Supervised Learning - Coursework 1

Yinji Zhu 18062537 yinji.zhu.18@ucl.ac.uk
Yao Zhao 18071889 yao.zhao.18@ucl.ac.uk

November 6, 2018

1 PART I

Linear Regression

1. a. Knowing the data set, two vectors are generated to hold values of x and y axes respectively. A function $Function\_fit$ is created to calculate values of w. Superimposing the four different curves in a graph, the graph can be seen in Figure 1.

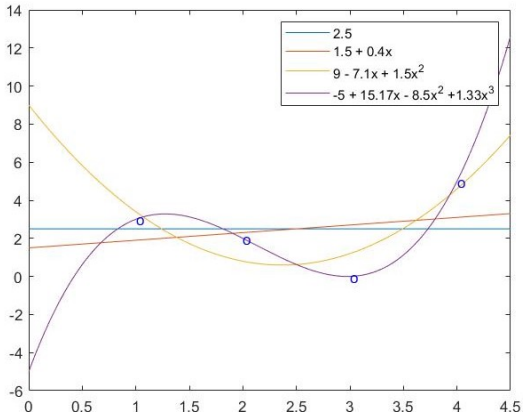

Figure 1: Data set (1,3),(2,2),(3,0),(4,5) fitted with bases of dimension 1, 2, 3, 4.

b. When K = 1, the equation is: $y = 2.5$. When K = 2, the equation is: $y = 1.5 + 1.4x$. When K = 3, the equation is: $y = 9 - 7.1x + 1.5x^2$ When K = 4, the equation is:$y = -5 + 15.17x - 8.5x^2 + 1.33x^3$

c. Firstly, SSE can be obtained, which is:

k	1	2	3	4
SSE	13	12.2	3.2	0.0468

$MSE = SSE/m$,the mean square error can be seen in the table:

k	1	2	3	4
MSE	3.25	3.05	0.8	0.0117

2. a. i. The graph of the function $sin^2(2\pi x)$ in the range$0 \leq x \leq 1$can be seen in Figure.2. The points with random variable are also superimposed in the graph.

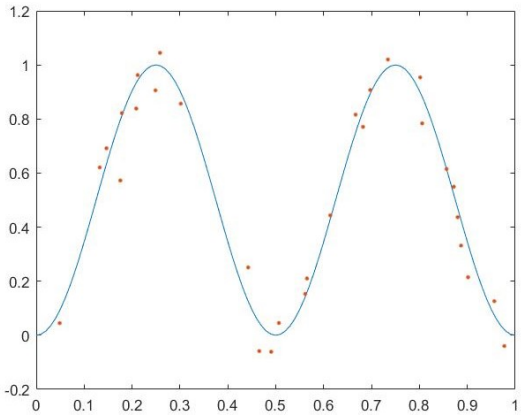

Figure 2: $sin^2(2\pi x)$with the points existing noise

ii. Set the polynomial bases of dimension to $k = 2, 5, 10, 14, 18$, the five curves can be seen in Figure 3.

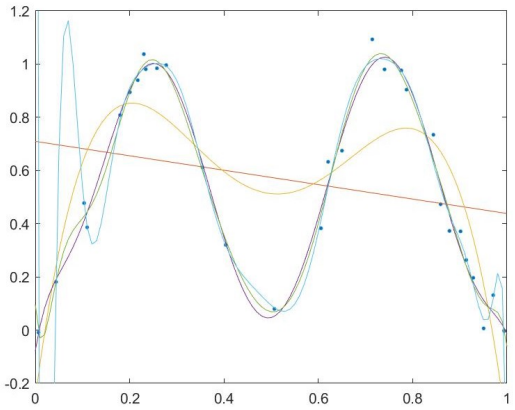

Figure 3: bases of dimension $k = 2, 5, 10, 14, 18$

b. Changing the polynomial dimension to $k = 1, ..., 18$, the MSE of the fitting of the data set S with polynomial basis of dimension k is obtained. The graph of the log of the training error versus k can be seen in Figure 4.

