Testing and Validating, Bias and Variance

Lecture 5 - damlf | ml1

human+machine INTELLIGENCE

Frankfurt School

Confronting Generalization Error

Suppose your h is accurate on a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ wrt with a **regularized cross-entropy loss** cost function, J:

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)})^2 + \lambda\sum_{j}^{n}\theta_j^2\right)$$

but this h makes unacceptably **large generalization errors** in out-of-sample predictions.

What should you do?

Confronting Generalization Error

Suppose your h is accurate on a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ wrt with a **regularized cross-entropy loss** cost function, J:

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h(x^{(i)}; \boldsymbol{\theta}) - y^{(i)})^2 + \lambda \sum_{j}^{n} \theta_j^2\right)$$

but this h makes unacceptably **large generalization errors** in out-of-sample predictions.

What should you do?

Get more training examples?

Confronting Generalization Error

Suppose your h is accurate on a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ wrt with a **regularized cross-entropy loss** cost function, J:

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)})^2 + \lambda\sum_{j}^{n}\theta_j^2\right)$$

but this h makes unacceptably **large generalization errors** in out-of-sample predictions.

What should you do?

Get more training examples?

Reduce the number of features?

Confronting Generalization Error

Suppose your h is accurate on a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ wrt with a **regularized cross-entropy loss** cost function, J:

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h(x^{(i)}; \boldsymbol{\theta}) - y^{(i)})^2 + \lambda \sum_{j}^{n} \theta_j^2\right)$$

but this h makes unacceptably **large generalization errors** in out-of-sample predictions.

What should you do?

Get more training examples?

Reduce the number of features?

Increase the number of features? $\{x_1, \ldots, x_n\} \cup \{x_{n+1}, \ldots, x_{n+q}\}$

Confronting Generalization Error

Suppose your h is accurate on a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ wrt with a **regularized cross-entropy loss** cost function, J:

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h(x^{(i)}; \boldsymbol{\theta}) - y^{(i)})^2 + \lambda \sum_{j}^{n} \theta_j^2\right)$$

but this h makes unacceptably **large generalization errors** in out-of-sample predictions.

What should you do?

Get more training examples?

Reduce the number of features?

Increase the number of features? $\{x_1, \ldots, x_n\} \cup \{x_{n+1}, \ldots, x_{n+q}\}$

Add polynomial features? $(x_1^2, x_1 x_2, \ldots, x_1 x_n, x_2^2, x_2 x_3, \ldots)$

Confronting Generalization Error

Suppose your h is accurate on a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ wrt with a **regularized cross-entropy loss** cost function, J:

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)})^2 + \lambda \sum_{j}^{n} \theta_j^2\right)$$

but this h makes unacceptably **large generalization errors** in out-of-sample predictions.

What should you do?

Get more training examples?

Reduce the number of features?

Increase the number of features? $\{x_1, \ldots, x_n\} \cup \{x_{n+1}, \ldots, x_{n+q}\}$

Add polynomial features? $(x_1^2, x_1x_2, \ldots, x_1x_n, x_2^2, x_2x_3, \ldots)$

Increase λ?

Decrease λ?

Diagnostics and their Use

A **diagnostic** is a routine you can run to detect a particular kind of error.

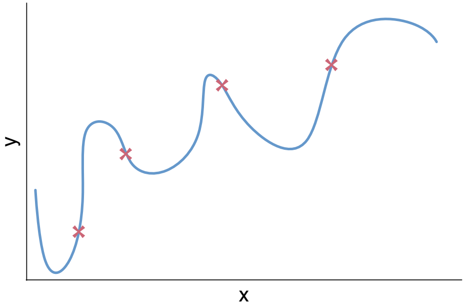
Diagnostics and their Use

A **diagnostic** is a routine you can run to detect a particular kind of error.

Implementing a diagnostic is often time consuming, but not nearly as time consuming as **randomly** going through the list of possible changes to your algorithm.

Evaluating your learned hypothesis

Evaluating your Learned Hypothesis

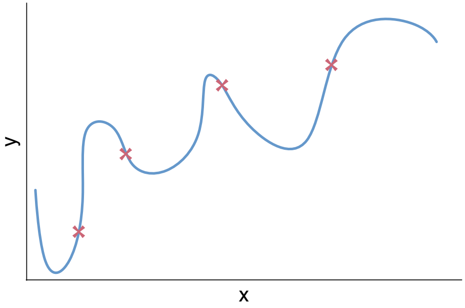


$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting

Previously, we discussed the problem of ***overfitting***, where **high training accuracy** fails to generalize to new, previously unobserved examples.

Evaluating your Learned Hypothesis



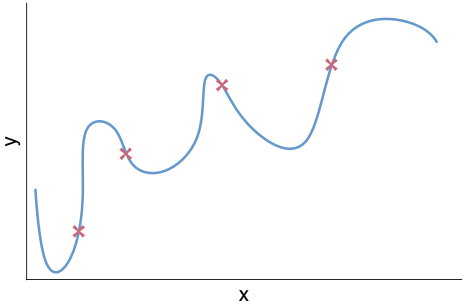
$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting

Previously, we discussed the problem of ***overfitting***, where **high training accuracy** fails to generalize to new, previously unobserved examples.

In simple **univariate** examples, we can simply plot the hypothesis.

Evaluating your Learned Hypothesis



$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting

Previously, we discussed the problem of ***overfitting***, where **high training accuracy** fails to generalize to new, previously unobserved examples.

In simple **univariate** examples, we can simply plot the hypothesis.

But, in **high-dimensional multi-feature** examples, directly visualizing overfitting is difficult if not impossible.

Evaluating your Learned Hypothesis

Overfitting

In **high-dimensional multi-feature** examples, plotting overfitting is difficult if not impossible:

$$
\begin{aligned}
x_1 &= \text{size } m^2 \\
x_2 &= \text{num of rooms} \\
x_3 &= \text{energy efficiency} \\
&\vdots \\
x_{100}
\end{aligned}
$$

Evaluating your Learned Hypothesis

Overfitting

In **high-dimensional multi-feature** examples, plotting overfitting is difficult if not impossible:

$$
\begin{aligned}
x_1 &= \text{size } m^2 \\
x_2 &= \text{num of rooms} \\
x_3 &= \text{energy efficiency} \\
&\vdots \\
x_{100}
\end{aligned}
$$

Let's return to our simple univariate regression problem to see how to evaluate models.

Evaluating your Learned Hypothesis

Data set

	Size in m^2 (x)	Price in Euros (y)
1)	52,14	164.220
2)	65,40	202.260
3)	65,38	261.710
4)	71,93	309.790
5)	82,69	384.490
6)	91,84	327.960
7)	94,83	418.900
8)	120,47	465.420
9)	100,20	622.150
10)	127,57	816.500

Model/Learning Hypothesis

$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$

How to evaluate $h(x; \boldsymbol{\theta})$:

Evaluating your Learned Hypothesis

Data set

	Size in m^2 (x)	Price in Euros (y)
1)	52,14	164.220
2)	65,40	202.260
3)	65,38	261.710
4)	71,93	309.790
5)	82,69	384.490
6)	91,84	327.960
7)	94,83	418.900
8)	120,47	465.420
9)	100,20	622.150
10)	127,57	816.500

Model/Learning Hypothesis

$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$

How to evaluate $h(x; \boldsymbol{\theta})$:

Split the data set:

- **Training Set** ($\approx$ 70%)
- **Test set** ($\approx$ 30%)

