EE2211 Tutorial 7

Question 1:
This question explores the use of Pearson's correlation as a feature selection metric. We are given the following training dataset.

	Datapoint 1	Datapoint 2	Datapoint 3	Datapoint 4	Datapoint 5
Feature 1	0.3510	2.1812	0.2415	-0.1096	0.1544
Feature 2	1.1796	2.1068	1.7753	1.2747	2.0851
Feature 3	-0.9852	1.3766	-1.3244	-0.6316	-0.8320
Target y	0.2758	1.4392	-0.4611	0.6154	1.0006

What are the top two features we should select if we use Pearson's correlation as a feature selection metric? Here's the definition of Pearson's correlation. Given N pairs of datapoints $\{(a_1, b_1), (a_2, b_2), \cdots, (a_N, b_N)\}$, the Pearson's correlation r is defined as $r = \frac{\frac{1}{N}\sum_{n=1}^{N}(a_i-\bar{a})(b_i-\bar{b})}{\sqrt{\frac{1}{N}\sum_{n=1}^{N}(a_i-\bar{a})^2}\sqrt{\frac{1}{N}\sum_{n=1}^{N}(b_i-\bar{b})^2}}$, where $\bar{a} = \frac{1}{N}\sum_{n=1}^{N} a_n$ and $\bar{b} = \frac{1}{N}\sum_{n=1}^{N} b_n$ are the empirical means of a and b respectively. $\sigma_a = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(a_i - \bar{a})^2}$ and $\sigma_b = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(b_i - \bar{b})^2}$ are referred to as the empirical standard deviation of a and b. $Cov(a, b) = \frac{1}{N}\sum_{n=1}^{N}(a_i - \bar{a})(b_i - \bar{b})$ is known as the empirical covariance between a and b Cov(a, y)

$$r = \frac{\text{cov}(a, y)}{\sigma_a \times \sigma_y}$$

Answer:

Mean of Feature 1 = $\mu_1 = \frac{0.3510+2.1812+0.2415-0.1096+0.1544}{5} = 0.5637$

Mean of Feature 2 = $\mu_2 = \frac{1.1796+2.1068+1.7753+1.2747+2.0851}{5} = 1.6843$

Mean of Feature 3 = $\mu_3 = \frac{-0.9852+1.3766-1.3244-0.6316-0.8320}{5} = -0.4793$

Mean of Target y = $\mu_y = \frac{0.2758+1.4392-0.4611+0.6154+1.0006}{5} = 0.5740$

Feature 1 std = $\sigma_1 = \sqrt{\frac{(0.3510-\mu_1)^2+(2.1812-\mu_1)^2+(0.2415-\mu_1)^2+(-0.1096-\mu_1)^2+(0.1544-\mu_1)^2}{5}} = 0.8229$

Feature 2 std = $\sigma_2 = \sqrt{\frac{(1.1796-\mu_2)^2+(2.1068-\mu_2)^2+(1.7753-\mu_2)^2+(1.2747-\mu_2)^2+(2.0851-\mu_2)^2}{5}} = 0.3924$

Feature 3 std = $\sigma_3 = \sqrt{\frac{(-0.9852-\mu_3)^2+(1.3766-\mu_3)^2+(-1.3244-\mu_3)^2+(-0.6316-\mu_3)^2+(-0.8320-\mu_3)^2}{5}} = 0.9552$

Target y std = $\sigma_y = \sqrt{\frac{(0.2758-\mu_y)^2+(1.4392-\mu_y)^2+(-0.4611-\mu_y)^2+(0.6154-\mu_y)^2+(1.0006-\mu_y)^2}{5}} = 0.6469$

Cov(Feature 1, y) = $\frac{1}{5}[(0.3510 - \mu_1)(0.2758 - \mu_y) + (2.1812 - \mu_1)(1.4392 - \mu_y) + (0.2415 - \mu_1)(-0.4611 - \mu_y) + (-0.1096 - \mu_1)(0.6154 - \mu_y) + (0.1544 - \mu_1)(1.0006 - \mu_y)] = 0.3188$

Cov(Feature 2,y) = $\frac{1}{5}[(1.1796 - \mu_2)(0.2758 - \mu_y) + (2.1068 - \mu_2)(1.4392 - \mu_y) + (1.7753 - \mu_2)(-0.4611 - \mu_y) + (1.2747 - \mu_2)(0.6154 - \mu_y) + (2.0851 - \mu_2)(1.0006 - \mu_y)] = 0.1152$

