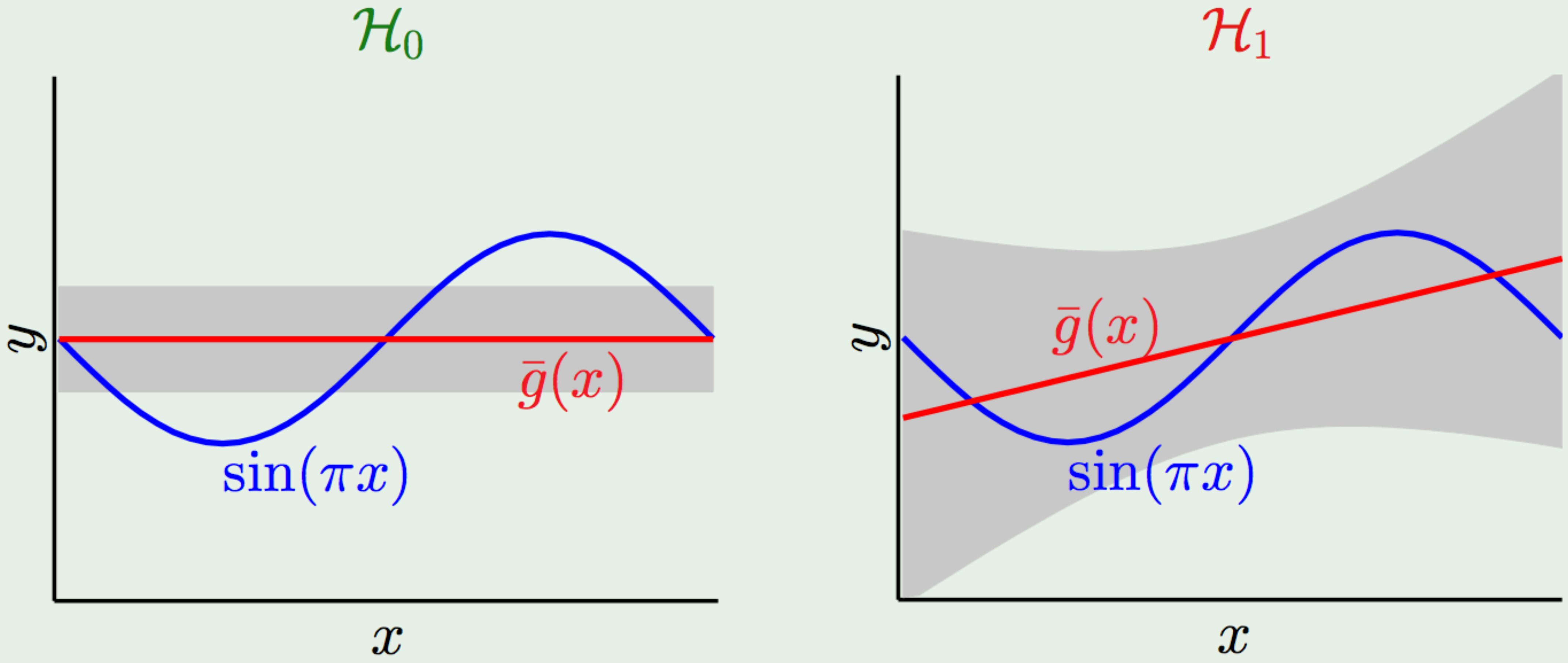
Bias and variance - $\mathcal{H}_1$





You will not be expected to compute these values

and the winner is ...



bias $=$ $\mathbf{0.50}$ var $=$ $\mathbf{0.25}$

bias $=$ $\mathbf{0.21}$ var $=$ $\mathbf{1.69}$

Lesson learned

Match the '**model complexity**'

to the data resources, not to the target complexity

Outline

- Motivating example: What polynomial degree should a [...]?
- Polynomial transformation
- Underfitting and overfitting
- Understanding error: Bias and varian[...]
- Learning curves
- validation and model selection
- Model selection (with limit[...]
- K-fold cross validation
- Regularization

Yea!

Uh oh....

How to create a more complex hypothesis

Understanding what went wrong

Understanding where the error comes from, and how to estimate $E_{\text{out}}[g(\mathbf{x})]$

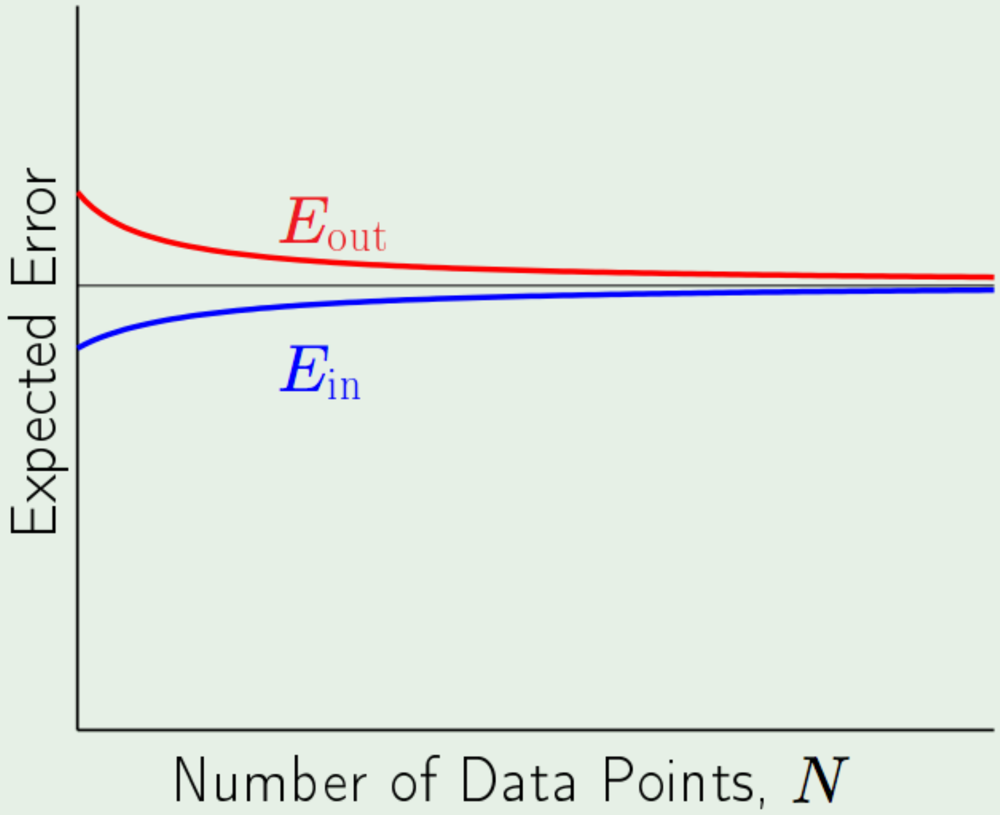
Our strategy

If we have many different hypothesis classes to choose from - how can we choose wisely? And how can we estimate $E_{\text{out}}[g(\mathbf{x})]$?