Evaluating your Learned Hypothesis

Data set

	Size in m^2 (x)	Price in Euros (y)	
1)	52,14	164.220	$(x^{(1)}, y^{(1)})$
2)	65,40	202.260	$(x^{(2)}, y^{(2)})$
3)	65,38	261.710	$(x^{(3)}, y^{(3)})$
4)	71,93	309.790	$(x^{(4)}, y^{(4)})$
5)	82,69	384.490	$(x^{(5)}, y^{(5)})$
6)	91,84	327.960	$(x^{(6)}, y^{(6)})$
7)	94,83	418.900	$(x^{(7)}, y^{(7)})$
8)	120,47	465.420	$(x_{test}^{(1)}, y_{test}^{(1)})$
9)	100,20	622.150	$(x_{test}^{(2)}, y_{test}^{(2)})$
10)	127,57	816.500	$(x_{test}^{(3)}, y_{test}^{(3)})$

Model/Learning Hypothesis

$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$

Notation:

m_{test} : denotes the number of **text examples**

Evaluating your Learned Hypothesis

Data set

	Size in m^2 (x)	Price in Euros (y)	
1)	52,14	164.220	$(x^{(1)}, y^{(1)})$
2)	65,40	202.260	$(x^{(2)}, y^{(2)})$
3)	65,38	261.710	$(x^{(3)}, y^{(3)})$
4)	71,93	309.790	$(x^{(4)}, y^{(4)})$
5)	82,69	384.490	$(x^{(5)}, y^{(5)})$
6)	91,84	327.960	$(x^{(6)}, y^{(6)})$
7)	94,83	418.900	$(x^{(7)}, y^{(7)})$
8)	120,47	465.420	$(x^{(1)}_{test}, y^{(1)}_{test})$
9)	100,20	622.150	$(x^{(2)}_{test}, y^{(2)}_{test})$
10)	127,57	816.500	$(x^{(3)}_{test}, y^{(3)}_{test})$

Model/Learning Hypothesis

$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$

Also:

You should **randomly** assign **70-30**% to the **training set** and **test set**, respectively.

Time series data is an exception – since data is time-dependent, by definition – thus handled differently.

Training & Testing Procedure for Linear Regression

1. Learn the vector $\hat{\theta}$ from **training set** by minimizing *training error*: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$.
2. Use this optimal $\hat{\theta}$ to compute the **test set** error:

Training & Testing Procedure for Linear Regression

1. Learn the vector $\hat{\boldsymbol{\theta}}$ from **training set** by minimizing *training error*: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$.
2. Use this optimal $\hat{\boldsymbol{\theta}}$ to compute the **test set** error:

$$J_{test}(\hat{\boldsymbol{\theta}}) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h(x_{test}^{(i)}; \hat{\boldsymbol{\theta}}) - y_{test}^{(i)} \right)^2$$

Training & Testing Procedure for Logistic Regression

1. Learn the vector $\hat{\theta}$ from **training set** by minimizing *training error*: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$.
2. Use this optimal $\hat{\theta}$ to compute the **test set** error with **cross-entropy loss**:

Training & Testing Procedure for Logistic Regression

1. Learn the vector $\hat{\theta}$ from **training set** by minimizing *training error*: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$.
2. Use this optimal $\hat{\theta}$ to compute the **test set** error with **cross-entropy loss**:

$$J_{test}(\hat{\boldsymbol{\theta}}) = -\frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h(x_{test}^{(i)}; \hat{\boldsymbol{\theta}}) \; + \; (1 - y_{test}^{(i)}) \log \left(1 - h(x_{test}^{(i)}; \hat{\boldsymbol{\theta}})\right)$$

- An alternative test-set performance metric is **0-1 misclassification error**:

0-1 misclassification error

Define the **error of a prediction** by

$$err\,(h(x;\boldsymbol{\theta}), y) = \begin{cases} 1 & \text{if } h(x;\boldsymbol{\theta}) \geqslant 0.5 \text{ and } y = 0 \quad \textit{or} \\ & \text{if } h(x;\boldsymbol{\theta}) < 0.5 \text{ and } y = 1 \\ 0 & \text{otherwise} \end{cases}$$

0-1 misclassification error

Define the **error of a prediction** by

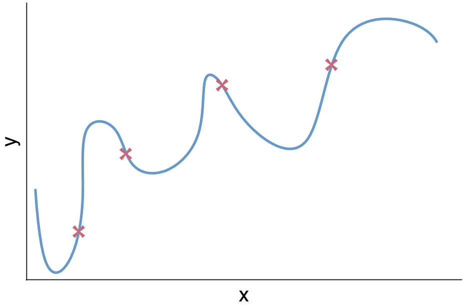
$$err\left(h(x;\boldsymbol{\theta}),y\right) = \begin{cases} 1 & \text{if } h(x;\boldsymbol{\theta}) \geqslant 0.5 \text{ and } y = 0 \quad \textit{or} \\ & \text{if } h(x;\boldsymbol{\theta}) < 0.5 \text{ and } y = 1 \\ 0 & \text{otherwise} \end{cases}$$

Define **0/1 misclassification test error** as:

$$\text{0/1 test error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err\left(h(x_{test}^{(i)};\boldsymbol{\theta}), y_{test}^{(i)}\right)$$

Rudiments of model selection

Evaluating your learned hypothesis

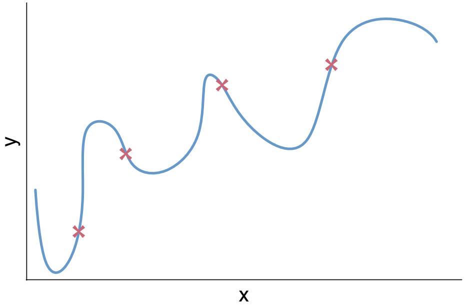


$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfitting

Training set error is **not** a good predictor of how the model will do on **new examples**.

Evaluating your learned hypothesis



$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

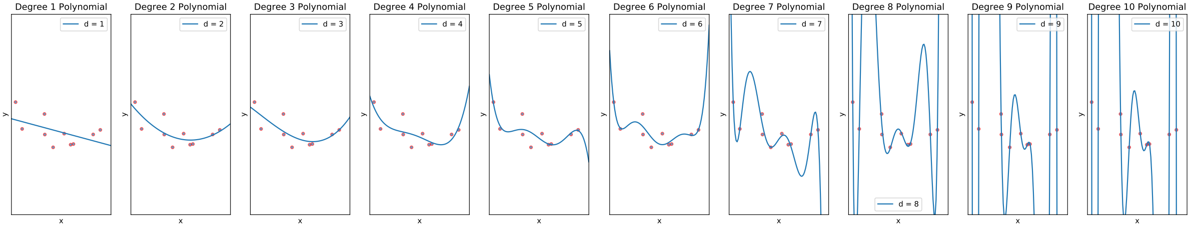
Overfitting

Training set error is **not** a good predictor of how the model will do on **new examples**.

That is, once parameters $\theta_0, \theta_1, \ldots, \theta_4$ are fit to the **training set** data, the error of the parameters – the training error $J(\theta)$ – is likely to be lower than the general **out-of-sample** error.

