EE660 Week2 HW Pin-Hsuan Lee 1139093874

1. This problem may be solved using MATLAB or Python; the functions/commands stated below are for MATLAB implementations.

   You are to implement a simple curve fitting problem using 1D regression. In this problem you are to **code the assigned portions of the regressions yourself**; using a package's regression or curve-fit function will not suffice.

   (a) Model the curve to be fit, $\hat{f}(x)$, as a d^{th} order polynomial. Write down the mean- squared error objective function for curve fitting, in terms of x_i, y_i, and w_m, in which i is the data point index ($i = 1,2,\cdots,N$), and m is the weight index ($m = 0,1,\cdots,d$).

$$\hat{f}(x) = w^T x$$

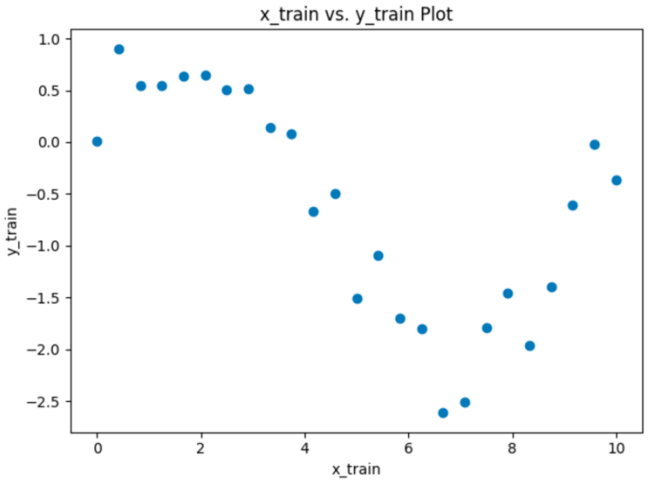
$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - w^T x_i)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \sum_{m=0}^{d} w_m x_i^m\right)^2$$

$$w^T = [w_0, w_1, \cdots, w_d], \quad x_i = \begin{bmatrix} x_i^0 \\ x_i^1 \\ x_i^2 \\ \vdots \\ x_i^d \end{bmatrix}, \quad x_i^0 = 1, \quad i = 1, 2, \cdots, N$$

   (b) Write the objective function in matrix form in terms of $\underline{\underline{\Phi}}, \underline{y}$, and $\underline{w}$. ($\underline{\underline{\Phi}}$ is the basis-set expansion version of $\underline{\underline{X}}$; the i^{th} row is $\underline{\phi}^T(x_i)$.)

$$J(\underline{w}) = \left\| \underline{y} - \underline{\underline{\Phi}}\,\underline{w} \right\|_2^2$$

   (c) Download the provided data from the dropbox and plot only the points of `x_train` *vs.* `y_train` (use the command `scatter(x, y)`).



   (d) Let the hypothesis set be polynomials in x of degree [1, 2, 3, 7, 10]. Find the curve parameters (using only data from `x_train` and `y_train`) for each of these polynomial degrees, using pseudo-inverse. (You can use commands `hold on` and `plot(x, y)` to visualize how well the curve fits to the training data, but this is not mandatory.) Show the computed weight vectors $\underline{w}_1, \underline{w}_2, \underline{w}_3, \underline{w}_7, \underline{w}_{10}$ where $\underline{w}_d$ denotes the weight vector for the d^{th} order polynomial.

   Hint: to set this up as a pseudo-inverse problem, use the basis function expansion of part (b) above.