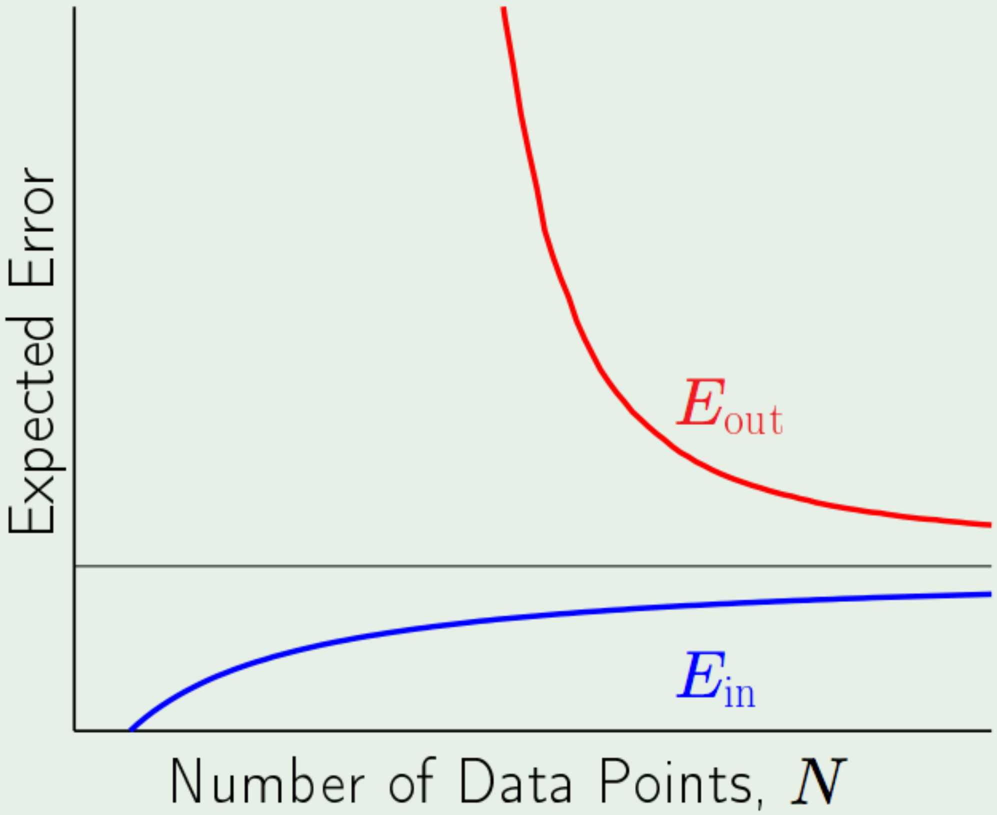
Pair Share

Do we expect the model to perform as well in the future as it performed on the training set?

The curves



Simple Model



Complex Model

Our goal is to minimize the generalization error (aka risk)
For linear regression, the goal is to minimize:

$$E_{\text{out}}(g(\mathbf{x})) = E[(y - g(\mathbf{x}))^2]$$

To do this we need to know the joint distribution of X and Y

How can we approximate this value?

Use our sample data!

…we could use our training examples to calculate our in-sample loss

$$E_{\text{in}}(g(\mathbf{x})) = \sum_{i=1}^{N} (y^{(i)} - g(\mathbf{x}^{(i)}))^2]$$

Empirical risk minimization by choosing the parameters with the highest likelihood

This is a very optimistic estimate!

