Cognitive Algorithms - Exercise 3

Daniel Wujecki

May 25, 2020

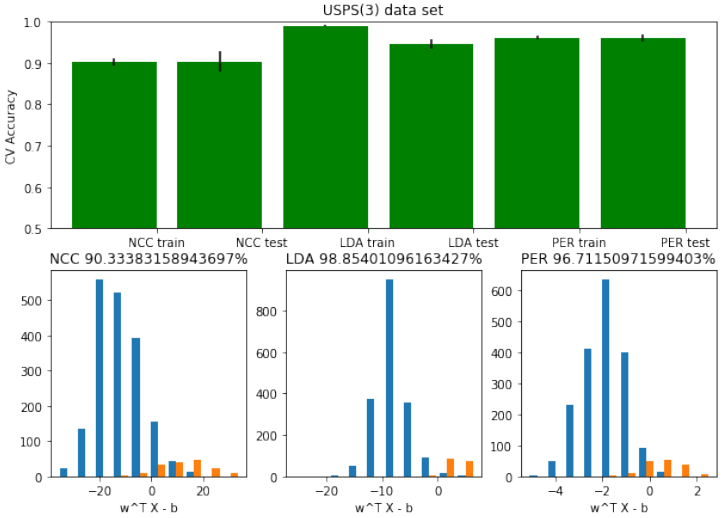
Organizational Remarks

- many mistakes in Quiz 2 on our side
- we will double check the following quizzes

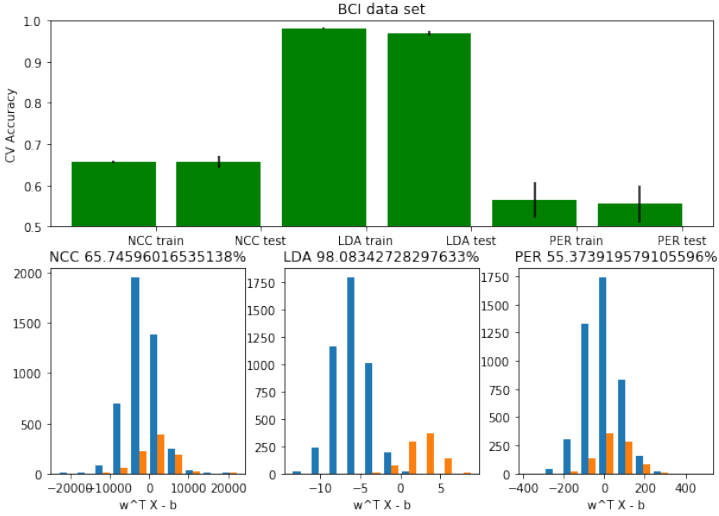
- please send mails only from your TU address

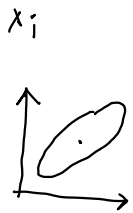
1. Repetition - Assignment 2
2. Linear Regression using a practical Example
3. Task 1 - Ordinary Least Squares (OLS)
4. Task 2 - Variance of OLS Estimation
5. Task 3 - Bias-Variance Tradeoff
6. Task 4 - Invariance under Transformations
7. Task 5 - Cross Validation

NCC vs. LDA vs. Perceptron - USPS

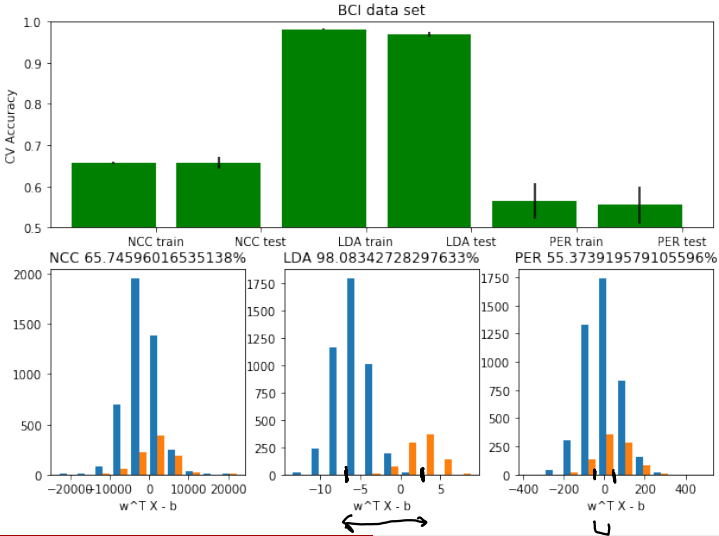


NCC vs. LDA vs. Perceptron - BCI





NCC vs. LDA vs. Perceptron - BCI



- LDA outperforms Perceptron and NCC on data with strong correlation
- Histogram shows LDAs maximization of between class difference

Generalization - Which algorithm generalizes best?

Accuracies:

1. 95 % Training accuracy, 43 % Test accuracy
2. 84 % Training accuracy, 72 % Test accuracy
3. 69 % Training accuracy, 68 % Test accuracy

Generalization - Which algorithm generalizes best?

Accuracies:

1. 95 % Training accuracy, 43 % Test accuracy
2. 84 % Training accuracy, 72 % Test accuracy
3. 69 % Training accuracy, 68 % Test accuracy

- model with highest test accuracy generalizes best
- generalization only depends on the **test accuracy**

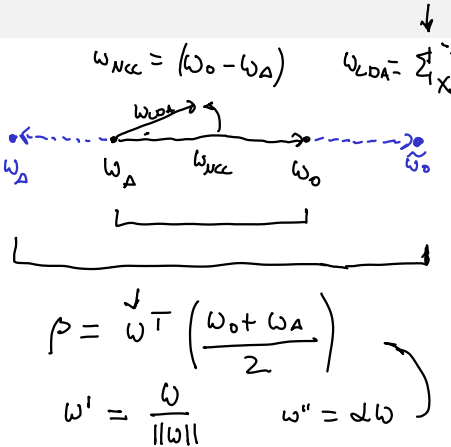
LDA with shifted means

1. Gaussian
2. $\Sigma_\Delta \approx \Sigma_o$
3. We know the true Σ

- data set $X \in \mathbb{R}^{d \times n}$
- optimal classifier (LDA) $w^T x - \beta \geq 0$
- Shifted means for $\alpha \geq 0$:

$$\tilde{w}_o = w_o + \alpha(w_o - w_\Delta)$$
$$\tilde{w}_\Delta = w_\Delta - \alpha(w_o - w_\Delta).$$

- classwise covariance remains the same



LDA with shifted means

- data set $X \in \mathbb{R}^{d \times n}$
- optimal classifier (LDA) $\mathrm{w}^T \mathrm{x} - \beta \geq 0$
- Shifted means for $\alpha \geq 0$:

$$\tilde{\mathrm{w}}_o = \mathrm{w}_o + \alpha(\mathrm{w}_o - \mathrm{w}_\Delta)$$
$$\tilde{\mathrm{w}}_\Delta = \mathrm{w}_\Delta - \alpha(\mathrm{w}_o - \mathrm{w}_\Delta).$$

- classwise covariance remains the same

Which statements are correct?