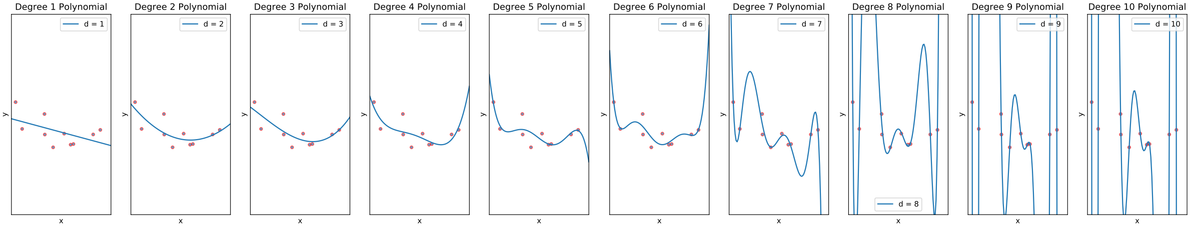
In general, the training error is a lousy predictor of general error.

Model Selection Problems



With this last point in mind, suppose you wish to **choose** what degree polynomial to fit to your data.

Model Selection Problems



With this last point in mind, suppose you wish to **choose** what degree polynomial to fit to your data.

This is a **model selection problem**.

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$		
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$		
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$		
	$\vdots$		
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$		

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(1)}$	
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$		
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$		
	$\vdots$		
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$		

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(1)}$	
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(2)}$	
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$		
	$\vdots$		
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$		

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(1)}$	
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(2)}$	
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(3)}$	
	$\vdots$		
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$		

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(1)}$	
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(2)}$	
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(3)}$	
	$\vdots$	$\vdots$	
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(10)}$	

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(1)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(1)})$
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(2)}$	
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(3)}$	
	$\vdots$	$\vdots$	
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(10)}$	

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(1)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(1)})$
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(2)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(2)})$
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(3)}$	
	$\vdots$	$\vdots$	
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(10)}$	

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(1)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(1)})$
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(2)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(2)})$
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(3)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(3)})$
	$\vdots$	$\vdots$	$\vdots$
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(10)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(10)})$

Model Selection

Suppose the parameter d denotes the **degree of polynomial** of $h(x; \boldsymbol{\theta})$.

d	Hypothesis	min train error	test set error
1.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(1)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(1)})$
2.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(2)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(2)})$
3.	$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(3)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(3)})$
	$\vdots$	$\vdots$	$\vdots$
10.	$h(x; \hat{\boldsymbol{\theta}}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \mapsto \hat{\boldsymbol{\theta}}^{(10)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(10)})$

Idea: Suppose I plug these optimal $\hat{\boldsymbol{\theta}}^{(d)}$ into $J_{test}(\cdot)$, then pick $J_{test}(\hat{\boldsymbol{\theta}}^{(d)})$ that is minimal.

Model Selection

d	Hypothesis	min train error	test set error
1.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\hat{\boldsymbol{\theta}}^{(1)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(1)})$
2.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\hat{\boldsymbol{\theta}}^{(2)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(2)})$
3.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\hat{\boldsymbol{\theta}}^{(3)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(3)})$
	$\vdots$	$\vdots$	$\vdots$
10.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\hat{\boldsymbol{\theta}}^{(10)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(10)})$

Suppose then $J_{test}(\hat{\boldsymbol{\theta}}^{(4)})$ has the **lowest** test error of all $J_{test}(\hat{\boldsymbol{\theta}}^{(d)})$, for $d \in \{1, 2, \ldots, 10\}$;
So I choose the $d = 4$ polynomial model parameterized by $\hat{\boldsymbol{\theta}}^{(4)} = [\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4]$.

Model Selection

d	Hypothesis	min train error	test set error
1.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\hat{\boldsymbol{\theta}}^{(1)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(1)})$
2.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\hat{\boldsymbol{\theta}}^{(2)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(2)})$
3.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\hat{\boldsymbol{\theta}}^{(3)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(3)})$
	$\vdots$	$\vdots$	$\vdots$
10.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\hat{\boldsymbol{\theta}}^{(10)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(10)})$

Suppose then $J_{test}(\hat{\boldsymbol{\theta}}^{(4)})$ has the **lowest** test error of all $J_{test}(\hat{\boldsymbol{\theta}}^{(d)})$, for $d \in \{1, 2, \ldots, 10\}$;

So I choose the $d = 4$ polynomial model parameterized by $\hat{\boldsymbol{\theta}}^{(4)} = [\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4]$.

Question: How well does this (d=4) fitted model generalize?

Model Selection

d	Hypothesis	min train error	test set error
1.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\hat{\boldsymbol{\theta}}^{(1)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(1)})$
2.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\hat{\boldsymbol{\theta}}^{(2)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(2)})$
3.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	$\hat{\boldsymbol{\theta}}^{(3)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(3)})$
	$\vdots$	$\vdots$	$\vdots$
10.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\hat{\boldsymbol{\theta}}^{(10)}$	$J_{test}(\hat{\boldsymbol{\theta}}^{(10)})$

Suppose then $J_{test}(\hat{\boldsymbol{\theta}}^{(4)})$ has the **lowest** test error of all $J_{test}(\hat{\boldsymbol{\theta}}^{(d)})$, for $d \in \{1, 2, \ldots, 10\}$;

So I choose the $d = 4$ polynomial model parameterized by $\hat{\boldsymbol{\theta}}^{(4)} = [\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4]$.

Question: How well does this (d=4) fitted model generalize?

To answer this question, you might look at the $J_{test}(\hat{\boldsymbol{\theta}}^{(4)})$ as an ***estimate*** of how well this 4th order polynomial will generalize to out of sample cases .

The problem with using $J_{test}(\hat{\theta}^{(4)})$ as an ***estimate*** of out-of-sample error is that it will not provide a **fair** estimate of generalization error.

The reason is that the extra parameter d (degree of polynomial) is **fit to the test set**; we chose the value of d that minimized $J_{test}(\hat{\theta}^{(4)})$; so, we've **fit** d to this specific set, X_{test}.

The problem with using $J_{test}(\hat{\theta}^{(4)})$ as an ***estimate*** of out-of-sample error is that it will not provide a **fair** estimate of generalization error.

The reason is that the extra parameter d (degree of polynomial) is **fit to the test set**; we chose the value of d that minimized $J_{test}(\hat{\theta}^{(4)})$; so, we've **fit** d to this specific set, x_{test}.

Intuitively, I've corrupted my test set by fitting the new parameter d to *that* test set.

The problem with using $J_{test}(\hat{\theta}^{(4)})$ as an ***estimate*** of out-of-sample error is that it will not provide a **fair** estimate of generalization error.

The reason is that the extra parameter *d* (degree of polynomial) is **fit to the test set**; we chose the value of *d* that minimized $J_{test}(\hat{\theta}^{(4)})$; so, we've **fit** *d* to this specific set, x_{test}.

Intuitively, I've corrupted my test set by fitting the new parameter *d* to *that* test set.

So, it is no longer fair to *evaluate* my hypothesis on this test set. My hypothesis is likely to do better on *this* test set than it would on new examples that it hasn't seen before.

To address this problem, we introduce a new “intermediate” category of set-aside data from our original training examples.