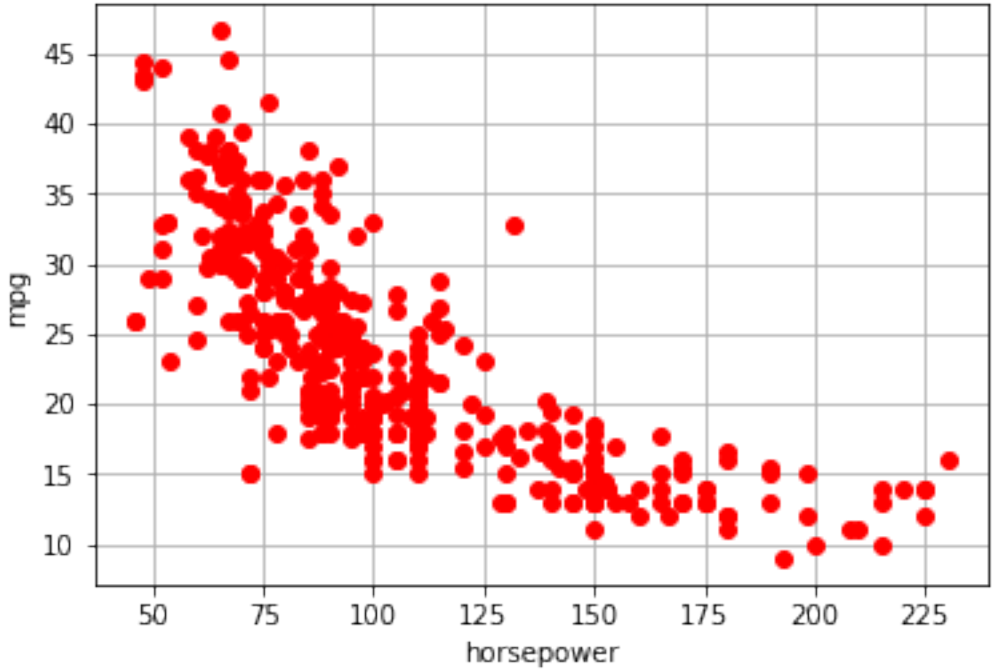
Pair share

The training error (cost) doesn't give the real world cost

$$E_{\text{out}}(g(\mathbf{x})) = E[(y - g(\mathbf{x}))^2]$$

$$E_{\text{in}}(g(\mathbf{x})) < < E_{\text{out}}(g(\mathbf{x}))$$

Data

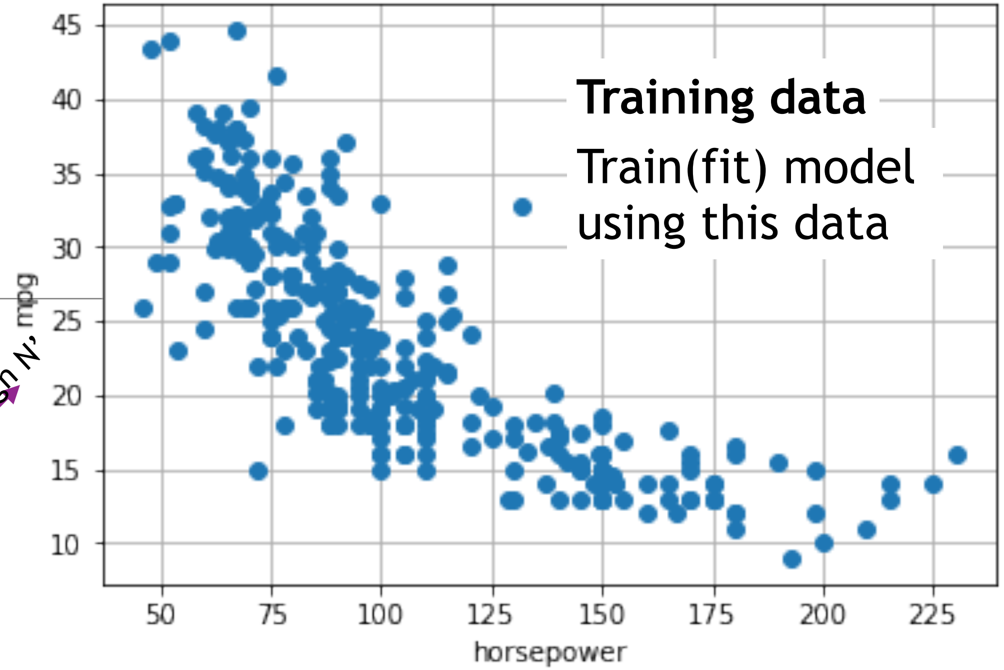


Always shuffle data before train test split

Randomly split the data

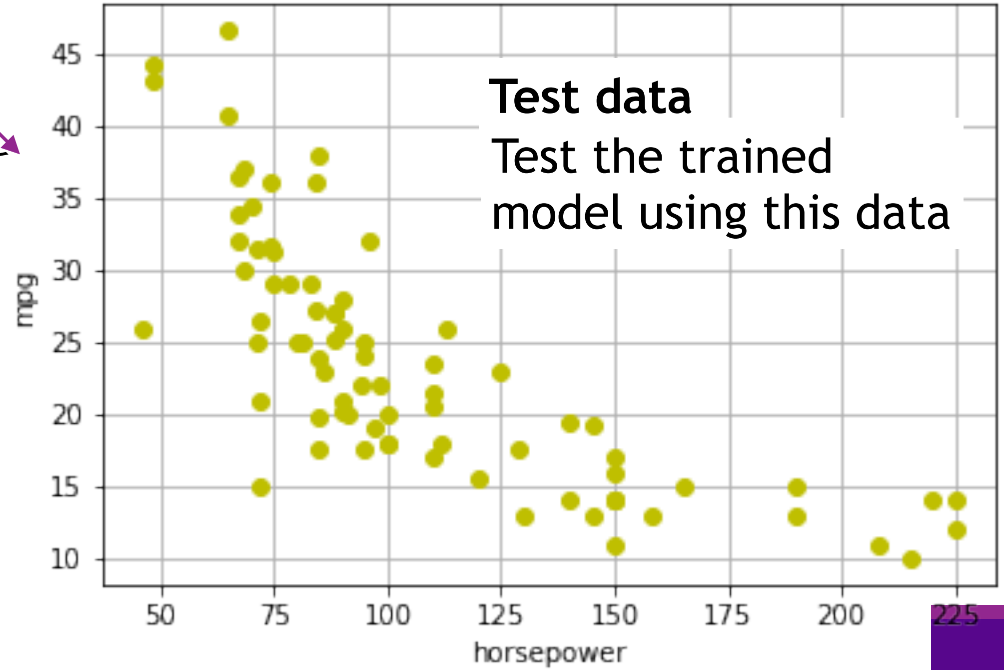
Examples K+1 through N'

