We’re trying to plan future ML course offerings, and I would like some feedback on HW0. Please take this anonymous poll (also linked to on Slack). Thank you! https://tinyurl.com/ybhr5dfn

We have a Slack channel.
Whether you are registered or not, please join: https://tinyurl.com/y97uha42

\( \theta \) is uniform on \([0, \theta]\) for unknown \( \theta \). Observe \( U_1, \ldots, U_n \).

1) What is \( \hat{\theta}_{\text{MLE}} \)?

2) Suppose given a prior \( P(\theta) = \begin{cases} \frac{1}{\theta^2} & 0 < \theta < 1 \\ 0 & \text{otherwise} \end{cases} \)
Linear Regression

Machine Learning – CSE546
Kevin Jamieson
University of Washington

Oct 5, 2017
given past sales data on zillow.com, predict:

\[ y = \text{House sale price from} \]
\[ x = \{ \# \text{ sq. ft., zip code, date of sale, etc.} \} \]

Training Data:
\[ \{(x_i, y_i)\}_{i=1}^n \]

\[ x_i \in \mathbb{R}^d \]
\[ y_i \in \mathbb{R} \]
The regression problem

Given past sales data on zillow.com, predict:

\[ y = \text{House sale price} \text{ from } \]
\[ x = \{ \text{# sq. ft., zip code, date of sale, etc.} \} \]

Training Data:
\[ \{(x_i, y_i)\}_{i=1}^{n} \]

Hypothesis: linear
\[ y_i \approx x_i^T w \]

Loss: least squares
\[ \min_w \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]
The regression problem in matrix notation

\[ \hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 \]

\[ = \arg \min_w (y - Xw)^T (y - Xw) \]

\[ y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \]
The regression problem in matrix notation

\[
\hat{w}_{LS} = \arg \min_{\mathbf{w}} \| \mathbf{y} - \mathbf{Xw} \|^2_2 \\
= \arg \min_{\mathbf{w}} (\mathbf{y} - \mathbf{Xw})^T (\mathbf{y} - \mathbf{Xw})
\]
The regression problem in matrix notation

\[ \hat{w}_{LS} = \arg \min_w ||y - Xw||^2_2 \]
\[ = (X^T X)^{-1} X^T y \]

What about an offset?

\[ \hat{w}_{LS}, \hat{b}_{LS} = \arg \min_{w, b} \sum_{i=1}^n (y_i - (x_i^T w + b))^2 \]
\[ = \arg \min_{w, b} ||y - (Xw + 1b)||^2_2 \]
Dealing with an offset

\[ \mathbb{1} = n = \sum_{i=1}^{n} y_i^2 \]

\[ \nabla f(x,y,z) = \left[ \frac{\partial f(x,y,z)}{\partial w}, \frac{\partial f(x,y,z)}{\partial b} \right] = \arg \min_{w,b} \|y - (Xw + 1b)\|_2^2 \]

\[ X = \begin{bmatrix} x_1^T \\ x_2^T \end{bmatrix} \]

\[ \nabla_b c = 0 = -1^T(y - (Xw - 1b)) = -1^Ty + 1^TXw + 1^T1b \]

\[ b = \frac{1}{n}(1^Ty - 1^TXw) = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^Tw) \]
Dealing with an offset

\[ \hat{w}_{LS}, \hat{b}_{LS} = \arg \min_{w,b} \| y - (Xw + 1b) \|^2_2 \]

\[ X^T X \hat{w}_{LS} + \hat{b}_{LS} X^T 1 = X^T y \]
\[ 1^T X \hat{w}_{LS} + \hat{b}_{LS} 1^T 1 = 1^T y \]

If \( X^T 1 = 0 \) (i.e., if each feature is mean-zero) then

\[ \hat{w}_{LS} = (X^T X)^{-1} X^T Y \]
\[ \hat{b}_{LS} = \frac{1}{n} \sum_{i=1}^{n} y_i \]
The regression problem in matrix notation

\[ \hat{w}_{LS} = \arg \min_w \| y - Xw \|^2_2 \]
\[ = (X^T X)^{-1} X^T y \]

But why least squares?