Cov(Feature 3,y) = $\frac{1}{5}\left[(-0.9852-\mu_3)\left(0.2758-\mu_y\right)+(1.3766-\mu_3)\left(1.4392-\mu_y\right)+(-1.3244-\mu_3)\left(-0.4611-\mu_y\right)+(-0.6316-\mu_3)\left(0.6154-\mu_y\right)+(-0.8320-\mu_3)\left(1.0006-\mu_y\right)\right]=0.4949$

Correlation of Feature 1 & y = $\frac{Cov(Feature\ 1,y)}{\sigma_1\sigma_y}=\frac{0.3188}{0.8229\times0.6469}=0.5988$

Correlation of Feature 2 & y = $\frac{Cov(Feature\ 2,y)}{\sigma_2\sigma_y}=\frac{0.1152}{0.3924\times0.6469}=0.4537$

Correlation of Feature 3 & y = $\frac{Cov(Feature\ 3,y)}{\sigma_3\sigma_y}=\frac{0.4949}{0.9552\times0.6469}=0.8009$

Therefore, the top 2 features are Feature 1 and Feature 3.

Question 2:

This question further explores linear regression and ridge regression. The following data pairs are used for training:

$\{x = -10\} \rightarrow \{y = 4.18\}$

$\{x = -8\} \rightarrow \{y = 2.42\}$

$\{x = -3\} \rightarrow \{y = 0.22\}$

$\{x = -1\} \rightarrow \{y = 0.12\}$

$\{x = 2\} \rightarrow \{y = 0.25\}$

$\{x = 7\} \rightarrow \{y = 3.09\}$

The data for testing are as follows:

$\{x = -9\} \rightarrow \{y = 3\}$

$\{x = -7\} \rightarrow \{y = 1.81\}$

$\{x = -5\} \rightarrow \{y = 0.80\}$

$\{x = -4\} \rightarrow \{y = 0.25\}$

$\{x = -2\} \rightarrow \{y = -0.19\}$

$\{x = 1\} \rightarrow \{y = 0.4\}$

$\{x = 4\} \rightarrow \{y = 1.24\}$

$\{x = 5\} \rightarrow \{y = 1.68\}$

$\{x = 6\} \rightarrow \{y = 2.32\}$

$\{x = 9\} \rightarrow \{y = 5.05\}$

(a) Use the polynomial model from orders 1 to 6 to train and test the data without regularization. Plot the Mean Squared Errors (MSE) over orders from 1 to 6 for both the training and the test sets. Which model order provides the best MSE in the training and test sets? Why? [Hint: the underlying data was generated using a quadratic function + noise]

(b) Use regularization (ridge regression) λ=1 for all orders and repeat the same analyses. Compare the plots of (a) and (b). What do you see? [Hint: the underlying data was generated using a quadratic function + noise]

Answer:

Please see code: Tut7_Q2_yeo.py

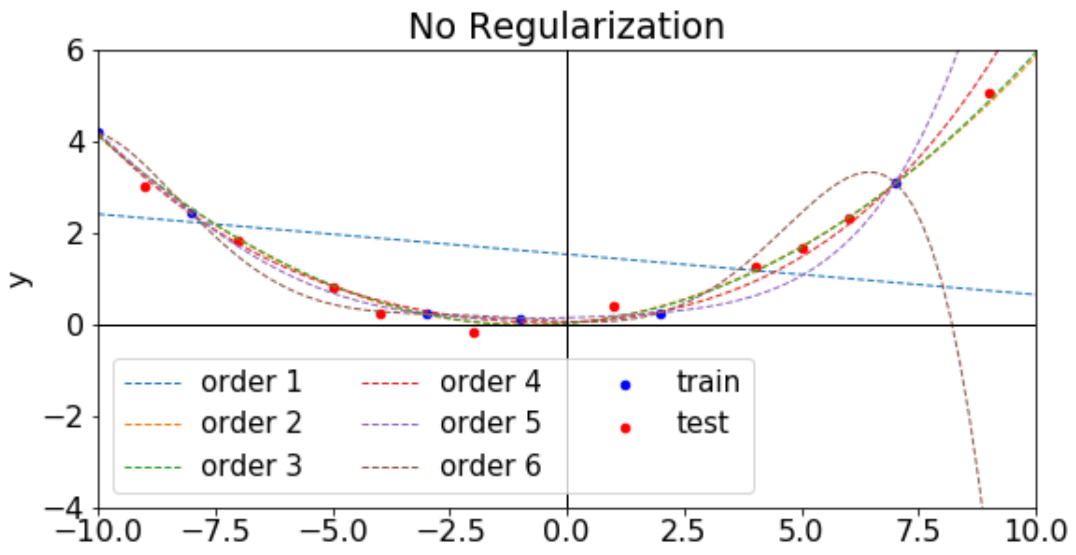
Q1(a)