Evaluating your learned hypothesis

Data set

	Size in m^2 (x)	Price in Euros (y)	
1)	52,14	164.220	$(x^{(1)}, y^{(1)})$
2)	65,40	202.26	$(x^{(2)}, y^{(2)})$
3)	65,38	261.710	$(x^{(3)}, y^{(3)})$
4)	71,93	309.790	$(x^{(4)}, y^{(4)})$
5)	82,69	384.490	$(x^{(5)}, y^{(5)})$
6)	91,84	327.960	$(x^{(6)}, y^{(6)})$
7)	94,83	418.900	$(x_{cv}^{(1)}, y_{cv}^{(1)})$
8)	120,47	465.420	$(x_{cv}^{(2)}, y_{cv}^{(2)})$
9)	100,20	622.150	$(x_{test}^{(1)}, y_{test}^{(1)})$
10)	127,57	816.500	$(x_{test}^{(2)}, y_{test}^{(2)})$

Notation:

m : denotes the number of **training set** ($\approx$ 60%) ($\approx$ 99% for big data)

m_{cv}: denotes the number of **cross validation set** ($\approx$ 20%) ($\approx$ 0.5% for big data)

m_{test} : denotes the number of **test set** ($\approx$ 20%) ($\approx$ 0.5% for big data)

Training- Validation- and Test-errors

Training Error:

$$J_{train}(\boldsymbol{\theta}) = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x^{(i)}; \boldsymbol{\theta}) - y^{(i)} \right)^2$$

Cross Validation Error:

$$J_{cv}(\boldsymbol{\theta}) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} \left(h(x_{cv}^{(i)}; \boldsymbol{\theta}) - y_{cv}^{(i)} \right)^2$$

Test Error:

$$J_{test}(\boldsymbol{\theta}) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h(x_{test}^{(i)}; \boldsymbol{\theta}) - y_{test}^{(i)} \right)^2$$

Model Selection

Now let's return to our model selection problem.

d	Hypothesis	min train error	cv set error
1.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x$	$\hat{\boldsymbol{\theta}}^{(1)}$	$J_{cv}(\hat{\boldsymbol{\theta}}^{(1)})$
2.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$	$\hat{\boldsymbol{\theta}}^{(2)}$	$J_{cv}(\hat{\boldsymbol{\theta}}^{(2)})$
3.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_3 x^3$	$\hat{\boldsymbol{\theta}}^{(3)}$	$J_{cv}(\hat{\boldsymbol{\theta}}^{(3)})$
	$\vdots$	$\vdots$	$\vdots$
10.	$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$	$\hat{\boldsymbol{\theta}}^{(10)}$	$J_{cv}(\hat{\boldsymbol{\theta}}^{(2)})$

Choose: $h(x;\boldsymbol{\theta})$ with the lowest cross validation error

(i.e., smallest = $J_{cv}(\hat{\boldsymbol{\theta}}^{(d)})$ for $d \in \{1, \ldots, 10\}$) So, we fit d to J_{cv}

Now, we have set-aside test data to test the performance of this $\hat{\boldsymbol{\theta}}^{(d)}$.

Estimate generalization error for test set $J_{test}(\hat{\boldsymbol{\theta}}^{(d)})$, for d selected by $\min_{\hat{\boldsymbol{\theta}}^{(d)}} J_{cv}(\hat{\boldsymbol{\theta}}^{(d)})$.

Training, Cross-validation, Test

Model Selection

Our particular model selection problem is to select from the set of degree d polynomial models, $d = 1, 2, \ldots, 10$. Here is how a three-way split of your data into a training set, cross-validation set, and test set is used to achieve this:

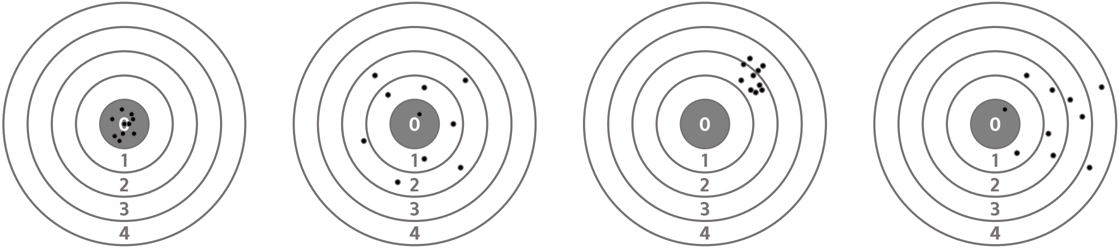
- **Training set**: used to minimize $\boldsymbol{\theta}$ wrt training data and cost function $J(\boldsymbol{\theta}^{(d)})$ for **each** model class $d = 1, 2, \ldots, 10$. So,
  - $\hat{\boldsymbol{\theta}}^{(1)}$ denotes the $\boldsymbol{\theta}$ that $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ for $h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$
  - $\hat{\boldsymbol{\theta}}^{(2)}$ denotes the $\boldsymbol{\theta}$ that $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ for $h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$

    $\vdots$
- **CV set**: used to pick which $J(\hat{\boldsymbol{\theta}}^{(1)}), J(\hat{\boldsymbol{\theta}}^{(2)}), \ldots, J(\hat{\boldsymbol{\theta}}^{(10)})$ that minimizes J_{cv} on the cv set. Just as the vector $\boldsymbol{\theta}$ is fit to the training set for each model class ($d = 1, \ldots, 10$), the model parameter d is fit to the cross validation set by selecting that $\hat{\boldsymbol{\theta}}^{(d)}$ which minimizes J_{cv} .
- **Test set**: used to evaluate the performance of $\hat{\boldsymbol{\theta}}^{(d)}$ on unseen test data.

How to Diagnose Bias and Variance

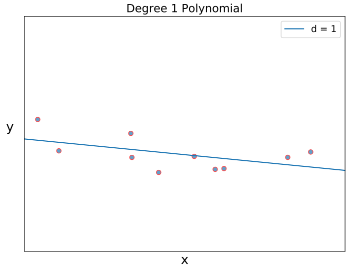
If you find that your algorithm is not performing well (on out of sample examples), in nearly all cases it is because your algorithm either **underfits** the training data or **overfits** the training data.

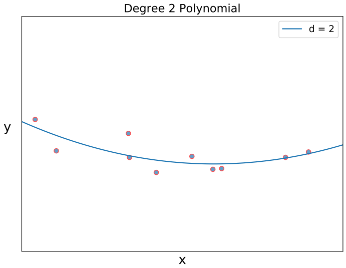
Bias and Variance

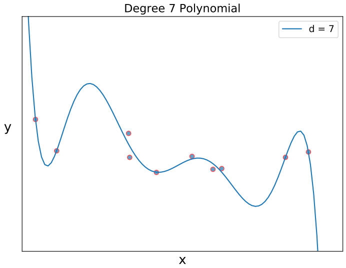


low bias & low variance | low bias & high variance | high bias & low variance | high bias & high variance

Bias/Variance Trade-off







$\theta_0 + \theta_1 x$

$\theta_0 + \theta_1 x + \theta_2 x^2$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7$$

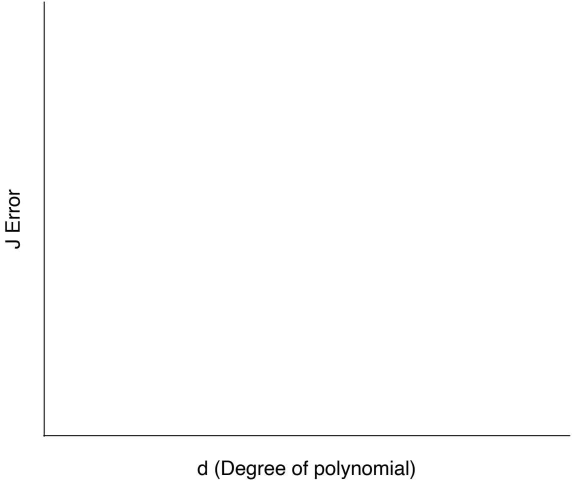
underfit: high bias

overfit: high variance

Bias/Variance Trade-off

Training Error: $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x_{train}^{(i)}; \boldsymbol{\theta}) - y_{train}^{(i)} \right)^2$