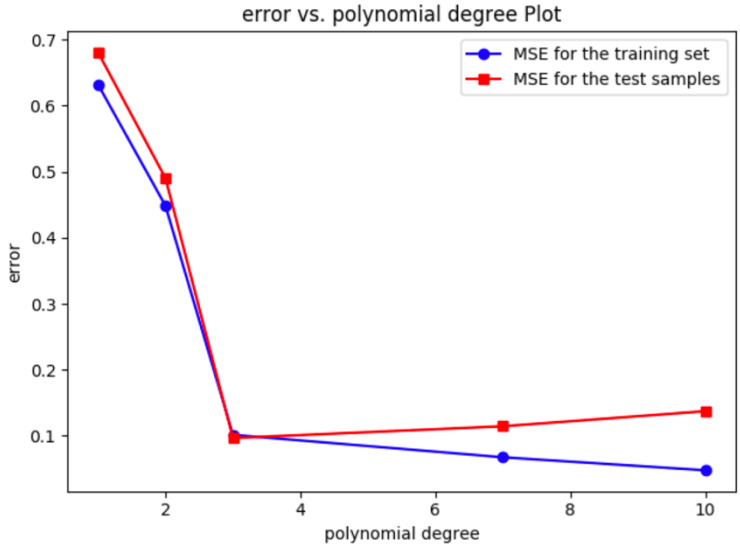
(e) Compute the mean squared error (MSE) on the training set for each one, i.e.,

$$MSE_d = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - \underline{w}_d^T \underline{\phi}\left(\underline{x}_i\right)\right]^2 .$$

Plot error *vs.* polynomial degree. Which polynomial degree seems to be the best model based on the training sample MSE only?

(f) Using the same weights ($\underline{w}_1, \underline{w}_2, \underline{w}_3, \underline{w}_7, \underline{w}_{10}$), compute the MSE for the test samples, *i.e.*, using `x_test` and `y_test`. Plot error *vs.* polynomial degree again. Which polynomial degree seems to be the best model based on the test sample MSE only?

```
Question1.(d)~(f)

When degree= 1
w 1 = [ 0.55787413 -0.23494215]
MSE(train)= 0.6320445845059596
MSE(test)= 0.6807097919920291

When degree= 2
w 2 = [ 1.40766343 -0.76698414  0.0532042 ]
MSE(train)= 0.44836937486829886
MSE(test)= 0.4901401122690916

When degree= 3
w 3 = [ 0.18301369  0.87201349 -0.36501611  0.02788135]
MSE(train)= 0.10158531043884199
MSE(test)= 0.09673216746172301

When degree= 7
w 7 = [ 1.39650229e-01  1.39453740e+00 -1.22891436e+00  5.21807571e-01
 -1.25503213e-01  1.52807298e-02 -8.47085117e-04  1.60809492e-05]
MSE(train)= 0.06759853879817167
MSE(test)= 0.11453445401536166

When degree= 10
w 10 = [ 6.28104530e-03  6.54193530e+00 -1.64579009e+01  1.80400445e+01
 -1.06255253e+01  3.71504543e+00 -8.09168766e-01  1.10852572e-01
 -9.28047527e-03  4.33547963e-04 -8.65667249e-06]
MSE(train)= 0.047898416569504755
MSE(test)= 0.1372249828934701
```

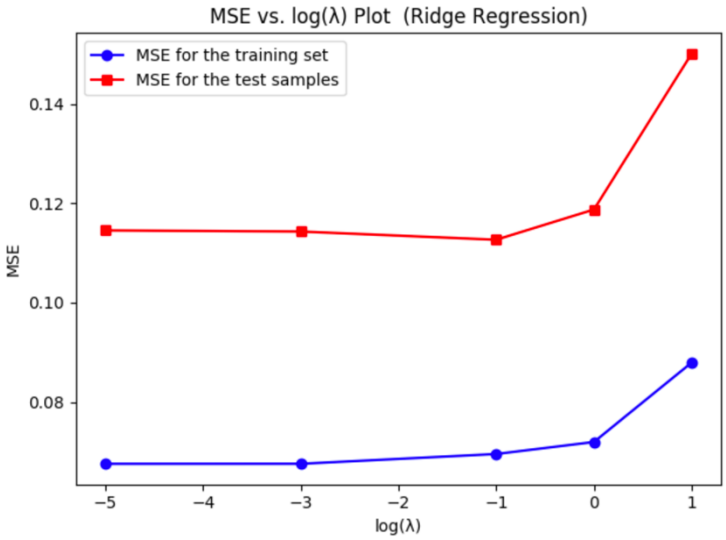


From the plot above, when polynomial degree is equal to 10, it's the best model based on the training sample MSE only. However, it's the best model based on the test sample MSE only when polynomial degree is equal to 3.

(g) Now, let's fix the polynomial degree to 7. Solve using ridge regression with penalty term $\lambda = [10^{-5}, 10^{-3}, 10^{-1}, 1, 10]$. Show the computed weights.