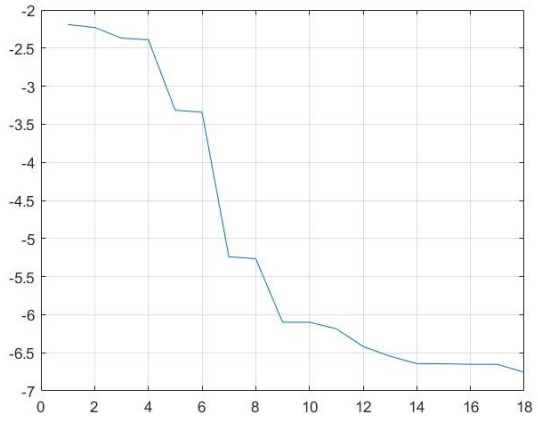

Figure 4: the training error versus the polynomial dimension

c. Change the test set to 1000, the MSE graph with overfitting can be seen In figure 5. As what can be seen in the graph, the equation starts to increase when k is around 11.

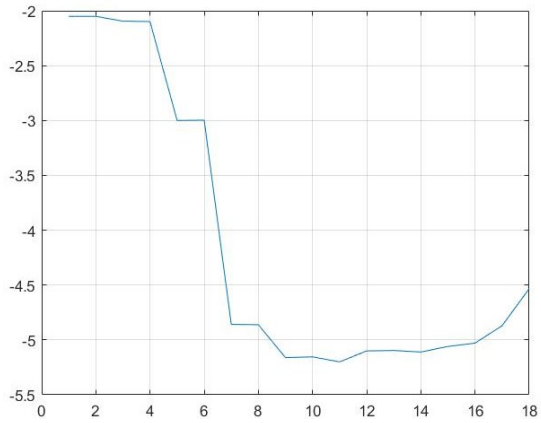

Figure 5: overfitting situation

d. Running the programs of 2b and 2c 100 times, the average results are obtained and plotted in the Figure 6. The curves of the results are more smooth comparing with the 2b and 2c.

3. a. Change the bases to $sin(1\pi x), sin(2\pi x)....sin(k\pi x)$, the MSE of the fitting of the data set S with polynomial basis of dimension k is obtained. The graph of the log of the training error versus k can be seen in Figure 7.

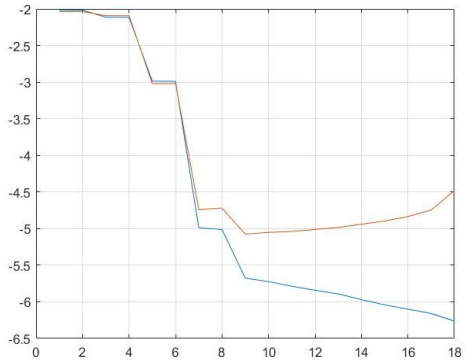

Figure 6: the average results of 100 runs

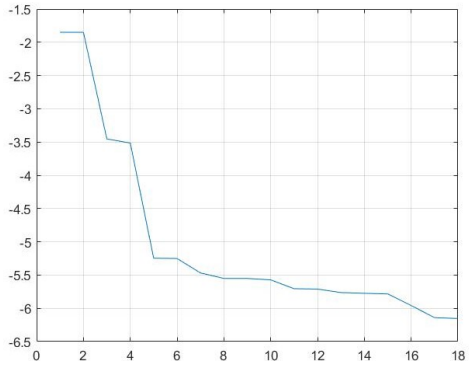

Figure 7: the training error versus the polynomial dimension

b. Change the test set to 1000, the MSE graph with overfitting can be seen In figure 8. As what can be seen in the graph, the equation starts to increase when k is around 11.

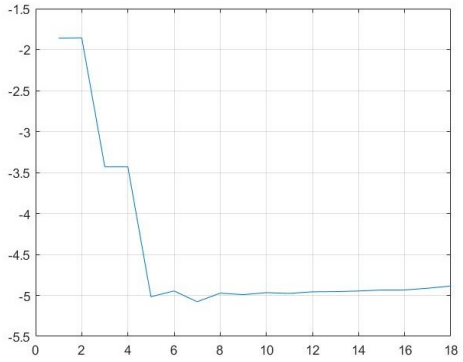

Figure 8: overfitting situation

c. Running the programs of 3a and 3b 100 times, the average results are obtained and plotted in the Figure 9. The curves of the results are more smooth comparing with the 3a and 3b.