- ✗ The direction of the weight vector w changes, such that it should be recalculated
- ✓ The direction of the weight vector w remains the same

LDA with shifted means

$$\tilde{w}_{LDA} = \tilde{\Sigma}_1^{-1}(\tilde{w}_o - \tilde{w}_\Delta)$$
$$\simeq (1+2\alpha)\Sigma_1^{-1}(w_o - w_\Delta)$$

- data set $X \in \mathbb{R}^{d \times n}$
- optimal classifier (LDA) $w^T x - \beta \geq 0$
- Shifted means for $\alpha \geq 0$:

$$\tilde{w}_o = w_o + \alpha(w_o - w_\Delta)$$
$$\tilde{w}_\Delta = w_\Delta - \alpha(w_o - w_\Delta).$$

- classwise covariance remains the same

Which statements are correct?

- The direction of the weight vector w changes, such that it should be recalculated
- The direction of the weight vector w remains the same
- ✗ The weight vector w **has** to be scaled with $(1 + 2\alpha)$ (then also β has to be adapted)
- ✓ The weight vector w **can** be scaled with $(1 + 2\alpha)$ (then also β has to be adapted).

LDA with shifted means

- data set $X \in \mathbb{R}^{d \times n}$
- optimal classifier (LDA) $\mathrm{w}^T \mathrm{x} - \beta \geq 0$
- Shifted means for $\alpha \geq 0$:

$$\tilde{\mathrm{w}}_o = \mathrm{w}_o + \alpha(\mathrm{w}_o - \mathrm{w}_\Delta)$$

$$\tilde{\mathrm{w}}_\Delta = \mathrm{w}_\Delta - \alpha(\mathrm{w}_o - \mathrm{w}_\Delta).$$

- classwise covariance remains the same

Which statements are correct?

- The direction of the weight vector w changes, such that it should be recalculated
- The direction of the weight vector w remains the same
- The weight vector w **has** to be scaled with $(1 + 2\alpha)$ (then also β has to be adapted)
- The weight vector w **can** be scaled with $(1 + 2\alpha)$ (then also β has to be adapted).
- The decision boundary is invariant to the shifted means, that is, it is still optimal

Linear Regression using a practical Example

$\Rightarrow$ see jupyter notebook

Linear Regression using a practical Example

$\Rightarrow$ see jupyter notebook

Binary Classification

$$f : \mathbb{R}^d \to \{0, 1\}$$

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} - \beta \geq 0$$

Linear Regression

$$f : \mathbb{R}^d \to \mathbb{R}$$

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} - \beta$$

From the data to the linear function

- Given some data $\mathrm{x}_1, \ldots, \mathrm{x}_n \in \mathbb{R}^d$ with respective class labels $y_1, \ldots, y_n \in \mathbb{R}$
- We assume a linear relationship between x_i and corresponding label y_i, that is

$$x_{i,1}w_1 + x_{i,2}w_2 + \ldots + x_{i,d}w_d - \beta = y_i$$

$$\mathrm{w}^\top \mathrm{x}_i - \beta = y_i$$

From the data to the linear function

- Given some data $\mathrm{x}_1, \ldots, \mathrm{x}_n \in \mathbb{R}^d$ with respective class labels $y_1, \ldots, y_n \in \mathbb{R}$
- We assume a linear relationship between x_i and corresponding label y_i, that is

$$x_{i,1}w_1 + x_{i,2}w_2 + \ldots + x_{i,d}w_d - \beta = y_i$$
$$\mathrm{w}^\top \mathrm{x}_i - \beta = y_i$$

- this gives us a set of n linear equations

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} & 1 \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \\ \beta \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

From the data to the linear function - 1D case

- data $x_1, \ldots, x_n \in \mathbb{R}^1$ with respective class labels $y_1, \ldots, y_n \in \mathbb{R}$

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ \beta \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

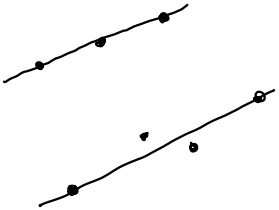
$$X^\top \mathrm{w} = \mathrm{y}^\top$$

From the data to the linear function - 1D case

- data $x_1, \ldots, x_n \in \mathbb{R}^1$ with respective class labels $y_1, \ldots, y_n \in \mathbb{R}$

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ \beta \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

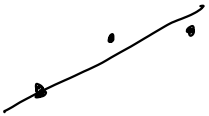
$$X^\top \mathrm{w} = \mathrm{y}^\top$$



- How do we solve that LSE? Gaussian elimination?

From the data to the linear function - 1D case

- data $x_1, \ldots, x_n \in \mathbb{R}^1$ with respective class labels $y_1, \ldots, y_n \in \mathbb{R}$

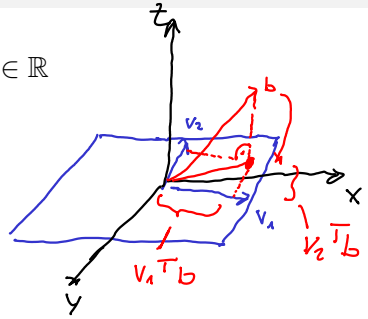


$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ \beta \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

3×2

$$X^\top w = y^\top$$

$$X X^\top w = X y^\top$$



- How do we solve that LSE? Gaussian elimination?
- Assume that the data is noisy and not perfectly fit a linear function
- We have to solve a overdetermined LSE

From the data to the linear function - 1D case

- Solution: Project y into the column space spanned by X^T, that is, $X\mathrm{y}$
- Applying X to both sides yields

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ \beta \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$\Leftrightarrow \quad XX^\top \mathrm{w} = X\mathrm{y}^\top$$

$$\mathrm{w} = (XX^\top)^{-1} X\mathrm{y}^\top$$

From the data to the linear function - 1D case

- Solution: Project y into the column space spanned by X^T, that is, $X\mathrm{y}$
- Applying X to both sides yields