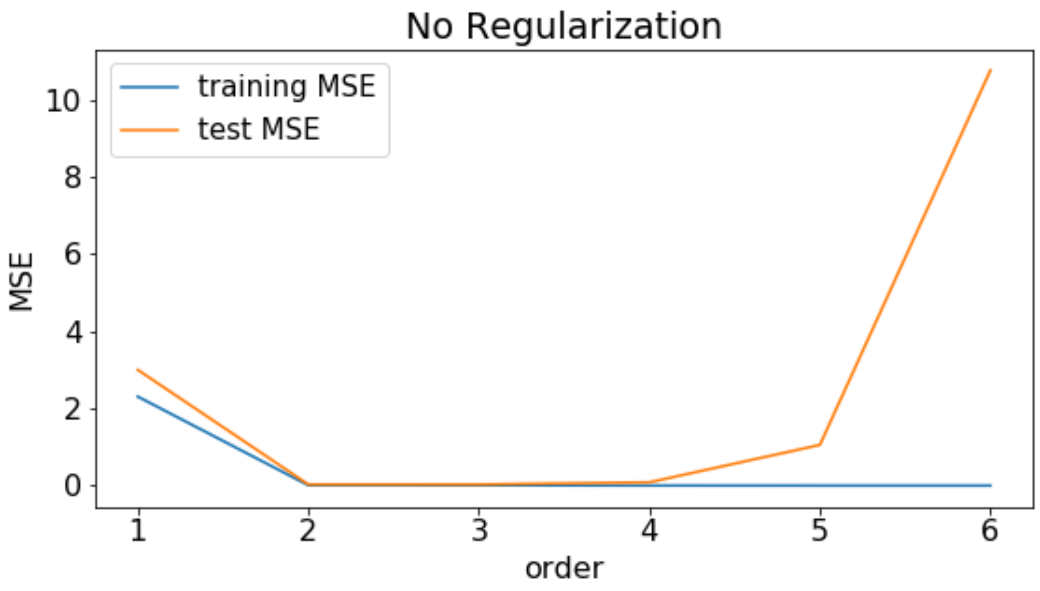
- There are 6 training data points. For polynomial orders 1 to 5, we can use the **Primal** solution: $\hat{w} = (P^T P)^{-1} P^T y$. For order 6, there are 7 unknowns, so the system is under-determined, so we use the **Dual** solution: $\hat{w} = P^T (P P^T)^{-1} y$
- See plots for estimated polynomial curves and MSE below

- ====== No Regularization =======

```
Training MSE:   [2.3071.      8.4408e-03   8.3026e-03   1.7348e-03   3.8606e-25     2.3656e-17]
Test MSE:        [ 3.0006       0.0296         0.0301           0.0854          1.0548           10.7674]
```

- Observe that the estimated polynomial curves for orders 5 and 6 pass through the training samples exactly. This results in training MSE of virtually 0, but high test MSE => overfitting
- Note that even though the true underlying data came from a quadratic model (order = 2), estimated polynomial curves for orders 2, 3 and 4 have relatively low training and test MSE.
- Polynomial curve of order 1 (linear curves) have high training and test MSE => underfitting