Examples 1 through K



Training data

Train(fit) model using this data

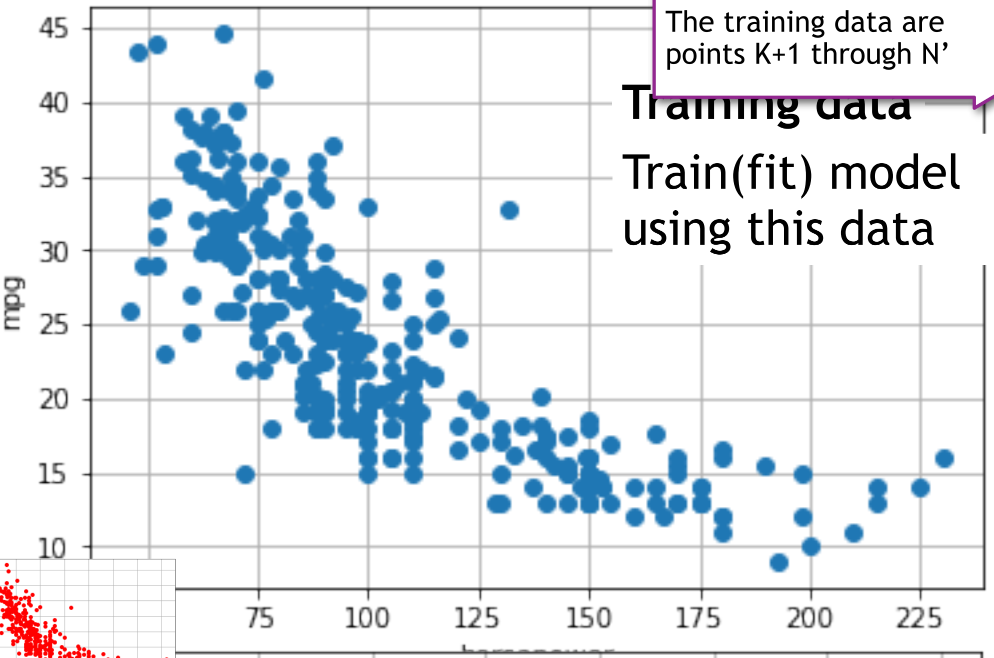


Test data

Test the trained model using this data

Fit model using the training data

The training data are points K+1 through N'



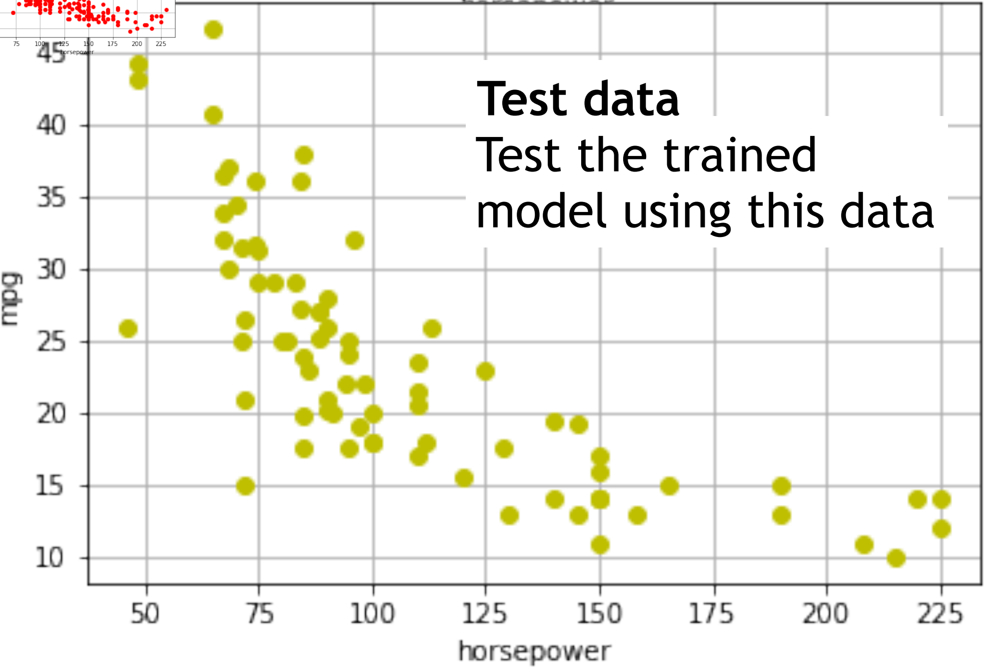
Training data

Train(fit) model using this data

Find the model that best fits **all** the training data

Determine $\hat{w}$ our estimated model parameters

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{|\text{training}|} \sum_{j \in \text{training}} \left(\text{mpg}^{(j)} - (w_0 + w_1 \text{horsepower}^{(j)}) \right)^2$$



Test data

Test the trained model using this data

Estimate the generalization error, $E_{out}(w)$, by using the test data

$$E_{\text{test}}(\mathbf{w}) = \frac{1}{|\text{test}|} \sum_{j \in \text{test}} \left(\text{mpg}^{(j)} - (w_0 + w_1 \text{horsepower}^{(j)}) \right)^2$$

For binary classification, how good is our estimate for E_{out}

Is $|E_{out} - E_{test}|$ likely to be small?

"Hoeffding's inequality is a powerful technique—perhaps the most important inequality in learning theory"

from http://cs229.stanford.edu/extra-notes/hoeffding.pdf

Generalization Bound for classification

iid: each example "has the same probability distribution as the others and all are mutually independent."

Suppose our test set contained K randomly chosen examples

then by using Hoeffding's inequality

the probability our E_{out} differs from E_{test} by more than $\epsilon > 0$ occurs with probability at most $2e^{-2\epsilon^2 K}$

Example:

If K=500 and $\epsilon = 0.1$, then setting $\delta = 2e^{-2(0.1)^2(500)} = 0.0001$ then with probability $1 - \delta$ the true error is within 0.1 of the average error on the test set.

Generaliz

proof) ... size K, where each ... ed in [a,b]] the ... obability that the average value, v, of the random ... deviate from its average μ by more than ϵ is:

$$P\big[\,|v-\mu|>\epsilon\big]\le 2e^{-2\epsilon^2K/(b-a)^2}=\delta \text{ for any } \epsilon>0$$

Thus if $K \ge \dfrac{\log(2/\delta)(b-a)^2}{2\epsilon^2}$ then with probability $1-\delta$

v is ϵ close to μ

Example:

Let g be a binary classifier (g outputs 0,1), let v be the average error of g on the test set of size K, and let μ be the true error of g. The probability that $|v-\mu|>\epsilon$ is at most $2e^{-2\epsilon^2K}$

If K=500 and $\epsilon=0.1$, then setting $\delta=2e^{-2(0.1)^2(500)}$ then with probability $1-\delta$ the true error is within 0.1 of the average error on the test set.