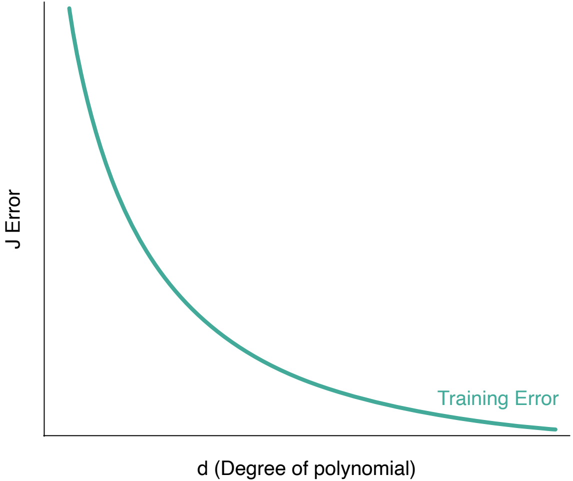
Cross Validation Error: $J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} \left(h(x_{cv}^{(i)}; \boldsymbol{\theta}) - y_{cv}^{(i)} \right)^2$



Bias/Variance Trade-off

Training Error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$

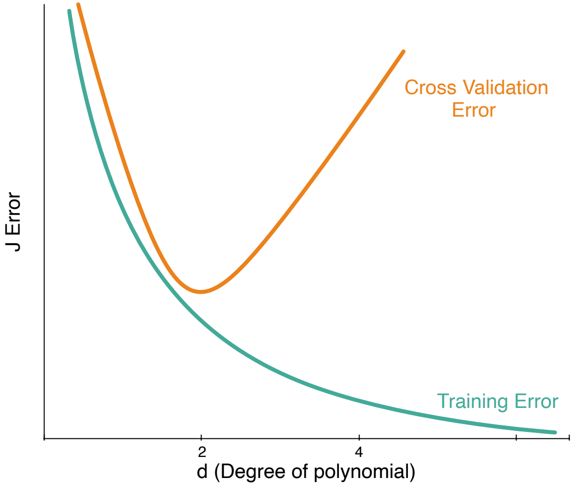
Cross Validation Error: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$



Bias/Variance Trade-off

Training Error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$

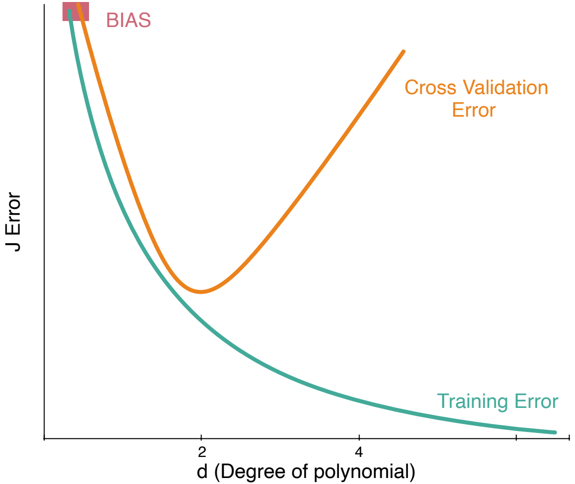
Cross Validation Error: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$



Bias/Variance Trade-off

Training Error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$

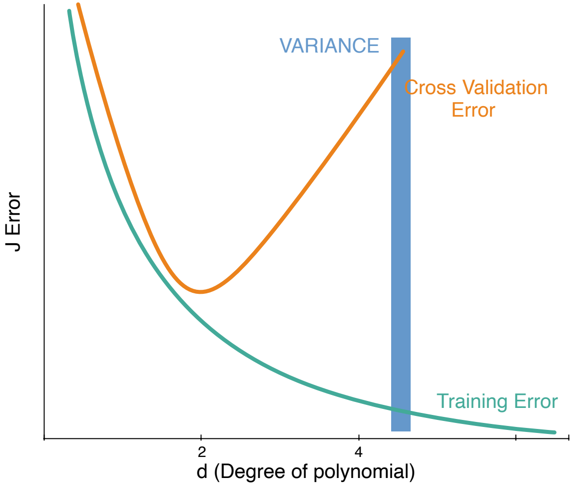
Cross Validation Error: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$



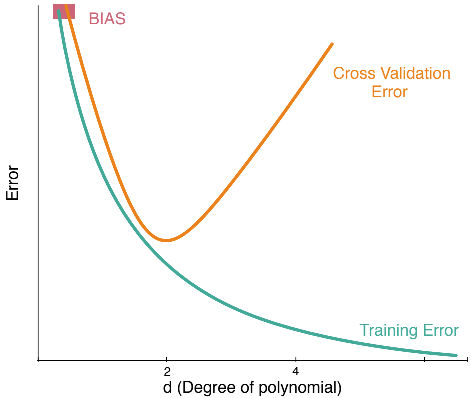
Bias/Variance Trade-off

Training Error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$

Cross Validation Error: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$



Diagnosing Bias vs Variance

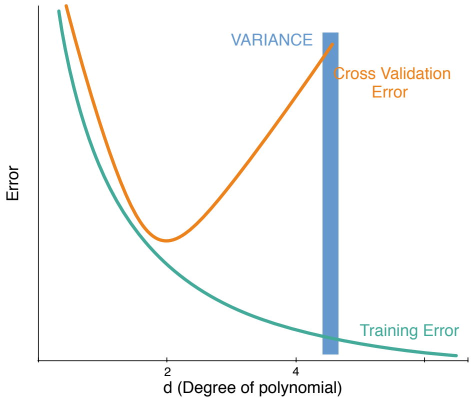


If **bias** (underfitting) is the problem:

$J_{train}(\theta)$ will be **high**

$J_{train}(\theta) \approx J_{cv}(\theta)$

Diagnosing Bias vs Variance



If **variance** (overfitting) is the problem:

$J_{train}(\theta)$ will be **low**

$J_{train}(\theta) \ll J_{cv}(\theta)$

Bias, Variance, and Regularization

Regularization

Suppose we have a high-degree polynomial hypothesis,

$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7$$

with

$$\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7]$$

and to prevent overfitting we introduce an L2 **regularized cost function**,

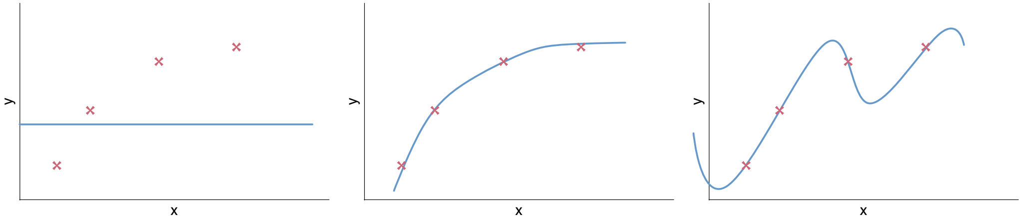
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x^{(i)}; \boldsymbol{\theta}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

Suppose we have a high-degree polynomial hypothesis,

$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7$$

and vary the regularization term λ in

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$



underfit: high bias

λ is large (e.g., 10^5)

overfit: high variance

λ is very small (e.g., 0)

How to choose λ

Model: $h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7$

Optimization Objective: $J(\boldsymbol{\theta}) = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x^{(i)}; \boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2$

How to choose λ

Model: $h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7$

Optimization Objective: $J(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$

Training Error: $J_{train}(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$

Cross Validation Error: $J_{cv}(\boldsymbol{\theta}) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$