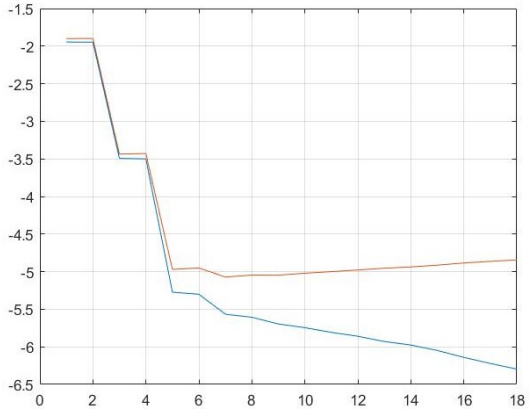

Figure 9: the average results of 100 runs

Boston housing and kernels

4. a. There are 506 entries in the Boston housing dataset. Knowing that the training set should be $2/3$ and the test set should be $1/3$, 338 entries are put into the training and 168 entries are put into the test set. After averaging the results in 20 times, the MSE on the training set is around 74.69 and the MSE on the test set is around 64.32.

   b. The constant value is obtained through calculating w using values in the training set. Using this value, MSEs on the training and test sets can be obtained. Running it over 20 times, the average results can be obtained. Since the MSEs are calculated without attributes, values are really large. It illustrates that this regression will generate a large variation between y_{test} and y_{expect}. The value on the training set is similar to the value on the test set, since they use the related dataset. It show that the naive regression can be used in the test set.

   c. After adding a bias term, the inputs are augmented with an additional 1 entry $(x_i, 1)$. Consequently, the linear regression with single attributes can be obtained. Averaging the results in 20 runs, the average weight vectors in each attribute can be got.

k	1	2	3	4	5	6	7
w_0	24.04	20.78	29.77	22.05	41.52	-34.09	31.17
w_1	-0.441	0.145	-0.657	6.249	-34.40	9.007	-0.127
k	8	9	10	11	12	13	
w_0	18.22	26.34	32.94	62.95	10.48	34.56	
w_1	1.112	-0.405	-0.026	-2.194	0.034	-0.952	

   d. Using all attributes to do linear regression after incorporating a bias term, the MSE values can be calculated, which are:
   MSE on the training set is: 8.7312
   MSE on the test set is: 15.4070
   Which are much smaller than the MSE values in naive regression.

Kernelised ridge regression

5. a. A vector of γ ranging from 2^{-40} to 2^{-26} and a vector of θ ranging from 2^7 to 2^{13} are created respectively. The vector γ has 15 entries and the vector θ has 13. To choose the indices of the γ and σ values that perform the best to compute the predictor, a two dimensions' vector is created to store cross validation error. Using five-fold cross-validation to perform kernel ridge regression on the training set, the training set is divided into five parts. Four of five samples use for training, and the single one use for validating. The procedure can be seen in matlab and the minimum value is 2.4914 which has values of $\gamma = 2^{-31}$ and $\sigma = 2^{10}$.

   b. Using the function surf(), the cross validation error as a function of γ and σ can be seen in Figure 10.

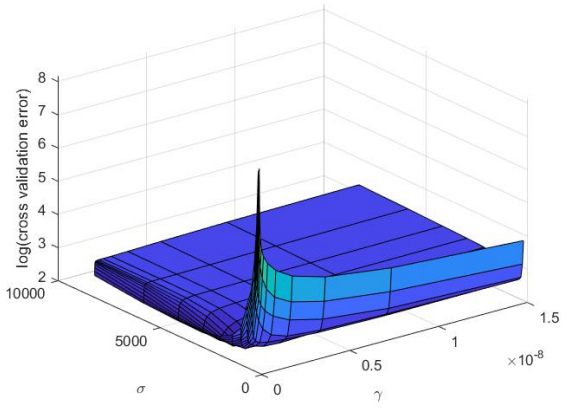

Figure 10: the cross validation error as a function of γ and σ

   c. Using the best γ and σ which are found in 5a, values of MSEs can be obtained, which are:
   On the training set, MSE = 7.8430
   On the test set, MSE = 13.7513

   d. Repeat the previous exercise over 20 random, the results can be seen in the table with train/test error and the standard deviations of the train/test errors:

Method	MSE train	MSE test
Naive Regression	74.9031±3.5437	96.0146±38.4531
Linear Regression (attribute 1)	71.5376±5.7926	73.1687±12.1088
Linear Regression (attribute 2)	73.3346±5.4684	73.1687±11.2065
Linear Regression (attribute 3)	64.8086±6.1070	74.2102±12.3552
Linear Regression (attribute 4)	81.2515±5.2276	64.9141±10.9012
Linear Regression (attribute 5)	68.9864±5.9938	83.8036±12.1528
Linear Regression (attribute 6)	43.1038±4.7698	69.5725±9.8765
Linear Regression (attribute 7)	72.5927±6.3649	45.1248±12.8429
Linear Regression (attribute 8)	79.0455±6.457	72.6746±13.1095
Linear Regression (attribute 9)	72.2299±6.5978	79.9446±13.3904
Linear Regression (attribute 10)	66.0157±4.9605	72.5855±13.4700
Linear Regression (attribute 11)	62.7385±5.6750	66.2624±10.2517
Linear Regression (attribute 12)	74.6100±6.4873	63.0432±11.6439
Linear Regression (attribute 13)	38.7510±2.9157	76.3692±6.1609
Linear Regression (all attributes)	9.0826±13.5792	18.3533±25.8583
Kernel Ridge Regression	7.0376±0.9177	13.9386±3.5989

2 PART II

6. a. Based on the definition of the imbalanced classification loss function from the question,

$$L_c(y, \hat{y}) := [y \neq c]c_y = \begin{cases} 0, & y = \hat{y} \\ c_y, & y \neq \hat{y} \end{cases}$$

Given that y is ground truth while $\hat{y}$ is the predictor. Then the expected error will be

$$\varepsilon(f) = \mathbf{E}[L_c(y, \hat{y})] = \int L_c(y, \hat{y}) P(x, y)\, dx\, dy$$

Applying the decomposition $P(y, x) = P(y|x)P(x)$ so that

$$\varepsilon(f) = \int_x \left\{ \int_y L_c(y, \hat{y})\, dP(y \mid x) \right\} dP(x) = E_x \sum_{k=1}^{k} [L_c(y, \hat{y})] P(y \mid x)$$

To get the Bayes estimator, we need to find the minimum of $\varepsilon(f)$ by minimizing $x \in X$ one by one. Therefore, the Bayes estimator is

$$\begin{aligned} f^*(x) &= \arg\min_{y \in Y} \sum_{k=1}^{k} [L_c(y, \hat{y})] P(y \mid x) \\ &= \arg\min_{y \in Y} \sum_{k=1}^{k} P(y \neq \hat{y} \mid x) \\ &= \arg\min_{y \in Y} \sum_{k=1}^{k} P(y \neq \hat{y} \mid x) \\ &= \arg\min_{y \in Y} (1 - P(y = \hat{y} \mid x)) \\ &= \arg\max_{y \in Y} P(y = \hat{y} \mid x) \\ &= \arg\max_{y \in Y} P(y \mid x) \end{aligned}$$

The Bayes estimator is the mode of the posterior distribution.

b. In this subquestion, the loss function has turned to $L_c(y, \hat{y}) = |y - \hat{y}|$. To be the same as last question,

$$\begin{aligned} \varepsilon(f) &= \int_x \left\{ \int_y L_c(y, \hat{y})\, dP(y \mid x) \right\} dP(x) \\ &= \sum_{x \in X} \left\{ \sum_{y \in Y} [L_c(y, \hat{y})] P(y \mid x) \right\} P(x) \\ &= \sum_{x \in X} \left\{ \sum_{y \in Y} |y - \hat{y}| P(y \mid x) \right\} P(x) \end{aligned}$$