Generalization

Cannot get a range - instead get a confidence interval

Hoeffding inequality (stated without proof) for any sample size K, where each random variable is bounded in [a,b] the probability that the average value, v, of the random variables will deviate from its average μ by more than ϵ is:

$$P\left[\,|v-\mu| > \epsilon\right] \le 2e^{-2\epsilon^2 K/(b-a)^2} = \delta \text{ for any } \epsilon > 0$$

Thus if $K \geq \frac{\log(2/\delta)(b-a)^2}{2\epsilon^2}$ then with probability $1-\delta$

v is ϵ close to μ

We are assuming the K examples are drawn iid from a distribution

Example:
Let g be a binary classifier (g outputs 0,1), let v be the average error of g on the test set of size K, and let μ be the true error of g. The probability that $|v-\mu| > \epsilon$ is at most $2e^{-2\epsilon^2 K}$

If K=500 and $\epsilon = 0.1$, then setting $\delta = 2e^{-2(0.1)^2(500)}$ then with probability $1-\delta$ the true error is within 0.1 of the average error on the test set.

Generalization

Hoeffding inequality (stated without proof) for any sample size K, where each random variable is bounded in [a,b]] the probability that the average value, v, of the random variables will deviate from its average μ by more that ϵ is:

$$P\left[\,|v-\mu|>\epsilon\right]\leq 2e^{-2\epsilon^2 K/(b-a)^2}=\delta \text{ for any } \epsilon>0$$

Thus if $K \geq \dfrac{\log(2/\delta)(b-a)^2}{2\epsilon^2}$ then with probability $1-\delta$

We are assuming the K examples are drawn iid from a distribution

v is ϵ close to μ

Example:

Let g be a binary classifier (g outputs 0,1), let v be the average error of g on the test set of size K, and let μ be the true error of g. The probability that $|v-\mu|>\epsilon$ is at most $2e^{-2\epsilon^2 K}$

If K=100 and $\epsilon=0.2$, then $\underline{\delta=2e^{-2\cdot(0.2)^2\cdot 100}}$. With probability $1-\delta$ = 0.999 our estimated test set error is within 0.2 of the out of sample error

Outline

- ❑Motivating example: What polynomial degree should a [obscured]e?
- ❑Polynomial transformation
- ❑Underfitting and overfitting
- ❑Understanding error: Bias and varianc[obscured]
- ❑Learning curves
- ❑validation and model selection
- ❑Model selection (with limit[obscured]
- ❑ K-fold cross validation

- ❑Regularization

Yea!

Uh oh….

How to create a more complex hypothesis

Understanding what went wrong

Understanding where the error comes from, and how to estimate $E_{\text{out}}[g(\mathbf{x})]$

Our strategy

If we have many different hypothesis classes to choose from - how can we choose wisely? And how can we estimate $E_{\text{out}}[g(\mathbf{x})]$?

Estimating the generalization error:

One model:

- Data → Training, Test

Comparing several models and/or different hyper-parameters:

- Data → Training, Validation, Test
- Data → Training, Validation
- Data → k-fold cross Validation, Test
- Data → k-fold cross Validation

How to choose the best model (aka hypothesis class)

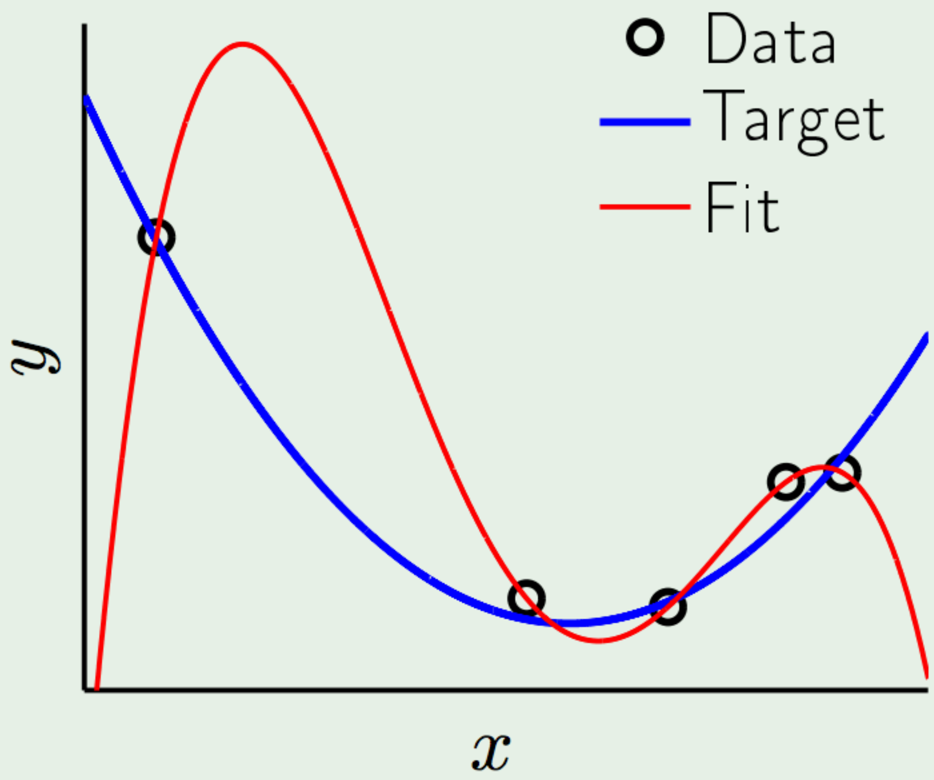
How to prevent overfitting?