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ \beta \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$\Leftrightarrow \quad XX^\top \mathrm{w} = X\mathrm{y}^\top$$

$$\mathrm{w} = (XX^\top)^{-1} X\mathrm{y}^\top$$

- This is exactly the same solutions as in the lecture
- Note that β is included in w

$\omega^T X = y$

Linear Regression - The Solution

- data $x_1, \ldots, x_n \in \mathbb{R}^d$ with respective class labels $y_1, \ldots, y_n \in \mathbb{R}$
- the function $f(\mathrm{x}) = \mathrm{w}^\top \mathrm{x}$ is optimized for

$$\rightsquigarrow \mathrm{w} = (XX^\top)^{-1} X\mathrm{y}^\top$$

Linear Regression - The Solution

- data $x_1, \ldots, x_n \in \mathbb{R}^1$ with respective class labels $y_1, \ldots, y_n \in \mathbb{R}$
- the function $f(\mathrm{x}) = \mathrm{w}^\top \mathrm{x}$ is optimized for

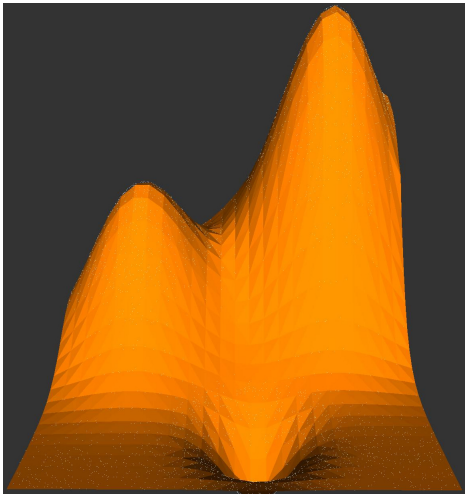
$$\mathrm{w} = (XX^\top)^{-1} X \mathrm{y}^\top$$

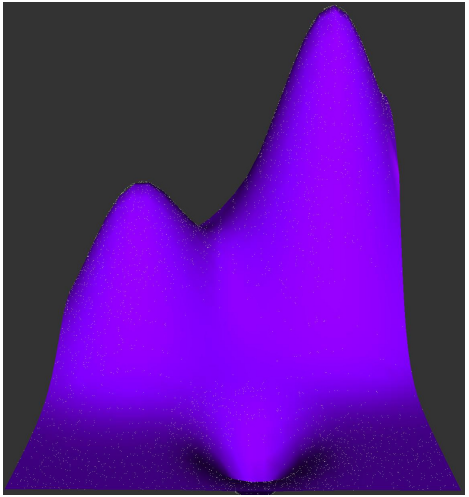
- in the sense of ordinary least squares, that is

$$\mathcal{E}_{lsq}(\mathrm{w}) = \sum_{i=1}^{N} (y_i - \mathrm{w}^\top \mathrm{x}_i)^2 = ||\mathrm{y} - \mathrm{w}^\top X||^2$$

- *Optimizing* $\mathcal{E}_{lsq}(\mathrm{w})$ *by setting* $\frac{\partial \mathcal{E}_{lsq}(\mathrm{w})}{\partial \mathrm{w}} = 0$ *leads to the same result.*

Ordinary Least Squares - Example from another Domain





Polynomial Regression

- Sometimes assumption of linearity is to simple
- a polynomial might be a better fit
- we generalize to a higher degree

$$h(x) = w_1 + w_2 x + w_3 x^2 + \ldots + w_d x^{d-1}$$
$$= \mathbf{w}^\top \phi(x)$$

$$\phi(x) : \mathbb{R} \ni x \mapsto \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{d-1} \end{bmatrix} \in \mathbb{R}^d$$

Polynomial Regression - LSE

- This leads again to a LSE, that can be solved

$$\Phi = [\phi(x_1), \phi(x_2), \ldots, \phi(x_n)] = \begin{bmatrix} 1 & 1 & \cdots \\ x_1 & x_2 & \\ x_1^2 & x_2^2 & \end{bmatrix}$$

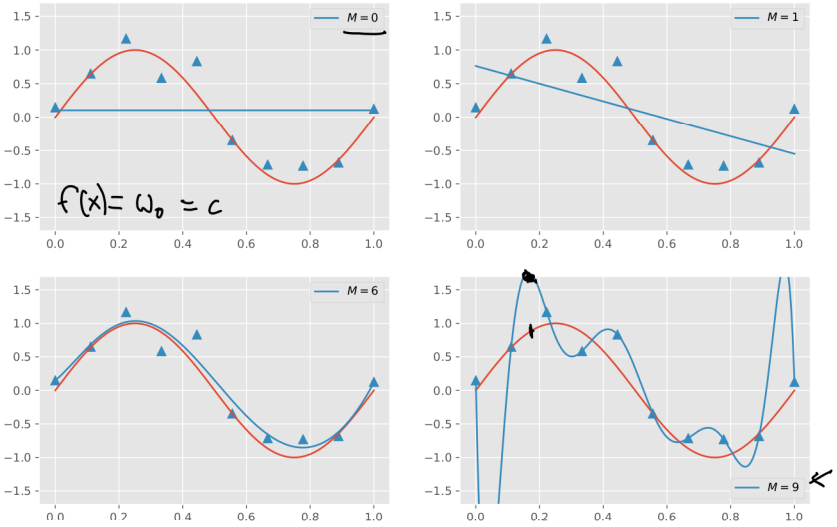
$$\Phi^\top w = y^\top$$

$$\Phi\Phi^\top w = \Phi y^\top$$

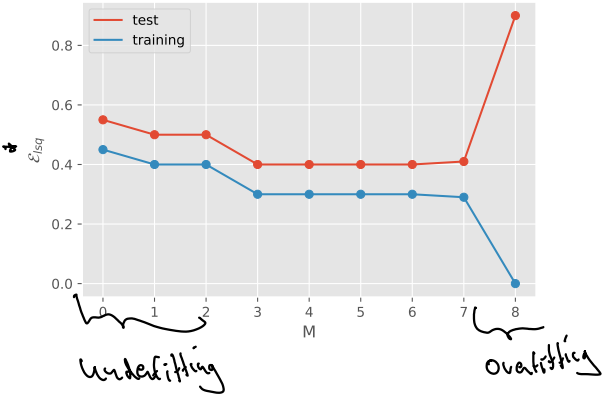
$$w = (\Phi\Phi^\top)^{-1}\Phi y^\top$$

- Other choices of basis expansion ϕ yield e.g. spline regression

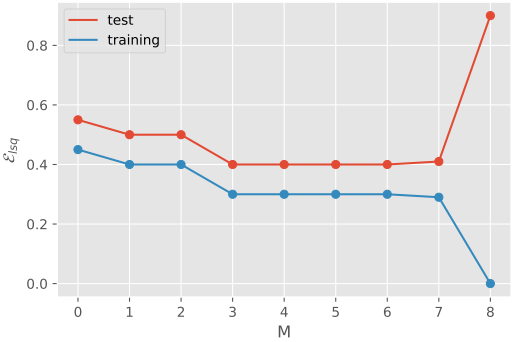
Polynomial Regression - Fitting Sinus Curve