(h) Compute train and test MSE of the fit from part (g) and plot both vs $\log(\lambda)$. What are your conclusions?

```
Question1.(g)~(h)

When λ= 1e-05
w= [ 1.39722954e-01  1.39401406e+00 -1.22822857e+00  5.21447110e-01
 -1.25409398e-01  1.52678852e-02 -8.46198828e-04  1.60566527e-05]
MSE(train)= 0.06759853950300042
MSE(test)= 0.11453199639033475

When λ= 0.001
w= [ 1.46572053e-01  1.34468110e+00 -1.16360339e+00  4.87488371e-01
 -1.16572783e-01  1.40582115e-02 -7.62738908e-04  1.37689100e-05]
MSE(train)= 0.06760493124837373
MSE(test)= 0.11430580140409514

When λ= 0.1
w= [ 2.67755109e-01  5.05409144e-01 -1.02239087e-01 -5.65926056e-02
  2.24703565e-02 -4.71641837e-03  5.18727311e-04 -2.10549261e-05]
MSE(train)= 0.06957497836996134
MSE(test)= 0.11266052465043398

When λ= 1
w= [ 2.83302412e-01  2.51242694e-01  6.83721533e-02 -7.68300114e-02
  1.41197188e-02 -2.14577212e-03  2.64068828e-04 -1.23729062e-05]
MSE(train)= 0.0719924092759618
MSE(test)= 0.11873599096092358

When λ= 10
w= [ 1.30648926e-01  1.07289568e-01  7.77265364e-02  2.03327704e-02
 -3.10453463e-02  6.03166173e-03 -4.02835140e-04  8.05983975e-06]
MSE(train)= 0.08796191544222298
MSE(test)= 0.15008628601090948
```



From the plot above, when log(λ) =-5, it's the best model based on the training sample MSE only. However, it's the best model based on the test sample MSE only when log(λ) =-1. Besides, we can find that as log(λ) increases, the MSE for the training set also increases, but it's not always true for the test samples.

2. Murphy Exercise 7.4. **Hint:** Start from Murphy Eq. (7.8), and assume $\hat{w}$ is given.

Exercise 7.4 MLE for σ^2 for linear regression

Show that the MLE for the error variance in linear regression is given by

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\hat{\mathbf{w}})^2 \tag{7.96}$$

This is just the empirical variance of the residual errors when we plug in our estimate of $\hat{\mathbf{w}}$.

$$\ell(\theta) = \sum_{i=1}^{N} \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} e^{-\frac{(y_i - \underline{w}^T\underline{x}_i)^2}{2\sigma^2}}\right] = \frac{-1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \underline{w}^T\underline{x}_i)^2 - \frac{N}{2}\log(2\pi\sigma^2)$$

$$\frac{d\,\ell(\theta)}{d\sigma} = (-2)\times\frac{-1}{2}\times\sigma^{-3}\sum_{i=1}^{N}(y_i - \underline{w}^T\underline{x}_i)^2 - \frac{N}{2}\times\frac{1}{2\pi\sigma^2}\times 4\pi\sigma = 0$$

$$\Rightarrow \frac{1}{\sigma^3}\sum_{i=1}^{N}(y_i - \underline{w}^T\underline{x}_i)^2 = \frac{N}{\sigma}$$

$$\Rightarrow \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \underline{w}^T\underline{x}_i)^2 \quad Q.E.D.$$

3. *Bayesian concept learning.* Read Murphy 3.1, 3.2 up to first paragraph of 3.2.4, inclusive. The rest of 3.2 is optional.