Test Error: $J_{test}(\boldsymbol{\theta}) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}\left(h(x_{test}^{(i)};\boldsymbol{\theta}) - y_{test}^{(i)}\right)^2$

How to choose λ

Model: $h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7$

Optimization Objective: $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$

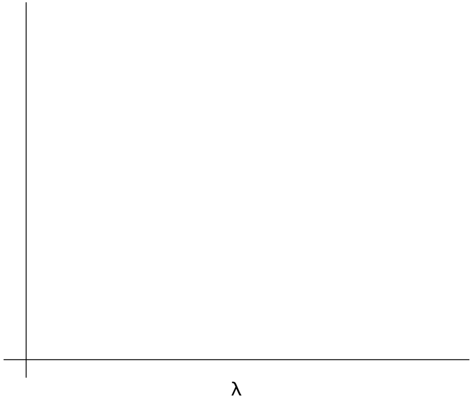
	Regularization parameter λ	min train error	cv set error
1.	Try $\lambda = 0$	$\min_\theta J(\theta) \mapsto \theta^{(1)}$	$J_{cv}(\theta^{(1)})$
2.	Try $\lambda = 0.01$	$\min_\theta J(\theta) \mapsto \theta^{(2)}$	$J_{cv}(\theta^{(2)})$
3.	Try $\lambda = 0.02$	$\min_\theta J(\theta) \mapsto \theta^{(3)}$	$J_{cv}(\theta^{(3)})$
4.	Try $\lambda = 0.04$	$\min_\theta J(\theta) \mapsto \theta^{(4)}$	$J_{cv}(\theta^{(4)})$
5.	Try $\lambda = 0.08$	$\min_\theta J(\theta) \mapsto \theta^{(5)}$	$J_{cv}(\theta^{(5)})$
	$\vdots$	$\vdots$	$\vdots$
10.	Try $\lambda = 10.24$	$\min_\theta J(\theta) \mapsto \theta^{(10)}$	$J_{cv}(\theta^{(10)})$

Bias/variance as a function of the parameter λ

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x^{(i)}; \boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x_{train}^{(i)}; \boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} \left(h(x_{cv}^{(i)}; \boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$

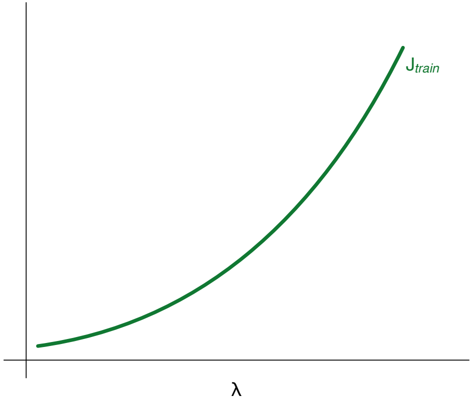


Bias/variance as a function of the parameter λ

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$

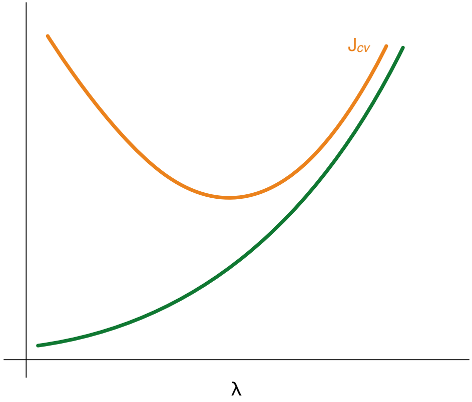


Bias/variance as a function of the parameter λ

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$

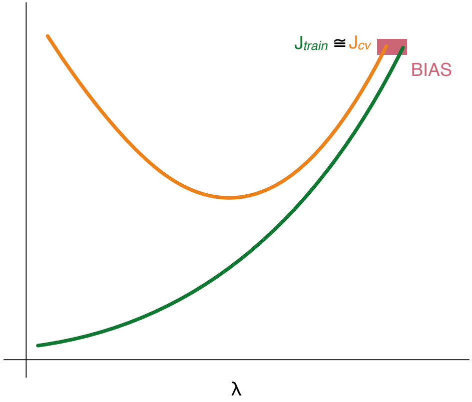


Bias/variance as a function of the parameter λ

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$

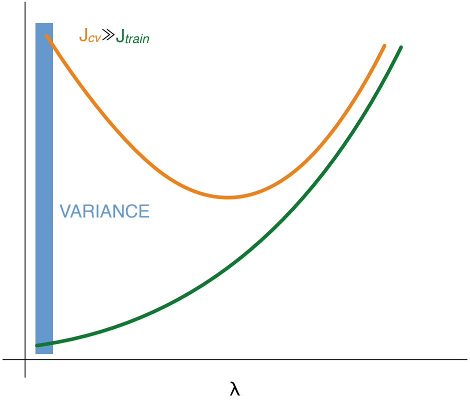


Bias/variance as a function of the parameter λ

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$

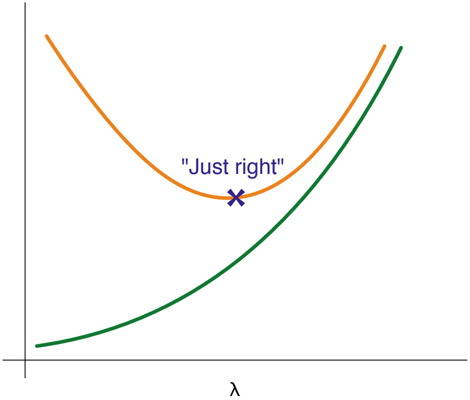


Bias/variance as a function of the parameter λ

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$



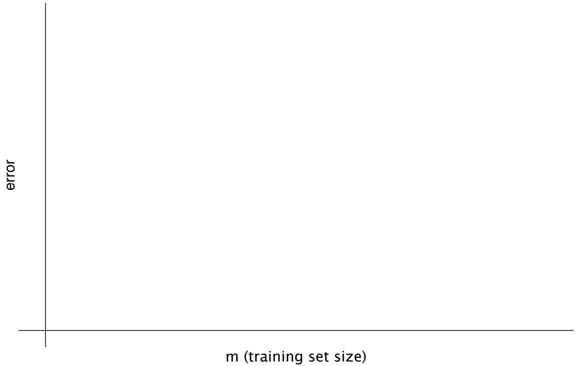
Bias/variance as a function of training set size *m*

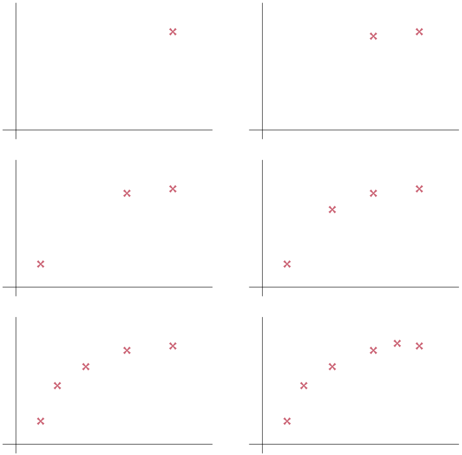
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$

$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$$





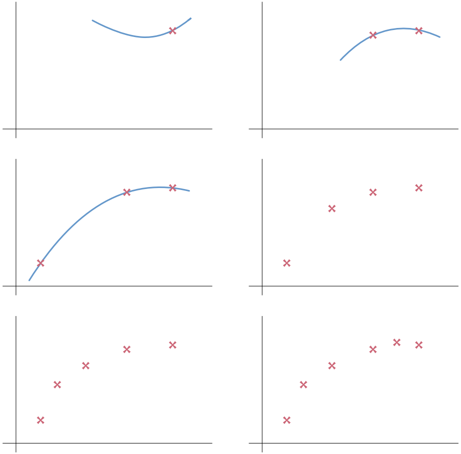
Bias/variance as a function of training set size *m*