Then find the value of $f^*(x)$ at a specific point so that setting x := x',

$$\varepsilon(f(x')) = \sum_{x \in X} \left\{ \sum_{y \in Y} |y - f(x')| P(y \mid x') \right\} P(x')$$

Set e :=$\varepsilon(f(x'))$ and z := f(x'), and replace "=" with "$\propto$"

$$e \propto \sum_{y \in Y} |y - z|\, P(y \mid x')$$

To find the Bayes estimator we need to indicate the minimum of the absolute value for $|y - z|$ which is 0. Hence,

$$0 = \sum_{-\infty}^{z} (y - z)\, P(y \mid x') + \sum_{z}^{\infty} (z - y)\, P(y \mid x')$$

Hence, the Bayes estimator $\hat{y}$ will be the median of the data set Y. There may be two different situations to find median, since the median depends on the parity of the data set Y. For instance, the median will be the mean of the two middle points if the number of elements in Y is even. Besides, the centre of Y is the median of an odd data set. On account of the definition of PMF, if $\hat{y}$ is the median so that $P(\hat{y}|x) = 0$

7. a. When K_c is a positive semidefinite kernel, c should satisfy that $c \geq 0$.
Based on the definition of the positive semidefinite kernel, $\mathbf{K}$ is positive semidefinite if and only if

$$K(x,\, t) =< \phi(x),\, \phi(y) >,\, x, t \in R$$

Then we have that

$$\sum_{i,j=1}^{m} K(x_i, x_j) = \Big\langle \sum_{i=1}^{m} c_i\phi(x_i), \sum_{j=1}^{m} c_j\phi(x_j) \Big\rangle = ||sum_{i=1}^{m} c_i\phi(x_i)||^2 \geq 0$$

For this question,

$$\begin{aligned} K_c(x,\, z) &= c + \sum_{i=1}^{n} x_i z_i \\ &= \sqrt{c}\sqrt{c} + x_1 z_1 + x_2 z_2 + x_2 z_2 + \cdots + x_n z_n \\ &=< \phi(x), \phi(z) > \end{aligned}$$

Due the definition of square root, $\sqrt{c} \geq 0$ so that $c \geq 0$.

b. Considering the linear regression of the kernel function K_c, and c will have no obvious influence the solution. After adding c to every kernels, which will change the mean of the dataset. However, the mean square error will not be influenced because of the definition of MSE. The offset c have been added to every item, so that the difference between each item will stay still.

8. The k-Nearest Neighbor classifier is defined as,

$$y = \arg\max_{v \in V} \sum_{i=1}^{k} w_i \delta(v, y(i))$$

v is the sample, and there is a training set$\{(x_i, y_i) \in R, i = 1, 2, 3, \cdots, n\}$. Besides, the weight is the distance between v and x_i, $w_i = \{\frac{1}{d^2(x,x_i)}, i = 1, 2, 3, \cdots, k\ \}$.
$\delta(a, b) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$
For 1-Nearest Neighbor classifier in this question which k =1,

$$y = \arg\max_{v \in V} w_i \delta(v, y(i)),\, i = \{1, 2, \cdots, n\}, y \in \{1, -1\}$$

In this case, y_i is the ith training set so that the training weight is the Eucliden distance $w_i = \frac{1}{||v-x_i||^2}$. Based on this equation, the less the distance is, the higher the weight will be. Additionally, Given the Gaussian kernel function,

$$K(x,t) = e^{-\beta\,||\mathbf{x}-\mathbf{t}||},\ \beta > 0, \mathbf{x}, \mathbf{t} \in R$$

For this case,

$$K(x,t) = e^{-\beta\,||\mathbf{x}-\mathbf{t}||}\ \beta > 0, \mathbf{x}, \mathbf{t} \in R$$

Then, the evaluation of the regression function on the test point is:

$$y_{test} \propto sign\{\sum_{i=1}^{L} \alpha_i K(x_i, x_{test})\} = sign\{\sum_{i=1}^{L} \alpha_i\, e^{-\beta\,||\mathbf{x}-\mathbf{t}||}\}$$

To be similar with 1-NN, for $e^{-\beta\,||\mathbf{x}-\mathbf{t}||}$, β should be as big as possible for $\beta > 0$. Hence, the distance between x_i and x_{test} will be smaller so that the weight will increase. Nevertheless, bigger β may cause over-fitting while smaller β could lead under-fitting.
Therefore, β should be taken as bigger as possible.