Two cures

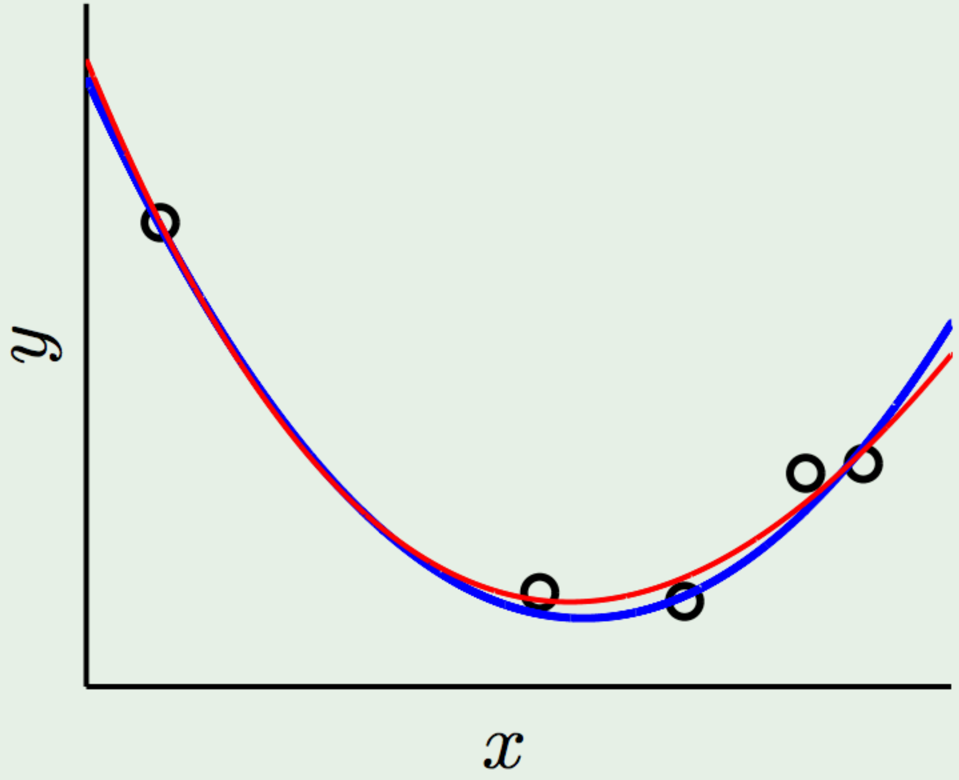
Regularization: Putting the brakes

Validation: Checking the bottom line

Putting the brakes



free fit



restrained fit

How do we choose the degree of the polynomial to avoid overfitting or underfitting?

WE NEED TO “TUNE” THE MODEL PARAMETER

Example Question

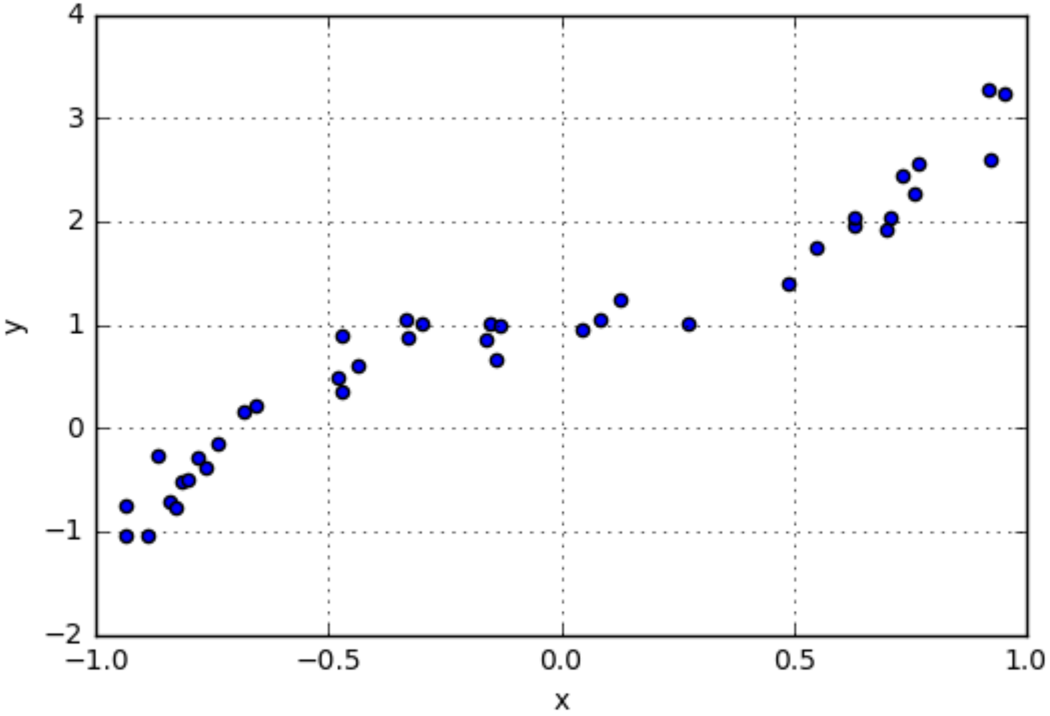
- You are given some data. The data has only one feature.
- You decide to find the polynomial transformation that best fits your data

$$\hat{y}^{(i)} = \tilde{\mathbf{w}}^T \Phi_d(\mathbf{x}^{(i)})$$

$$\hat{y}^{(i)} = \tilde{w}_0 + \tilde{w}_1 x^{(i)} + \tilde{w}_2 x^{(i)2} + \cdots + \tilde{w}_d x^{(i)d}$$

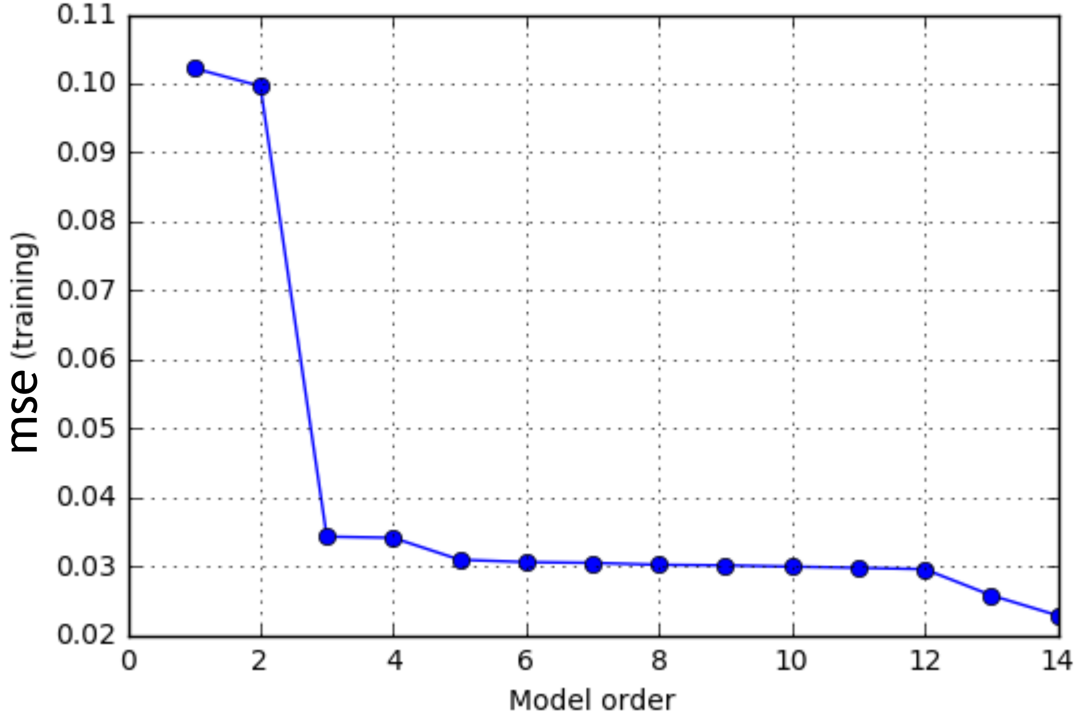
- What model order d should you use?