Polynomial Regression - Overfitting for high degrees



Polynomial Regression - Overfitting for high degrees

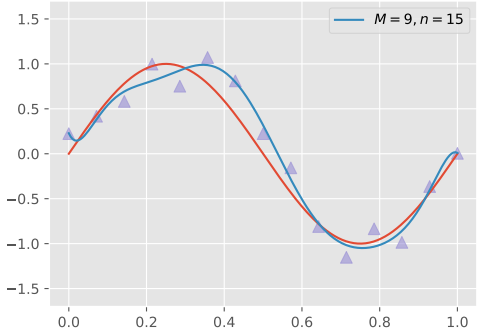


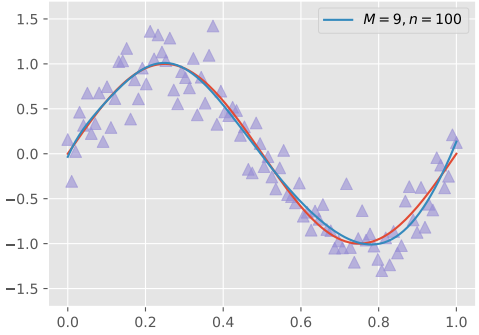
	$M=0$	$M=1$	$M=6$	$M=9$
w_0	0.11	-1.31	33.24	-102297.34
w_1		0.76	-135.49	470852.74
w_2			207.87	-911589.91
w_3			-129.08	963843.87
w_4			19.42	-604186.44
w_5			4.01	227748.07
w_6			0.15	-49782.98
w_7				5656.36
w_8				-244.40
w_9				0.15

$\lambda \|\vec{w}\|$

Polynomial Regression - Increasing number of samples

- Increasing the size of the data set reduces the overfitting problem
- But usually the number of samples is limited





Ridge Regression

- Regression with panelization: restrict large values for w
- Often it is important to control the complexity of the solution w

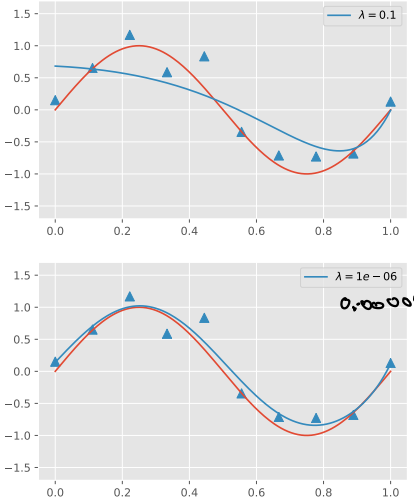
$$\mathcal{E}_{RR}(\mathrm{w}) = \underbrace{||y - \mathrm{w}^\top X||^2}_{\text{OLS}} + \underbrace{\lambda||w||^2}_{\text{RR}}$$

$$\frac{\partial \mathcal{E}_{RR}}{\partial w} = 0$$

- Solution is given by

$$\mathrm{w} = \underbrace{(XX^\top + \lambda I)^{-1} X\mathrm{y}^\top}_{\text{Hyperparameter}}$$

Ridge Regression - Fitting Sinus Curve



- $M = 9$

	$\lambda = 0.1$	$\lambda = 10^{-6}$	$\lambda = 0$
w_0	0.83	28.58	-102297.34
w_1	0.69	-34.35	470852.74
w_2	0.51	-26.34	-911589.91
w_3	0.26	13.48	963843.87
w_4	-0.07	36.93	-604186.44
w_5	-0.48	12.88	227748.07
w_6	-0.94	-36.35	-49782.98
w_7	-1.21	-0.41	5656.36
w_8	-0.27	5.55	-244.40
w_9	0.68	0.14	0.15

Task 1.1 - Ordinary Least Squares

- three data points, $x_1 = 0, x_2 = 1, x_3 = 2$ with respective labels $y_1 = 0, y_2 = 1, y_3 = 0$
- fit a simple linear model $f(x) = w \cdot x$ using OLS
- Recall the 1D OLS can be calculated as

$$w = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - f(x_i))^2 = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i x_i}$$

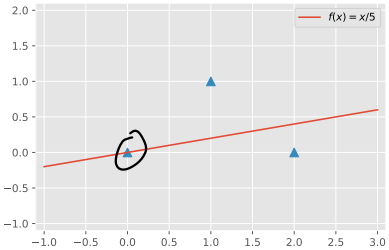
$$w = \frac{0 \cdot 0 + 1 \cdot 1 + 2 \cdot 0}{0 \cdot 0 + 1 \cdot 1 + 2 \cdot 2} = \frac{1}{5} \qquad f(x) = \frac{x}{5} \qquad w = (XX^T)^{-1} X y^T$$

$$w = \underbrace{\left(\begin{bmatrix} 0, 1, 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \right)^{-1}}_{\sum x_i x_i} \underbrace{\begin{bmatrix} 0, 1, 2 \end{bmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}}_{\sum x_i y_i} = 5^{-1} \cdot 1 = \frac{1}{5}$$

Task 1.1 - Ordinary Least Squares

- three data points, $x_1 = 0, x_2 = 1, x_3 = 2$ with respective labels $y_1 = 0, y_2 = 1, y_3 = 0$
- fit a simple linear model $f(x) = w \cdot x$ using OLS
- Recall the 1D OLS can be calculated as

$$w = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - f(x_i))^2 = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i x_i}$$



$\phi(x) = \begin{pmatrix} 1 \\ x \end{pmatrix}$

$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} = X$

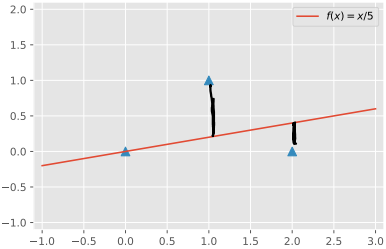
$w = (XX^T)^{-1} X y^T = \begin{pmatrix} 1 \\ 3 \\ 0 \end{pmatrix}$