Key concepts (to focus on during reading):

- What learning is
- Hypothesis space
- Version space
- Strong sampling assumption
- Likelihood
- Prior
- Posterior
- Posterior predictive distribution
- How these combine to give a prediction probability

(a) For the numbers game, take as the hypothesis set $\mathcal{H}$:

$$\mathcal{H} = \{h_{odd}, h_{even}, h_2, h_{P2}, h_5, h_{P5}, h_7, h_{P7}\}$$

in which

h_{odd} = all odd numbers

h_{even} = all even numbers

h_2 = all numbers ending in 2

h_{P2} = all powers of 2 (excluding 2^0)

h_5 = all numbers ending in 5

h_{P5} = all powers of 5 (excluding 5^0)

h_7 = all numbers ending in 7

h_{P7} = all powers of 7 (excluding 7^0)

(b) Also for the numbers game, let the training data $\mathcal{D}=\{16\}$. Suppose the hypothesis space $\mathcal{H}=\{h_{P2},h_{P4}\}$, in which:

$$h_{P2}=\{2,4,8,16,32,64\}$$
$$h_{P4}=\{4,16,64\}$$

Assume priors are $p(h_{P2})=0.6$, $p(h_{P4})=0.4$, and use the strong sampling assumption.

(i) Calculate the likelihood and the posterior for h_{P2}

(ii) Calculate the likelihood and the posterior for h_{P4}

(iii) Which posterior is larger?

(a) $\mathcal{D}=\{5,25\}$
version space: h_{odd}, h_5, h_{P5}

(b) $\mathcal{D}=16$, $H=\{h_{P2},h_{P4}\}$, $h_{P2}=\{2,4,8,16,32,64\}$, $h_{P4}=\{4,16,64\}$
$P(h_{P2})=0.6$, $P(h_{P4})=0.4$

(i) likelihood of $h_{P2}=p(\mathcal{D}|h_{P2})=1/6$

posterior for $h_{P2}=P(h_{P2}|\mathcal{D})=\frac{P(\mathcal{D}|h_{P2})P(h_{P2})}{P(\mathcal{D})}=\frac{P(\mathcal{D}|h_{P2})P(h_{P2})}{P(\mathcal{D}|h_{P2})P(h_{P2})+P(\mathcal{D}|h_{P4})P(h_{P4})}$

$=\frac{1/6\times 6/10}{1/6\times 6/10+1/3\times 4/10}=\frac{\frac{1}{10}}{\frac{7}{30}}=\frac{1}{10}\times\frac{30}{7}=\frac{3}{7}$

(ii) likelihood of $h_{P4}=P(\mathcal{D}|h_{P4})=1/3$

posterior for $h_{P4}=P(h_{P4}|\mathcal{D})=\frac{P(\mathcal{D}|h_{P4})P(h_{P4})}{P(\mathcal{D}|h_{P4})P(h_{P4})+P(\mathcal{D}|h_{P2})P(h_{P2})}=\frac{1/3\times 4/10}{1/6\times 6/10+1/3\times 4/10}$

$=\frac{2}{15}\times\frac{30}{7}=\frac{4}{7}$

(iii) The posterior of h_{P4} is larger

4. *Bayesian linear regression.* Read Murphy 7.6.0, 7.6.1, 7.6.2.

To get an overview of the algebra from Eq. (7.54) to Eq. (7.55), show that $p(\underline{w}|\underline{\underline{X}},\underline{y},\sigma^2)$ can be written in terms of $p(\underline{y}|\underline{\underline{X}},\underline{w},\sigma^2)$ and a prior term. Label the posterior, likelihood, and prior terms. **Do not** assume Gaussian densities in this problem.

$$\underbrace{P(\underline{w}|\underline{\underline{X}},\underline{y},\sigma^2)}_{\text{posterior}}=P(\underline{\underline{X}},\underline{y},\sigma^2|\underline{w})\cdot\frac{P(\underline{w})}{P(\underline{\underline{X}},\underline{y},\sigma^2)}=P(\underline{y}|\underline{\underline{X}},\sigma^2,\underline{w})\cdot P(\underline{\underline{X}},\sigma^2|\underline{w})\cdot\frac{P(\underline{w})}{P(\underline{\underline{X}},\underline{y},\sigma^2)}$$

$P(\underline{\underline{X}},\sigma^2|\underline{w})$: can be dropped

$$=\underbrace{P(\underline{y}|\underline{\underline{X}},\sigma^2,\underline{w})}_{\text{likelihood}}\cdot\underbrace{P(\underline{w})}_{\text{prior}}\cdot\boxed{\frac{P(\underline{\underline{X}},\sigma^2)}{P(\underline{\underline{X}},\underline{y},\sigma^2)}}$$

constant