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$

$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$$



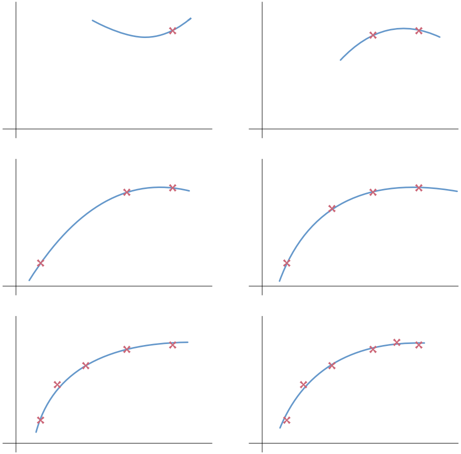
Bias/variance as a function of training set size *m*

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$

$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$$

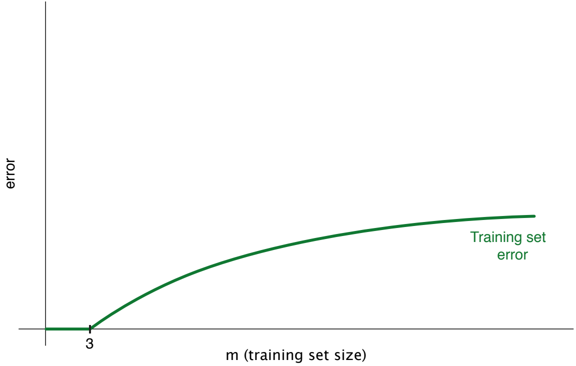


Bias/variance as a function of training set size *m*

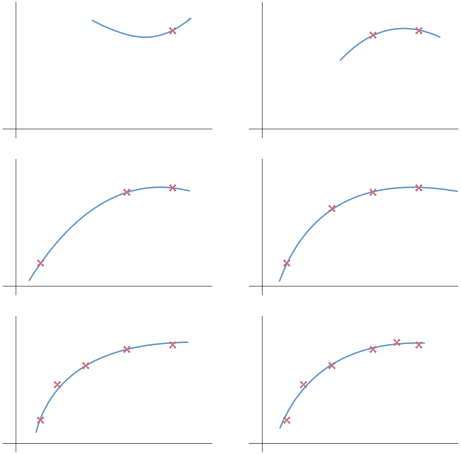
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x_{train}^{(i)};\boldsymbol{\theta}) - y_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x_{cv}^{(i)};\boldsymbol{\theta}) - y_{cv}^{(i)}\right)^2$$



$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$$

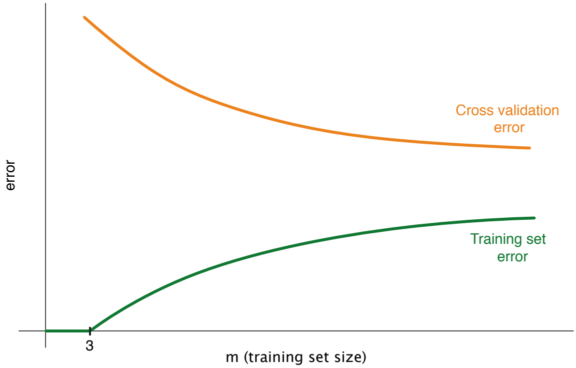


Bias/variance as a function of training set size *m*

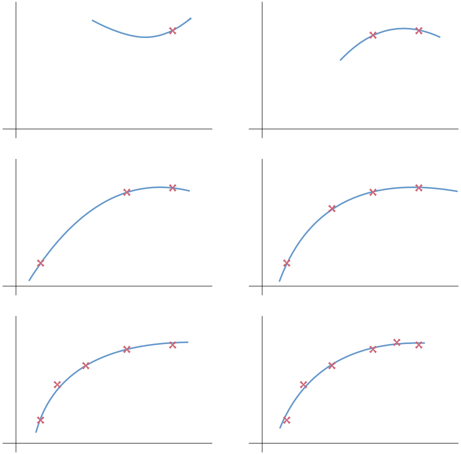
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)};\boldsymbol{\theta}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)}_{train};\boldsymbol{\theta}) - y^{(i)}_{train}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h(x^{(i)}_{cv};\boldsymbol{\theta}) - y^{(i)}_{cv}\right)^2$$



$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$$

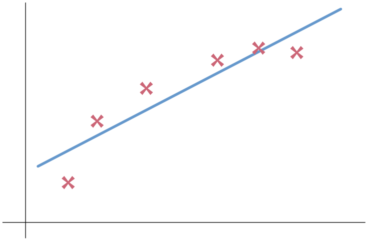


High Bias

$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$$

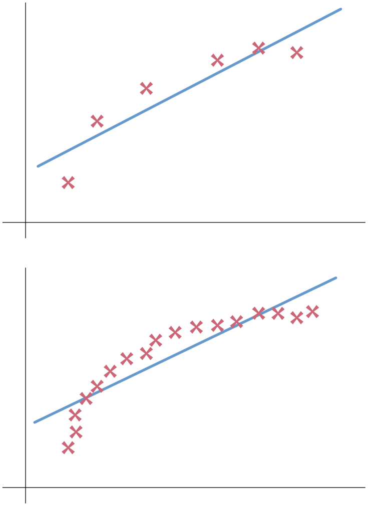
High Bias

$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$$

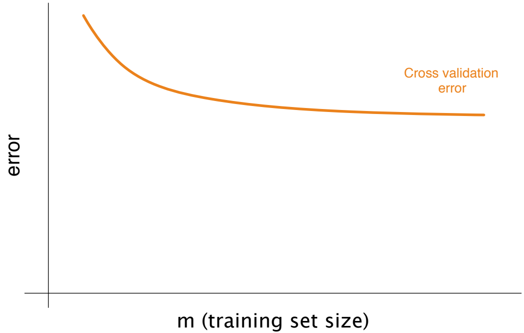


High Bias

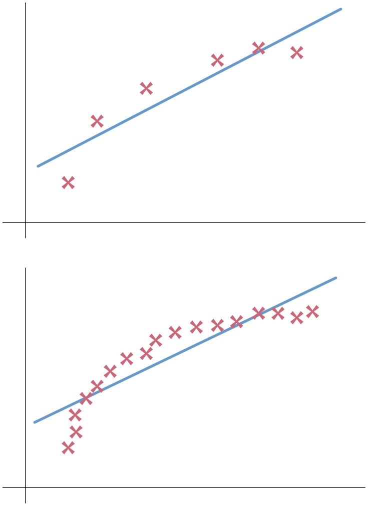
$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x$$



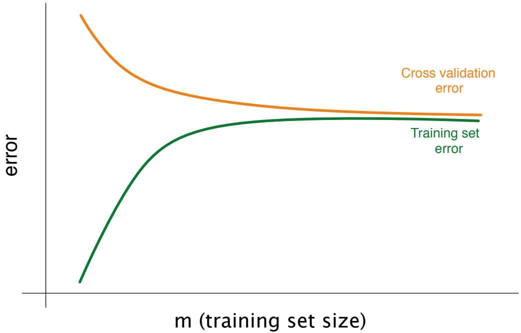
High Bias



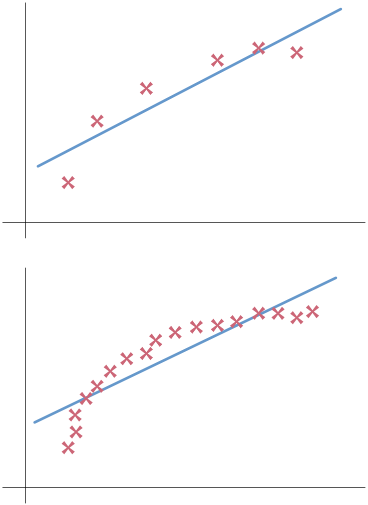
$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$$