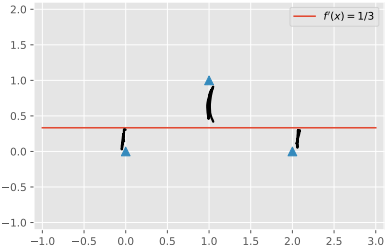
$f(x) = \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{pmatrix} 1/3 \\ 0 \end{pmatrix} = \frac{1}{3}$

Task 1.1 - Ordinary Least Squares

- three data points, $x_1 = 0, x_2 = 1, x_3 = 2$ with respective labels $y_1 = 0, y_2 = 1, y_3 = 0$
- fit a simple linear model $f(x) = w \cdot x$ using OLS
- Recall the 1D OLS can be calculated as

$$w = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - f(x_i))^2 = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i x_i}$$





Task 1.2 - Ordinary Least Squares

- fit a polynomial model $g(x) = w_1 \cdot x + w_2 \cdot x^2 = \mathrm{w}^\top \cdot \phi(x)$
- mapping ϕ is defined as $\phi : \mathbb{R} \ni x \mapsto \begin{bmatrix} x \\ x^2 \end{bmatrix} \in \mathbb{R}^2$
- Recall the OLS solution is obtained as

$$\mathrm{w} = \underset{\mathrm{w}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - g(x_i))^2 = \underbrace{(XX^\top)^{-1} X y^\top}$$

- *the inverse of a* 2×2 *matrix can be calculated by the following formular*

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}^{-1} = \frac{1}{ac - b^2} \begin{bmatrix} c & -b \\ -b & a \end{bmatrix}$$

Task 1.2 - Ordinary Least Squares

- fit a polynomial model $g(x) = w_1 \cdot x + w_2 \cdot x^2 = \mathrm{w}^\top \cdot \phi(x)$
- mapping ϕ is defined as $\phi : \mathbb{R} \ni x \mapsto \begin{bmatrix} x \\ x^2 \end{bmatrix} \in \mathbb{R}^2$ $\quad X = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 1 & 4 \end{bmatrix}$
- Recall the OLS solution is obtained as $\mathrm{w} = (XX^\top)^{-1} X y^\top$

$$\left(\begin{bmatrix} 0 & 1 & 2 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 4 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 & 1 & 2 \\ 0 & 1 & 4 \end{bmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{bmatrix} 5 & 9 \\ 9 & 17 \end{bmatrix}^{-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

$$f(x) = \mathrm{w}^\top \phi(x) = \begin{pmatrix} 2 \\ -1 \end{pmatrix}^\top \begin{pmatrix} x \\ x^2 \end{pmatrix} = 2x - x^2 = -x^2 + 2x - 1 + 1$$

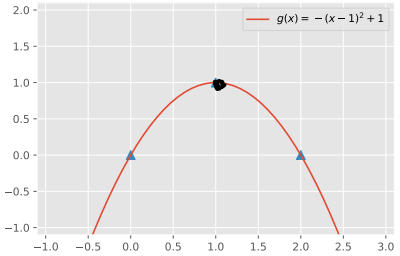
$$= -(x^2 - 2x + 1) + 1$$

$$= -(x-1)^2 + 1$$

$x_2 = 1 \quad y_2 = 1$

Task 1.2 - Ordinary Least Squares

- fit a polynomial model $g(x) = w_1 \cdot x + w_2 \cdot x^2 = \mathrm{w}^\top \cdot \phi(x)$
- mapping ϕ is defined as $\phi : \mathbb{R} \ni x \mapsto \begin{bmatrix} x \\ x^2 \end{bmatrix} \in \mathbb{R}^2$
- Recall the OLS solution is obtained as $\mathrm{w} = (XX^\top)^{-1}Xy^\top$

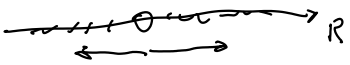


Task 2 - Variance of OLS Estimation

Algorithm 1: Variance of the OLS Estimator

Input: n (number of data points); σ_ϵ^2 (noise variance); σ_x^2 (data variance); w (true slope)

Output: variance of $\hat{w}$



1 Generate n gaussian data points $X = [x_1, \ldots, x_n]$, $x_i \sim \mathcal{N}(0, \sigma_x^2)$

2 **for** $r = 1, \ldots, 10^5$ **do**

3 generate n gaussian noise terms $E = [\epsilon_1, \ldots, \epsilon_n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$

4 compute $y = w \cdot X + E$

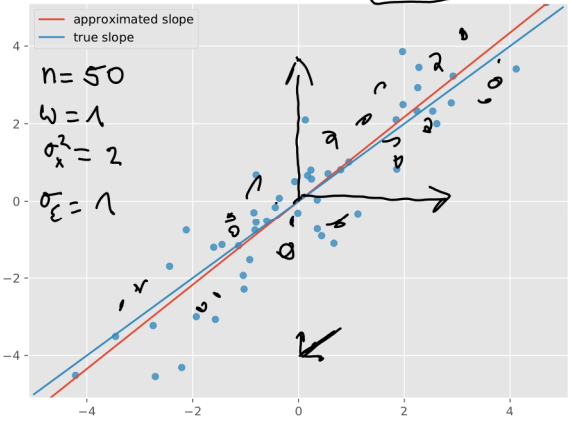
5 compute OLS estimate $\hat{w}[r] = (XX^\top)^{-1} X y^\top$

6 **return** $\mathrm{var}(\hat{w})$

Task 2a) - Variance of OLS Estimation

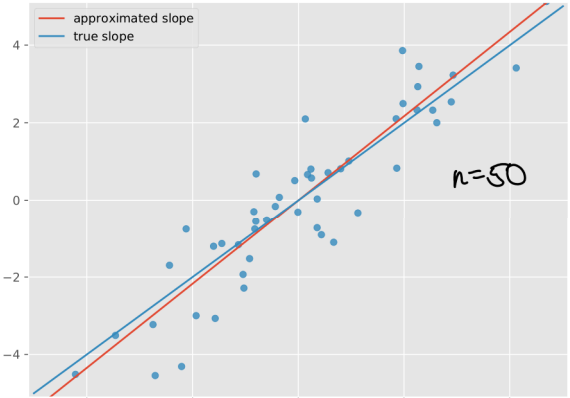
- If the number of data points n increases, the variance of $\hat{w}$ will

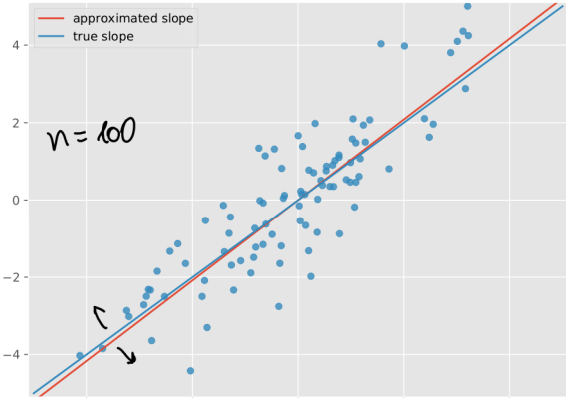
  (a) decrease (b) increase (c) remain the same.