Consider \( y_i = x_i^T w + \epsilon_i \) where \( \epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2) \)

\[
P(y|x, w, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( - \frac{(y - x^T w)^2}{2\sigma^2} \right)
\]
Maximizing log-likelihood

Maximize:

$$\log P(D|w, \sigma) = \log \left( \frac{1}{\sqrt{2\pi\sigma}} \right)^n \prod_{i=1}^{n} e^{-\frac{(y_i - x_i^T w)^2}{2\sigma^2}}$$

$$\hat{w}_{MLE} = (X^T X)^{-1} X^T y$$
MLE is LS under linear model

\[ \hat{w}_{LS} = \arg \min_w \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]

\[ \hat{w}_{MLE} = \arg \max_w P(D|w, \sigma) \]

if \( y_i = x_i^T w + \epsilon_i \) and \( \epsilon_i \sim \text{i.i.d. } \mathcal{N}(0, \sigma^2) \)

\[ \hat{w}_{LS} = \hat{w}_{MLE} = (X^T X)^{-1} X^T Y \]
The regression problem

Given past sales data on zillow.com, predict:

\[ y = \text{House sale price} \quad \text{from} \]
\[ x = \{\# \text{ sq. ft., zip code, date of sale, etc.}\} \]

Training Data:
\[ \{(x_i, y_i)\}_{i=1}^{n} \]

Hypothesis: linear

\[ y_i \approx x_i^T w \]

Loss: least squares

\[ \min_w \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]
The regression problem

Given past sales data on zillow.com, predict:

\[ y = \text{House sale price} \] from \[ x = \{ \# \text{ sq. ft.}, \text{zip code}, \text{date of sale}, \text{etc.} \} \]

Training Data:

\[ \{(x_i, y_i)\}_{i=1}^n \]

Hypothesis: linear

\[ y_i \approx x_i^T w \]

Loss: least squares

\[ \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 \]
The regression problem

Training Data:
\[ \{ (x_i, y_i) \}_{i=1}^{n} \]

Hypothesis: linear
\[ y_i \approx x_i^T w \]

Loss: least squares
\[ \min_w \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]
The regression problem

Training Data:
\[ \{ (x_i, y_i) \}_{i=1}^n \]
\[ x_i \in \mathbb{R}^d \]
\[ y_i \in \mathbb{R} \]

Hypothesis: linear
\[ y_i \approx x_i^T w \]

Loss: least squares
\[ \min_w \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]

Transformed data:
\[ h : \mathbb{R}^d \rightarrow \mathbb{R}^p \] maps original features to a rich, possibly high-dimensional space

in \( d=1 \):
\[ h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix} = \begin{bmatrix} x \\ x^2 \\ \vdots \\ x^p \end{bmatrix} \]

for \( d>1 \), generate \( \{ u_j \}_{j=1}^p \subset \mathbb{R}^d \)
\[ h_j(x) = \frac{1}{1 + \exp(u_j^T x)} \]
\[ h_j(x) = (u_j^T x)^2 \]
\[ h_j(x) = \cos(u_j^T x) \]
The regression problem

Training Data:
\[ \{(x_i, y_i)\}_{i=1}^n \]
where \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \)

Hypothesis: linear
\[ y_i \approx x_i^T w \]

Loss: least squares
\[ \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 \]

Transformed data:
\[ h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix} \]

Hypothesis: linear
\[ y_i \approx h(x_i)^T w \quad w \in \mathbb{R}^p \]

Loss: least squares
\[ \min_w \sum_{i=1}^n (y_i - h(x_i)^T w)^2 \]
The regression problem

Training Data: \( \{ (x_i, y_i) \}_{i=1}^{n} \)

\[ x_i \in \mathbb{R}^d \]
\[ y_i \in \mathbb{R} \]

Transformed data:

\[ h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix} \]

Hypothesis: linear

\[ y_i \approx h(x_i)^T w \quad w \in \mathbb{R}^p \]

Loss: least squares

\[ \min_w \sum_{i=1}^{n} (y_i - h(x_i)^T w)^2 \]
The regression problem

Training Data: \[ \{(x_i, y_i)\}_{i=1}^n \]
\[ x_i \in \mathbb{R}^d \]
\[ y_i \in \mathbb{R} \]