Thoughts?



Using RSS on Training Data?

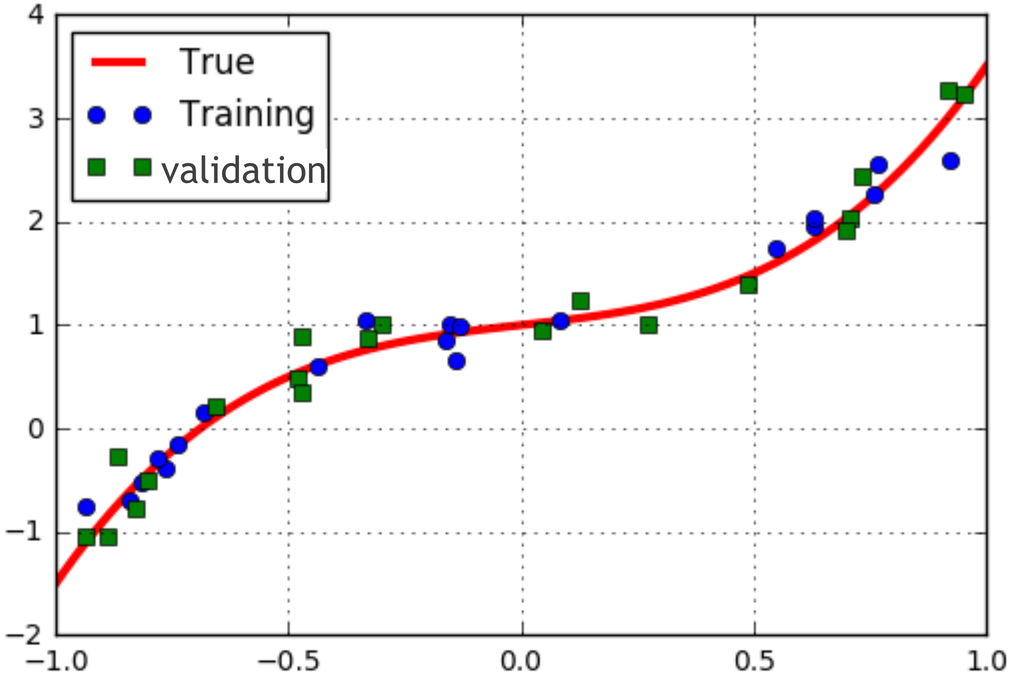
- Simple (but bad) idea:
  - For each model order, d,
    1. Compute $\tilde{\mathbf{w}}$ on transformed data, $\Phi_d(\mathbf{x})$. Predict labels on the transformed training data,
    $$\hat{y}^{(i)} = \tilde{\mathbf{w}}^T \Phi_d(\mathbf{x})$$
    2. Compute MSE $$MSE(d) = \frac{1}{N}\sum_i (y^{(i)} - \hat{y}^{(i)})^2$$
    3. Find d with lowest MSE
- This doesn't work
  - MSE(d) is always decreasing (Question: **Why**?)
  - Minimizing MSE(d) will pick d as large as possible
  - Leads to overfitting
- What went wrong?
- How can we do better?



Polynomial Example: Training Validation Split

- Example: Split data into 20 samples for training, 20 for validation

Shuffle your data before splitting it into training, validation and test data



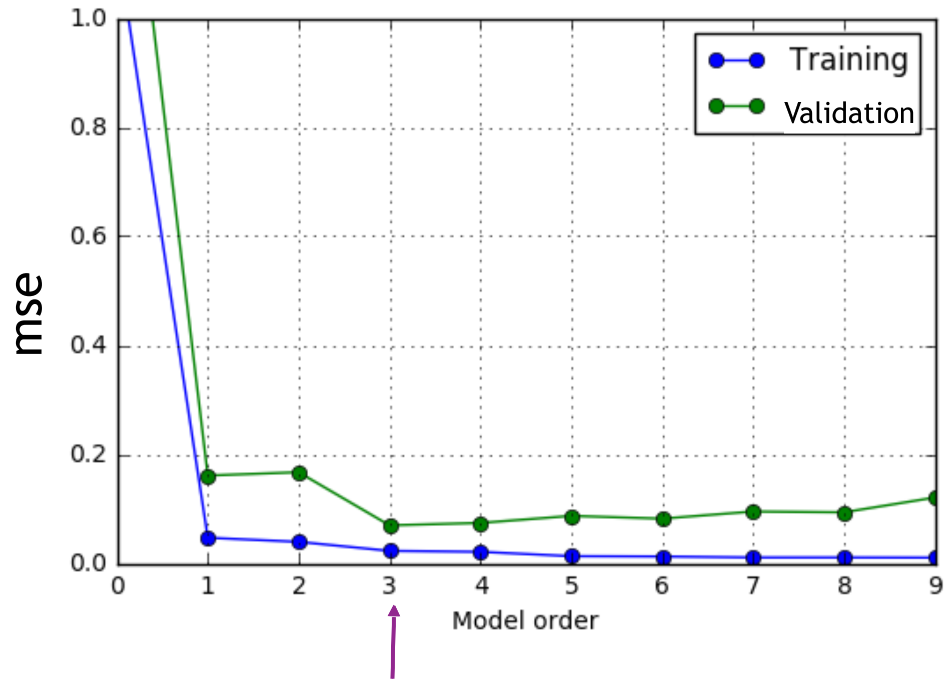

```
# Number of samples for training and Validation
ntr = nsamp // 2
nts = nsamp - ntr

# Training
xtr = xdat[:ntr]
ytr = ydat[:ntr]

# Validation
xVal = xdat[ntr:]
yVal = ydat[ntr:]
```