Task 2a) - Variance of OLS Estimation

- If the number of data points n increases, the variance of $\hat{w}$ will
  (a) decrease (b) increase (c) remain the same.

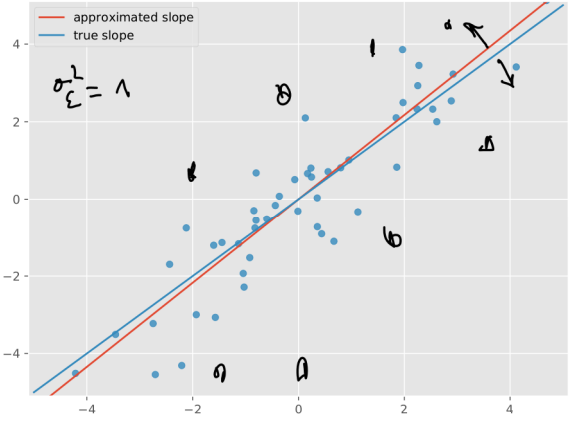




Task 2b) - Variance of OLS Estimation

- If the noise variance σ_ϵ^2 increases, the variance of $\hat{w}$ will

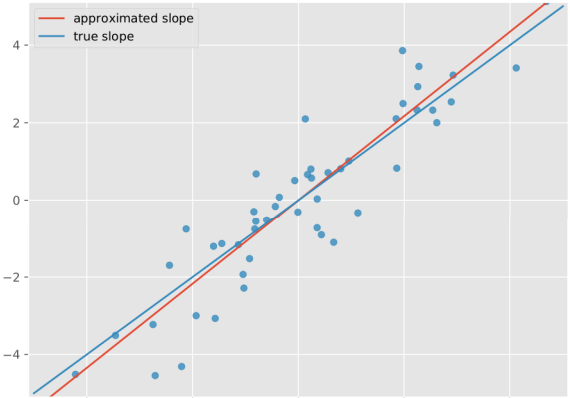
  (a) decrease (b) increase (c) remain the same.

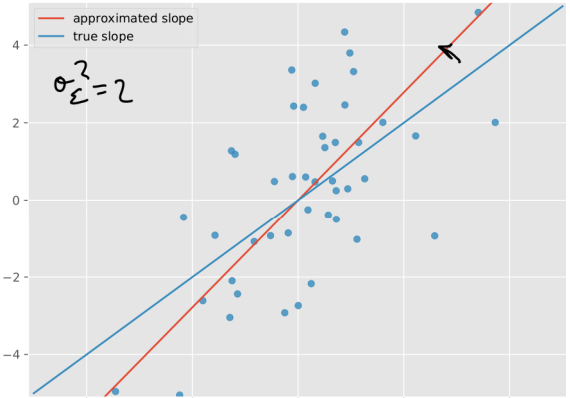


Task 2b) - Variance of OLS Estimation

- If the noise variance σ_ϵ^2 increases, the variance of $\hat{w}$ will

  (a) decrease (b) increase (c) remain the same.

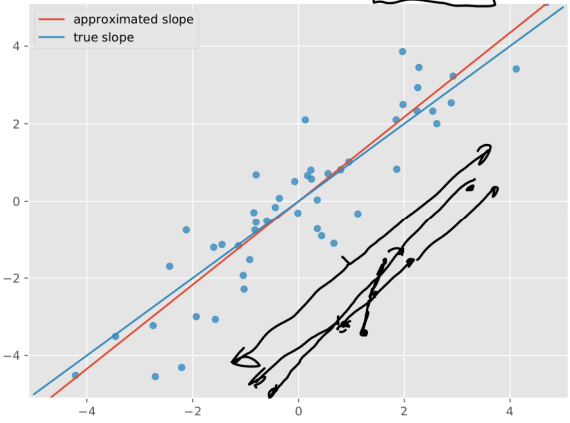




Task 2c) - Variance of OLS Estimation

- If the data variance σ_x^2 increases, the variance of $\hat{w}$ will

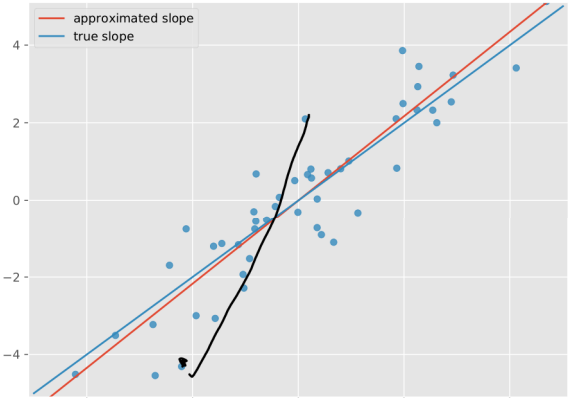
  (a) decrease (b) increase (c) remain the same.

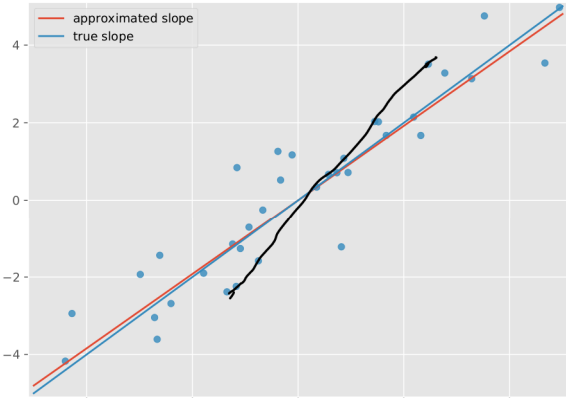


Task 2c) - Variance of OLS Estimation

- If the data variance σ_x^2 increases, the variance of $\hat{w}$ will

  (a) decrease (b) increase (c) remain the same.