Transformed data: \[ h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix} \]

Hypothesis: linear

\[ y_i \approx h(x_i)^T w \quad w \in \mathbb{R}^p \]

Loss: least squares

\[ \min_w \sum_{i=1}^n (y_i - h(x_i)^T w)^2 \]

Sale Price

Date of sale

small \( p \) fit
The regression problem \( \mathbf{A} \mathbf{x} = \mathbf{b}, \; \mathbf{x} = \mathbf{A}^{-1} \mathbf{b} \)

Training Data:
\[
\{(x_i, y_i)\}_{i=1}^n
\]

\( x_i \in \mathbb{R}^d \)
\( y_i \in \mathbb{R} \)

Transformed data:
\[
h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix}
\]

Hypothesis: linear
\[
y_i \approx h(x_i)^T \mathbf{w}
\]

Loss: least squares
\[
\min_{\mathbf{w}} \sum_{i=1}^n (y_i - h(x_i)^T \mathbf{w})^2
\]

What's going on here?
Bias-Variance Tradeoff

Machine Learning – CSE546
Kevin Jamieson
University of Washington

Oct 5, 2017
$P_{XY}(X = x, Y = y)$
Statistical Learning

\[ P_{XY}(X = x, Y = y) \]

\[ P_{XY}(Y = y | X = x_0) \]

\[ P_{XY}(Y = y | X = x_1) \]
Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{XY}[Y|X = x]$$

$$P_{XY}(Y = y|X = x_0)$$

$$P_{XY}(Y = y|X = x_1)$$
Statistical Learning

Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{XY}[Y|X = x]$$
Ideally, we want to find:

\[ \eta(x) = \mathbb{E}_{XY} [Y | X = x] \]

But we only have samples:

\((x_i, y_i) \sim i.i.d. \quad P_{XY} \quad \text{for } i = 1, \ldots, n\)
Ideally, we want to find:

\[ \eta(x) = \mathbb{E}_{XY} [Y|X = x] \]

But we only have samples:

\[ (x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n \]

and are restricted to a function class (e.g., linear) so we compute:

\[
\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2
\]
Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{XY}[Y|X = x]$$

But we only have samples:

$$(x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n$$

and are restricted to a function class (e.g., linear) so we compute:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

We care about future predictions: $\mathbb{E}_{XY}[(Y - \hat{f}(X))^2]$
Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{XY}[Y|X = x]$$

But we only have samples:

$$(x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n$$

and are restricted to a function class (e.g., linear) so we compute:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Each draw $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ results in different $\hat{f}$
Statistical Learning

Statistical Learning

Each draw $D = \{(x_i, y_i)\}_{i=1}^{n}$ results in different $\hat{f}$

Ideally, we want to find:
$$\eta(x) = \mathbb{E}_{XY}[Y|X = x]$$

But we only have samples:
$$(x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n$$

and are restricted to a function class (e.g., linear) so we compute:
$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$
Bias-Variance Tradeoff

\[ \eta(x) = \mathbb{E}_{X,Y}[Y|X = x] \]

\[ \hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

\[ \mathbb{E}_{Y|X=x}[\mathbb{E}_{D}[(Y - \hat{f}_D(x))^2]] = \mathbb{E}_{Y|X=x}[\mathbb{E}_{D}[(Y - \eta(x) + \eta(x) - \hat{f}_D(x))^2]] \]

\[ = \mathbb{E}_{Y|X=x}\left[ \mathbb{E}_{D}\left[ \left( Y - \eta(x) \right)^2 + 2 \left( Y - \eta(x) \right) \left( \eta(x) - \hat{f}_D(x) \right) \right] \right] \]

\[ \text{does not depend on } D \]

\[ \mathbb{E}_{Y|X=x}[\mathbb{E}_{D}[(Y - \hat{f}_D(x))^2]] \text{ does not depend on } Y \]

\[ = \mathbb{E}_{Y|X=x}\left[ (Y - \eta(x))^2 \right] + \mathbb{E}_{D}\left[ (\eta(x) - \hat{f}_D(x))^2 \right] \]
Bias-Variance Tradeoff