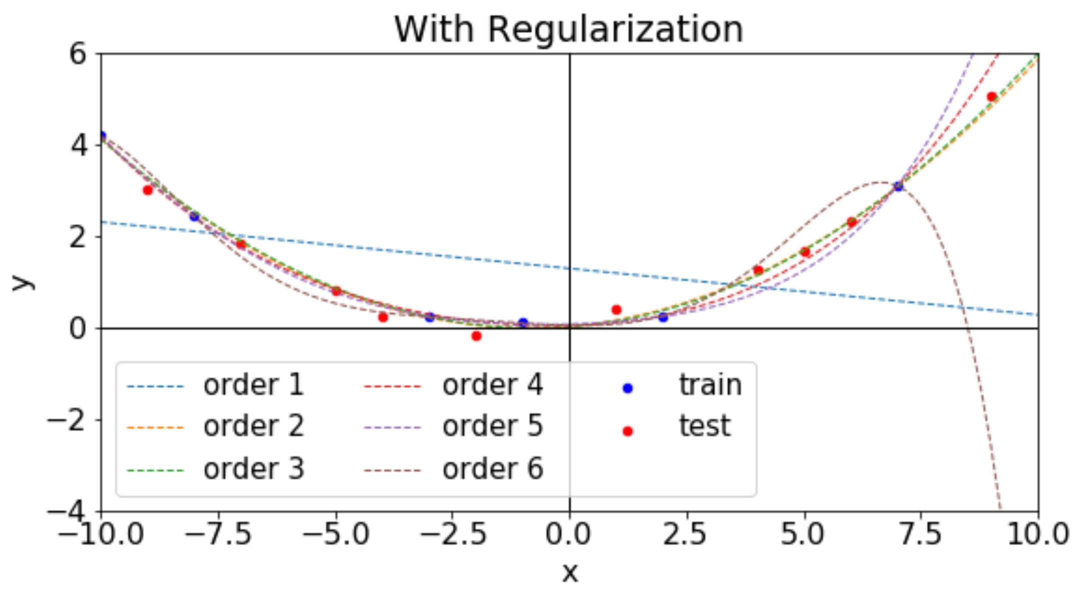


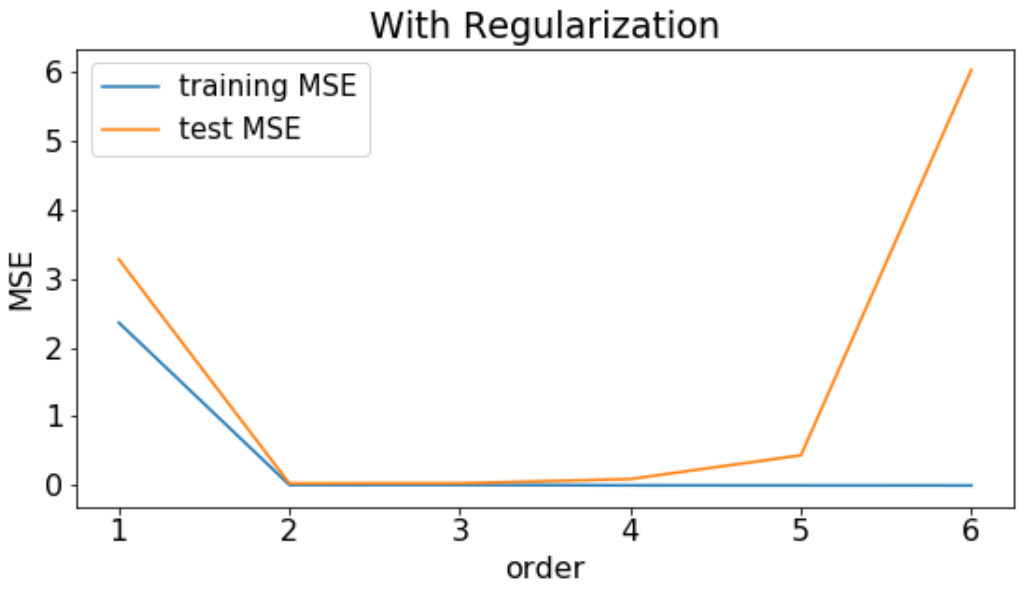


Q1(b)

- With regularization, we can simply use the primal solution even for order 6: $\hat{w} = (P^T P + \lambda I)^{-1} P^T y$.
- See plots for estimated polynomial curves and MSE below

- ====== Regularization =======
```
Training MSE:   [2.3586   8.4565e-03   8.3560e-03   1.8080e-03   7.2650e-04    1.9348e-04]
Test MSE:       [3.2756   0.0302       0.0314       0.0939       0.4369        6.0202]
```
- With the regularization, none of the polynomial curves passes through the training samples exactly. In the case of orders 5 and 6, test MSE dropped from 1.0548 (order 5) and 10.7674 (order 6) to 0.4369 (order 5) and 6.0202 (order 6) after regularization was added. Thus, the regularization reduces the overfitting
- On the other hand, the regularization did not help orders 1 to 4. Observe that the test MSE actually went up. In this case, the regularization was overly strong, which ended up hurting these orders.
- Note that the addition of the regularization does not necessarily favor the lower order (compared with the higher order). For example, the loss for order 2 is 0.018476, but the loss for order 6 is 0.0036308, which is lower. Thus, adding regularization does not help us choose the best polynomial order for best prediction.
- In fact, selecting the best regularization parameter or model complexity (e.g., polynomial order) is typically done through inner-loop (nested) cross-validation or training-validation-test scheme (**which will be taught in a future lecture**).




With Regularization
training MSE
test MSE
MSE
order
0
1
2
3
4
5
6
1
2
3
4
5
6