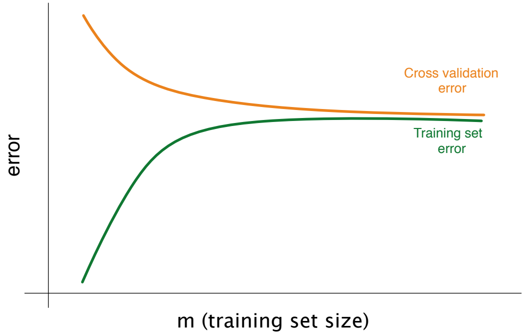
High Bias



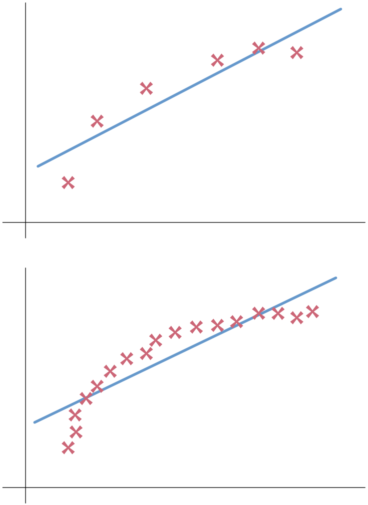
$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$$



High Bias



$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$$



If your learning algorithm suffers from **high bias**, getting more training data alone will **not** help reduce cv/test error.

High Variance

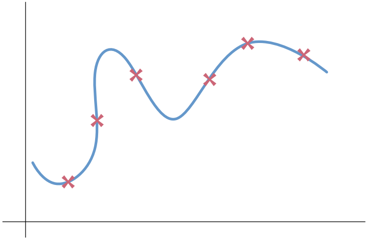
$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{10}$$

and small λ

High Variance

$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{10}$$

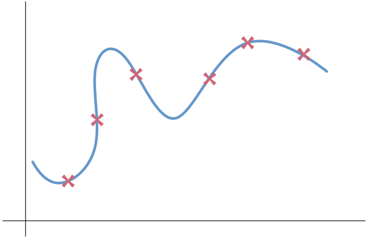
and small λ

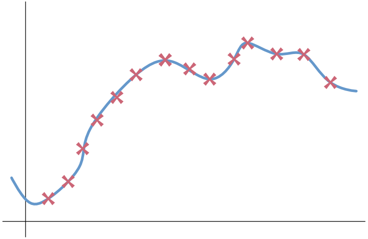


High Variance

$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{10}$$

and small λ

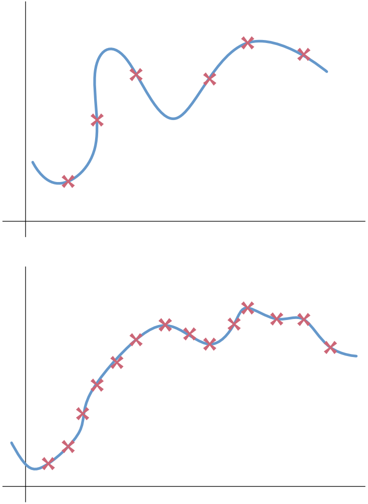




High Variance

$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{10}$$

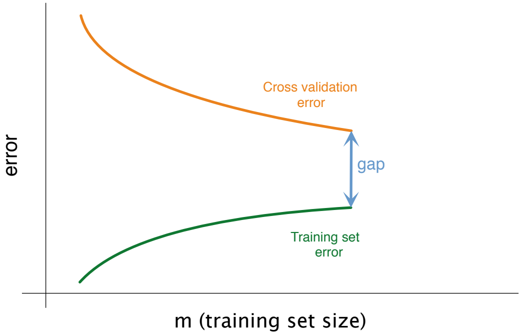
and small λ

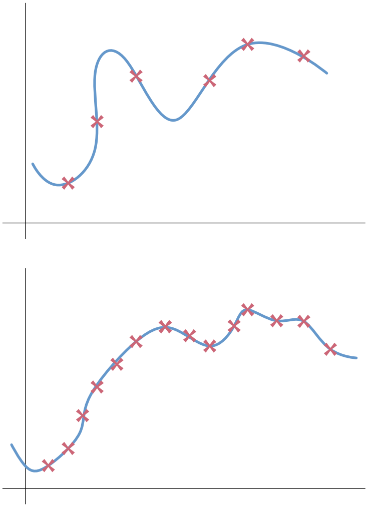


High Variance

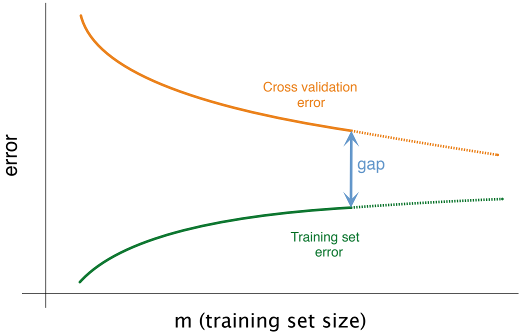
$$h(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{10}$$

and small λ



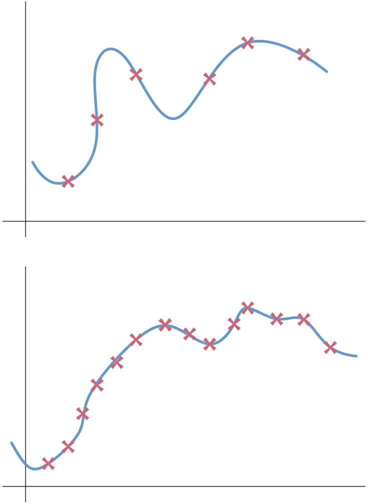


High Variance



$$h(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{10}$$

and small λ



If your learning algorithm suffers from **high variance**, getting more training data **is likely** to help reduce cv/test error.

Learning Curve Diagnostics

The previous examples of high bias error plots and high variance error plots are highly idealized in the sense that the curves were smooth and monotone.

Learning Curve Diagnostics

The previous examples of high bias error plots and high variance error plots are highly idealized in the sense that the curves were smooth and monotone.

In practice, plots of J_{train} and J_{cv} will generally be messier.

Learning Curve Diagnostics

The previous examples of high bias error plots and high variance error plots are highly idealized in the sense that the curves were smooth and monotone.

In practice, plots of J_{train} and J_{cv} will generally be messier.

Nevertheless, you should still be able to see the tell-tale signs of underfitting and overfitting highlighted here.

Putting the Pieces Together

Debugging

Suppose you have implemented **regularized linear regression** to predict housing prices.

Debugging

Suppose you have implemented **regularized linear regression** to predict housing prices.

However, suppose when you test your hypothesis on an unsold house you find that it makes unacceptably large errors in its prediction.

What should you do?

Debugging

Large errors in out-of-sample prediction.

What should you do?

Get more training examples

Reduce the number of features

Increase the number of features

Add polynomial features

Increase λ

Decrease λ