\[ \eta(x) = \mathbb{E}_{XY}[Y \mid X = x] \]

\[ \hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

\[ \mathbb{E}_{XY} \left[ \mathbb{E}_D[ (Y - \hat{f}_D(x))^2 ] \mid X = x \right] = \mathbb{E}_{XY} \left[ \mathbb{E}_D[ (Y - \eta(x) + \eta(x) - \hat{f}_D(x))^2 ] \mid X = x \right] \]

\[ = \mathbb{E}_{XY} \left[ \mathbb{E}_D[ (Y - \eta(x))^2 + 2(Y - \eta(x))(\eta(x) - \hat{f}_D(x)) + (\eta(x) - \hat{f}_D(x))^2 ] \mid X = x \right] \]

\[ = \mathbb{E}_{XY}[(Y - \eta(x))^2 \mid X = x] + \mathbb{E}_D[(\eta(x) - \hat{f}_D(x))^2] \]

**irreducible error**

Caused by stochastic label noise

**learning error**

Caused by either using too “simple” of a model or not enough data to learn the model accurately
Bias-Variance Tradeoff

\[ \eta(x) = \mathbb{E}_{XY}[Y | X = x] \]

\[ \hat{f} = \text{arg min}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

\[ \mathbb{E}_D[(\eta(x) - \hat{f}_D(x))^2] = \mathbb{E}_D[(\eta(x) - \mathbb{E}_D[\hat{f}_D(x)] + \mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2] \]
Bias-Variance Tradeoff

\[ \eta(x) = \mathbb{E}_{XY}[Y \mid X = x] \]

\[ \hat{f} = \arg \min_{f \in F} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

\[ \mathbb{E}_D[(\eta(x) - \hat{f}_D(x))^2] = \mathbb{E}_D[(\eta(x) - \mathbb{E}_D[\hat{f}_D(x)] + \mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2] \]

\[ = \mathbb{E}_D[(\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])^2 + 2(\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])(\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))] \]

\[ + (\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2 \]

\[ = (\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])^2 + \mathbb{E}_D[(\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2] \]

biased squared variance
Bias-Variance Tradeoff

\[ E_{XY}[E_D[(Y - \hat{f}_D(x))^2]|X=x] = E_{XY}[(Y - \eta(x))^2|X=x] \]

- irreducible error
- biased squared
- variance

Model too simple \(\rightarrow\) high bias, cannot fit well to data

Model too complex \(\rightarrow\) high variance, small changes in data change learned function a lot
Bias-Variance Tradeoff

\[
\mathbb{E}_{XY} \left[ \mathbb{E}_D \left[ (Y - \hat{f}_D(x))^2 \right] \mid X = x \right] = \mathbb{E}_{XY} \left[ (Y - \eta(x))^2 \right] \mid X = x
\]

irreducible error

\[
+ (\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])^2 + \mathbb{E}_D \left[ (\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2 \right]
\]

biased squared variance

![Graph showing bias, variance, and total error with respect to complexity.](image-url)
Overfitting

Machine Learning – CSE546
Kevin Jamieson
University of Washington

Oct 5, 2017
Bias-Variance Tradeoff

- Choice of hypothesis class introduces learning bias
  - More complex class $\rightarrow$ less bias
  - More complex class $\rightarrow$ more variance
- But in practice??
Bias-Variance Tradeoff

- Choice of hypothesis class introduces learning bias
  - More complex class → less bias
  - More complex class → more variance
- But in practice??
- Before we saw how increasing the feature space can increase the complexity of the learned estimator:

\[
\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots
\]

\[
\hat{f}_D^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - f(x_i))^2
\]

Complexity grows as k grows
Training set error as a function of model complexity

\[ F_1 \subset F_2 \subset F_3 \subset \ldots \quad D \overset{i.i.d.}{\sim} P_{XY} \]

\[ \widehat{f}_{D}^{(k)} = \arg \min_{f \in F_k} \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - f(x_i))^2 \]

\textbf{TRAIN error:}

\[ \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - \widehat{f}_{D}^{(k)}(x_i))^2 \]