Finding the Model Order

- ❑Estimated optimal model order = 3



MSE validation minimized at 3
MSE training always decreases

Model selection with lots of data

Shuffle your data before splitting it into training, validation and test data

Each model has its own weights/parameters. We are using a subscript to distinguish the different weights/parameters for the different models

- For each model (e.g. degree d)
  - **train** on the **training data** to find **parameters** $\mathbf{w}_d$
  - Estimate the **error** of $\mathbf{w}_d$ on the **validation data**
- Pick the best performing model (hypothesis) to be the model with the lowest validation error (e.g. call best degree d^*)
- **Estimate out of sample error** of the best model E_{out} using **test data** (e.g. $\mathbf{w}_d$)
- Typical splits:

test data	validation data	
10%	10%	80%
25%	25%	50%

https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch03.pdf

Model selection with lo

For each model we will discuss, we show how to make it more or less "flexible"

Shuffle your data before splitting it into training, validation and test data

Each model has its own weights/parameters. We are using a subscript to distinguish the different weights/parameters for the different models

- For each model (e.g. degree d)
  - **train** on the **training data** to find **parameters** $\mathbf{w}_d$
  - Estimate the **error** of $\mathbf{w}_d$ on the **validation data**
- Pick the best performing model (hypothesis) to be the model with the lowest validation error (e.g. call best degree d^*)
- **Estimate out of sample error** of the best model E_{out} using **test data** (e.g. $\mathbf{w}_d$)
- Typical splits:

test data	validation data	
10%	10%	80%
25%	25%	50%

https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch03.pdf

Outline

- Motivating example: What polynomial degree should a …e?
- Polynomial transformation
- Underfitting and overfitting
- Understanding error: Bias and varian…
- Learning curves
- validation and model selection
- Model selection (with li…
- K-fold cross validation
- Regularization

Yea!

Uh oh….

How to create a more complex hypothesis

Understanding what wrong

Understanding where the error comes from, and how to estimate $E_{\text{out}}[g(\mathbf{x})]$

Understanding what went wrong

If we have many different hypothesis classes to choose from - how can we choose wisely? And how can we estimate $E_{\text{out}}[g(\mathbf{x})]$?

Thought experiment

1. If we have enough examples in the validation set, is e is a good estimate of E_out?

- yes, it is unbiased
- it is an optimistic estimate, but relatively good estimate
- no, it is not a good estimate

Two hypothesis h_1, h_2

$$E_{\text{out}}(h_1) = E_{\text{out}}(h_2) = \frac{1}{2}$$

Given the error e_1, e_2 estimates for the hypothesis
where we assume (for this thought experiment) that e_1, e_2 is uniform on $[0,1]$

pick $h \in \{h_1, h_2\}$ where $e = \min(e_1, e_2)$

This example comes from Abu-Mostafa

Thought experiment

e_1	e_2	$e = \min\{e_1, e_2\}$
$e_1 > 0.5$	$e_2 > 0.5$	$e > 0.5$
$e_1 < 0.5$	$e_2 > 0.5$	$e < 0.5$
$e_1 > 0.5$	$e_2 < 0.5$	$e < 0.5$
$e_1 < 0.5$	$e_2 < 0.5$	$e < 0.5$

Two hypothesis h_1, h_2

$$E_{\text{out}}(h_1) = E_{\text{out}}(h_2) = \frac{1}{2}$$

Given the error e_1, e_2 estimates for the hypothesis
where we assume (for this thought experiment) that e_1, e_2 is uniform on $[0,1]$

pick $h \in \{h_1, h_2\}$ where $e = \min(e_1, e_2)$

Notice that $E[e] \leq 0.5$

We have an optimistic biased estimate of the error if we estimate

This example comes from Abu-Mostafa

The next slide was not presented in class

Generalization

Training Error

$$E_{\text{in}}(w_0, w_1) = \underbrace{\frac{1}{N}\sum_{i=1}^{N}}_{\text{Average error on the N training examples}} \overbrace{\text{error}}^{\text{Cost we chose for not being the same as true label}}\big(y^{(i)}, \underbrace{g(\mathbf{x}^{(i)})}_{\text{Prediction on input } \mathbf{x}^{(i)}}\big) = \frac{1}{N}\sum_{i=1}^{N}\Big(y^{(i)} - \underbrace{(w_0 + w_1\mathbf{x}^{(i)})}_{\text{Prediction on input } \mathbf{x}^{(i)}}\Big)^2$$

MSE over the training data is called the "in sample" error.

Generalization Error

$$E_{\text{out}}(w_0, w_1) = E_{\mathbf{x},y}\left[\text{error}(y, g(\mathbf{x}))\right]$$

Assumption is training data is from the same distribution as the hypothesis will be used

Expected error when the model is used on new examples. It is called the "out of sample error".

We cannot compute this value.

Expectation taken over all possible input/labels and the probability that input/label is seen

Testing Error

$$E_{\text{test}}(w_0, w_1) = \underbrace{\frac{1}{N'}\sum_{i=1}^{N'}}_{\text{Average error on the N testing g examples}} \overbrace{\text{error}}^{\text{Cost we chose for not being the same as true label}}\big(y^{(i)}, \underbrace{g(\mathbf{x}^{(i)})}_{\text{Prediction on input } \mathbf{x}^{(i)}}\big) = \frac{1}{N}\sum_{i=1}^{N}\Big(y^{(i)} - (w_0 + w_1\mathbf{x}^{(i)})\Big)^2$$

Unbiased estimate of the generalization error (the out of sample error)