9. To initialize a $n \times n$ board, a table M is designed to indicate the status of the mole. To be clearly,'0' indicates the mole hide underground while '1' means they pop-up.

M	1	2	3	$\cdots$	n-2	n-1	n
1	0	1	1	$\cdots$	0	1	0
2	1	0	1	$\cdots$	1	1	0
3	1	1	0	$\cdots$	0	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n-2	0	0	1	$\cdots$	0	0	0
n-1	1	1	0	$\cdots$	0	0	1
n	1	0	1	$\cdots$	1	0	1

Based on the given game rule, there are two obvious law can be found.
(a.) For a particular position, the results will be same if you knock it 1, 3 or 5 times. Besides, whacking a mole can only influence the adjacent holes. Hence, if a mole need to be whacked, it is no need to knock it 3 or 5 times more which saving the process since they have same effects.
(b.) For a specific hole, the status is only related to the whack of itself and its 4 adjacent holes.It will not be influenced by the sequence of whack.
Assume a function h(m, w) m= 0, 1w= 0, 1, which m is the status of holes. Then, w=1 means m will change and m stays still if w=0. $M_{a,b}$ is the hole position in the board and $W_{a,b}$ is the whack point. Therefore,
h(1, 0)=1; // stay still
h(1, 1)=0; //m change after whack
h(0, 1)=1; //m change after whack
h(0, 0)=0; //stay still

According to the four situations above, h(m, w) is what XOR arithmetic operation. Focusing on the first hole $M_{1,1}$, only $W_{1,1}$, $W_{1,2}$, $W_{2,1}$ may affect the status of $M_{1,1}$. Hence,

$$h(W_{2,1}, h(W_{1,2}, h(W_{1,1}, M_{1,1}))) = 0$$

Since h(m, w) is XOR so that the equation can be shown as,

$$\begin{aligned} W_{2,1} \wedge W_{1,2} \wedge W_{1,1} \wedge M_{1,1} &= 0 \\ W_{2,1} \wedge W_{1,2} \wedge W_{1,1} &= M_{1,1} \end{aligned}$$

Hence, for the initialized $n \times n$ broad about, the whole equations can be indicated using matrix production.

$$\begin{bmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,n} & W_{2,1} & \cdots & W_{2,n} & W_{n,1} & \cdots & W_{n,n} \\ W_{1,1} & W_{1,2} & \cdots & W_{1,n} & W_{2,1} & \cdots & W_{2,n} & W_{n,1} & \cdots & W_{n,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{1,1} & W_{1,2} & \cdots & W_{1,n} & W_{2,1} & \cdots & W_{2,n} & W_{n,1} & \cdots & W_{n,n} \end{bmatrix} \begin{bmatrix} W_{1,1} \\ W_{1,2} \\ \vdots \\ W_{n,n} \end{bmatrix} = \begin{bmatrix} M_{1,1} \\ M_{1,2} \\ \vdots \\ M_{n,n} \end{bmatrix}$$

$$Wx = M$$

Which W is a $n^2 \times n^2$ matrix, x and M are both $n^2 \times 1$ dimension. Since the all equations only has XOR arithmetic operation, Gaussian elimination will be an appropriate methods to solve it. For example,

$$x_1 \wedge x_2 \wedge x_3 = y_1$$

$$x_1 \wedge x_2 = y_2$$

Combine these two equations with XOR operation, and then

$$x_1 \wedge x_2 \wedge x_3 \wedge x_1 \wedge x_2 = x_3 = y_1 \wedge y_2$$

Therefore, Gaussian elimination can be used to solve the whole matrix production. Additionally, due to the input is n, the time complexity and space complexity will be

$$\begin{aligned} T(n) &= O(n^2) \\ S(n) &= O(n^2) \end{aligned}$$

END OF COURSEWORK