\textbf{TRUE error:}

\[ \mathbb{E}_{XY}[\{(Y - \widehat{f}_{D}^{(k)}(X))^2\}] \]
Training set error as a function of model complexity

\[ \mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots \quad \mathcal{D} \overset{i.i.d.}{\sim} P_{XY} \]

\[ \hat{f}_D^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - f(x_i))^2 \]

**TRAIN error:**
\[ \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - \hat{f}_D^{(k)}(x_i))^2 \]

**TRUE error:**
\[ \mathbb{E}_{XY}[(Y - \hat{f}_D^{(k)}(X))^2] \]

**TEST error:**
\[ \mathcal{T} \overset{i.i.d.}{\sim} P_{XY} \]
\[ \frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}_D^{(k)}(x_i))^2 \]

Important: \( \mathcal{D} \cap \mathcal{T} = \emptyset \)

©2017 Kevin Jamieson
Training set error as a function of model complexity

\[ F_1 \subset F_2 \subset F_3 \subset \ldots \]

\[ \hat{f}_D^{(k)} = \arg \min_{f \in F_k} \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - f(x_i))^2 \]

**TRAIN error:**

\[ \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - \hat{f}_D^{(k)}(x_i))^2 \]

**TRUE error:**

\[ \mathbb{E}_{XY}[(Y - \hat{f}_D^{(k)}(X))^2] \]

**TEST error:**

\[ \frac{1}{|T|} \sum_{(x_i, y_i) \in T} (y_i - \hat{f}_D^{(k)}(x_i))^2 \]

Each line is i.i.d. draw of \( D \) or \( T \)

**Important:** \( D \cap T = \emptyset \)
Training set error as a function of model complexity

\[ \mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots \quad \mathcal{D} \overset{i.i.d.}{\sim} P_{XY} \]

\[ \hat{f}_D^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2 \]

**TRAIN error** is optimistically biased because it is evaluated on the data it trained on. **TEST error** is unbiased only if \( T \) is never used to train the model or even pick the complexity \( k \).

**TRAIN error:**

\[ \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}_D^{(k)}(x_i))^2 \]

**TRUE error:**

\[ \mathbb{E}_{XY} [(Y - \hat{f}_D^{(k)}(X))^2] \]

**TEST error:**

\[ \frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}_D^{(k)}(x_i))^2 \]

Important: \( \mathcal{D} \cap \mathcal{T} = \emptyset \)
Test set error

- Given a dataset, **randomly** split it into two parts:
  - Training data: $\mathcal{D}$
  - Test data: $\mathcal{T}$

- Use **training data** to learn predictor
  - e.g.,
    \[
    \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}^{(k)}_\mathcal{D}(x_i))^2
    \]
  - use **training data** to pick complexity $k$ (next lecture)

- Use **test data** to report predicted performance
  \[
  \frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}^{(k)}_\mathcal{D}(x_i))^2
  \]
Overfitting: a learning algorithm overfits the training data if it outputs a solution $w$ when there exists another solution $w'$ such that:

$$[error_{\text{train}}(w) < error_{\text{train}}(w')] \land [error_{\text{true}}(w') < error_{\text{true}}(w)]$$
How many points do I use for training/testing?

- Very hard question to answer!
  - Too few training points, learned model is bad
  - Too few test points, you never know if you reached a good solution

- Bounds, such as Hoeffding’s inequality can help:
  \[ P(|\hat{\theta} - \theta^*| \geq \epsilon) \leq 2e^{-2N\epsilon^2} \]

- More on this later this quarter, but still hard to answer

- Typically:
  - If you have a reasonable amount of data 90/10 splits are common
  - If you have little data, then you need to get fancy (e.g., bootstrapping)
Recap

- Learning is…
  - Collect some data
    - E.g., housing info and sale price
  - Randomly split dataset into TRAIN and TEST
    - E.g., 80% and 20%, respectively
  - Choose a hypothesis class or model
    - E.g., linear
  - Choose a loss function
    - E.g., least squares
  - Choose an optimization procedure
    - E.g., set derivative to zero to obtain estimator
  - Justifying the accuracy of the estimate
    - E.g., report TEST error