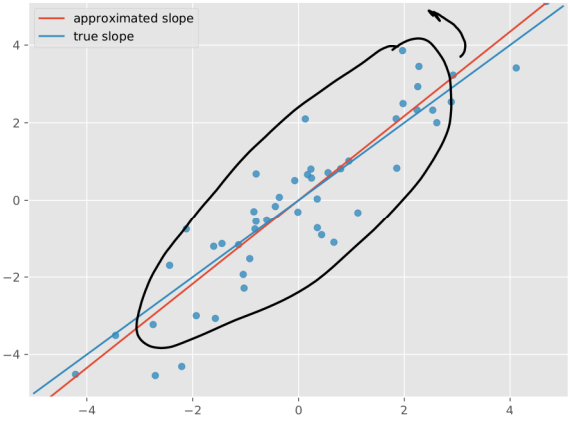




Task 2a) - Variance of OLS Estimation

- If the true slope w increases, the variance of $\hat{w}$ will

  (a) decrease (b) increase (c) remain the same.

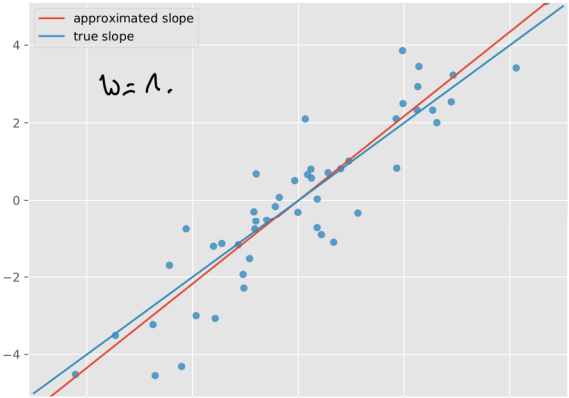


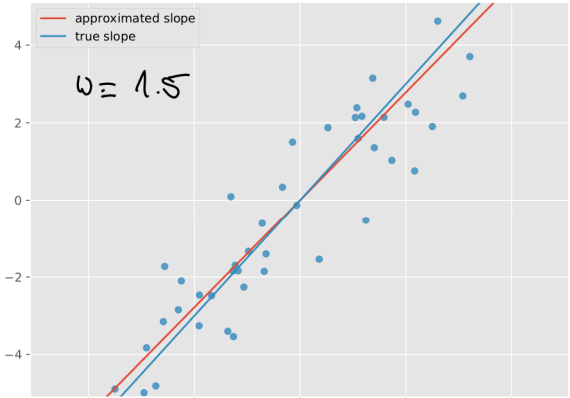
$Y = 0 \cdot X + \varepsilon$

Task 2a) - Variance of OLS Estimation

- If the true slope w increases, the variance of $\hat{w}$ will

  (a) decrease (b) increase (c) remain the same.

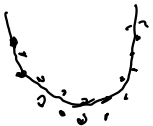




Task 3 - Bias Variance Tradeoff

- true, but unknown, non-linear relationship between one-dimensional input x and one-dimensional output y

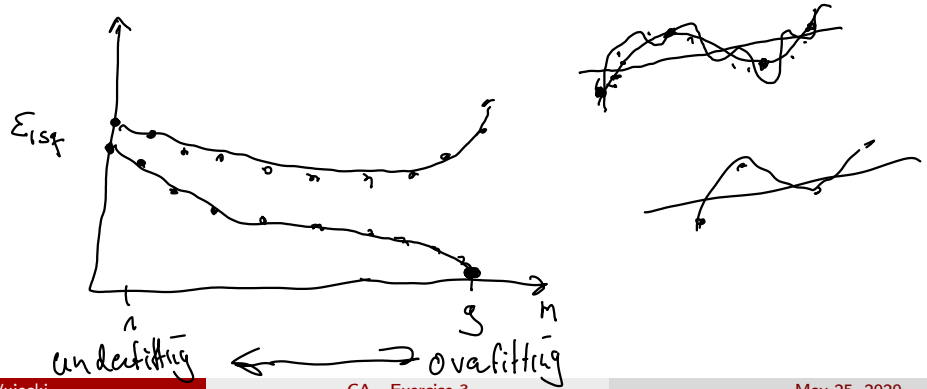
$$y = f(x) + \epsilon$$



- ϵ is uncorrelated noise
- fixed number of training data points n
- relationship is modeled as an m-th order polynomial $\hat{f}$ with ordinary least squares regression

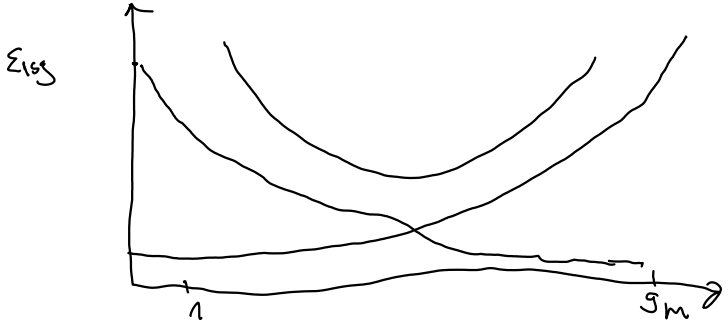
Task 3 - Bias Variance Tradeoff

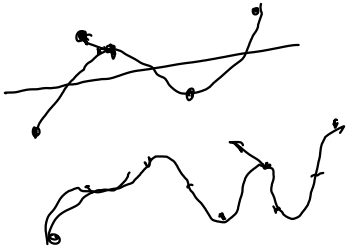
1. Draw a sketch showing two curves: training error vs. the number of features m and test error vs. the number of features m.
2. Annotate the plot with the two terms "Overfitting" and "Underfitting"



Task 3 - Bias Variance Tradeoff

1. Draw a sketch showing two curves: training error vs. the number of features m and test error vs. the number of features m.
2. Annotate the plot with the two terms "Overfitting" and "Underfitting"
3. Draw two more curves: The bias of $\hat{f}$ and the variance of $\hat{f}$ against m





Task 3.4 - Bias Variance Tradeoff

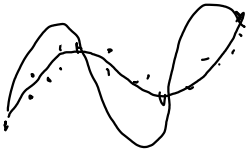
- Suppose we chose m such that we are in the "Overfitting" region
- Use Ridge Regression with a "good" parameter $\lambda > 0$
- Compared to OLS Regression

$\|w\|^2$

1. will the training error decrease, increase or is it ambigious?

Task 3.4 - Bias Variance Tradeoff

- Suppose we chose m such that we are in the "Overfitting" region
- Use Ridge Regression with a "good" parameter $\lambda > 0$
- Compared to OLS Regression



1. will the training error decrease, increase or is it ambigious?
2. will the test error decrease, increase or is it ambigious?

Task 3.4 - Bias Variance Tradeoff

- Suppose we chose m such that we are in the "Overfitting" region
- Use Ridge Regression with a "good" parameter $\lambda > 0$
- Compared to OLS Regression

1. will the training error decrease, increase or is it ambigious?
2. will the test error decrease, increase or is it ambigious?
3. will the bias of $\hat{f}$ decrease, increase or is it ambigious?

Task 3.4 - Bias Variance Tradeoff

- Suppose we chose m such that we are in the "Overfitting" region
- Use Ridge Regression with a "good" parameter $\lambda > 0$
- Compared to OLS Regression

1. will the training error decrease, increase or is it ambigious?
2. will the test error decrease, increase or is it ambigious?
3. will the bias of $\hat{f}$ decrease, increase or is it ambigious?
4. will the variance of $\hat{f}$ decrease, increase or is it ambigious?

$$(A^T)^{-1} = (A^{-1})^T$$

Task 4.1 - Invariance under Transformations

- Data set $X \in \mathbb{R}^{d \times n}$ and $\vec{y} \in \mathbb{R}^{1 \times n}$
- Let $A \in \mathbb{R}^{d \times d}$ be an invertible matrix
- Show that Linear Regression is invariant under transformations $X \mapsto AX$

Task 4.1 - Invariance under Transformations

$$(AX)^T = X^T A^T$$

$$(AX)^{-1} = X^{-1} A^{-1}$$

$$X \rightarrow AX$$

$$\hat{w}_{ols} = (XX^\top)^{-1} X y^\top$$

$$\hat{y}_{ols} = \hat{w}_{ols}^\top X$$

$$w = \left(AX(AX)^T\right)^{-1} AX y^T = \left(AXX^T A^T\right)^{-1} AX y^T$$

$$= (A^T)^{-1} (XX^T)^{-1} \underbrace{A^{-1} A}_{=I} X y^T$$

$$= (A^T)^{-1} \underbrace{(XX^T)^{-1} X y^T}_{w}$$

$$= (A^T)^{-1} w$$

$$\left(A^{T^{-1}}\right)^T = \left(A^{-1}\right)^{T^T} = A^{-1}$$

$$y = \left((A^T)^{-1} w\right)^T AX$$

$$= w^T \underbrace{A^{-1} A}_{I} X$$

$$= w^T X = y$$

Task 4.2 - Invariance under Transformations

- Data set $X \in \mathbb{R}^{d \times n}$ and $y \in \mathbb{R}^{1 \times n}$
- Let $A \in \mathbb{R}^{d \times d}$ be an invertible matrix
- Show that Ridge Regression is invariant under transformations $X \mapsto AX \Leftrightarrow A$ is orthogonal

Task 4.2 - Invariance under Transformations

$$A A^T = I$$

$$A^T = A^{-1}$$

$$\hat{w}_{ridge} = (XX^\top + \lambda I)^{-1} X y^\top$$

$$\hat{y}_{ridge} = \hat{w}_{ridge}^\top X$$

$$w = \left(AX(AX)^T + \lambda I\right)^{-1} AX y^T$$

$$= \left(AXX^T A^T + \lambda AA^T\right)^{-1} AX y^T$$

$$= \left(A\left(XX^T + \lambda I\right)A^T\right)^{-1} AXy^T$$

$$= (A^T)^{-1}\left(XX^T + \lambda I\right)^{-1} A^{-1} AX y^T$$

$$= (A^T)^{-1}\underbrace{\left(XX^T + \lambda I\right)^{-1} Xy^T}_{w} = A^{T^{-1}} w = A^{-1^T} w = Aw$$

$$y = (Aw)^T AX = w^T A^T AX$$

$$= w^T X$$

Cross Validation

- useful for small data sets
- Randomly split the data into train and test folds → not using the full data set

Cross Validation

- useful for small data sets
- Randomly split the data into train and test folds → not using the full data set

  Fold 1 $[x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}, x_{i_5}, x_{i_6}]$ acc = 80%

  $\mathcal{F}_1^{train}$ $\mathcal{F}_1^{test}$

  Fold 2 $[x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}, x_{i_5}, x_{i_6}]$ $[x_1, x_2, x_3, x_4, x_5, x_6]$ acc = 70%

  $\mathcal{F}_1^{test}$ $\mathcal{F}_2^{train}$ test training

  Fold 3 ... $\frac{1}{2}(80 + 70)$
- for each fold: train the data and test the data on the training and test folds

Cross Validation

- useful for small data sets
- Randomly split the data into train and test folds → not using the full data set

  Fold 1 $[\underbrace{x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}}_{\mathcal{F}_1^{train}} \underbrace{x_{i_5}, x_{i_6}}_{\mathcal{F}_1^{test}}]$

  Fold 2 $[\underbrace{x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}}_{\mathcal{F}_2^{test}} \underbrace{x_{i_5}, x_{i_6}}_{\mathcal{F}_2^{train}}]$

  Fold 3 ...
- for each fold: train the data and test the data on the training and test folds
- can be used for either **model selection** or **model evaluation**

Task 5.1 - Cross Validation

You are a reviewer for a international conference on machine learning and you read a paper that selected a small number of features out of a large number of features for a given classification problem. The paper argues as follows:

1. We uses all our available data to select a subset of "'good"' features that had fairly strong correlation with the class labels. => model selection
2. Our final model contained only those features. We evaluate the prediction error of the final model by 10-fold crossvaldiation on all the available data. => model evaluation
3. We obtained a low cross-validation error. Thus, we have achieved high classification accurarcy with only few meaningful features. (This is novel and amazing.)

Would you accept or reject the paper? Why?

Task 5.2 - Cross Validation

Majority Classifier: Given a set of training data, the majority classifier always outputs the class that is in the majority in the training set, regardless of the input.

X, $y = [0, 0, 0, 1, 1]$

↳ $y = 0$

Task 5.2 - Cross Validation

Majority Classifier: Given a set of training data, the majority classifier always outputs the class that is in the majority in the training set, regardless of the input.

Suppose you are testing a new algorithm on a data set consisting of 100 positive and 100 negative examples. You plan to use leave-one-out cross-validation (that is 200-fold cross-validation) and compare your algorithm to a baseline function, a simple majority classifier. You expect the majority classifier to achieve about 50% classification accuracy, but to your surprise, it scores zero every time. Why?