COMP0078: Supervised Learning **2023/24**

Coursework - ⟨1⟩

Due Date: ⟨**10 November 2023**⟩ *Student(s):* ⟨*23100379, 23092186*⟩

Part I

Question 1

a)

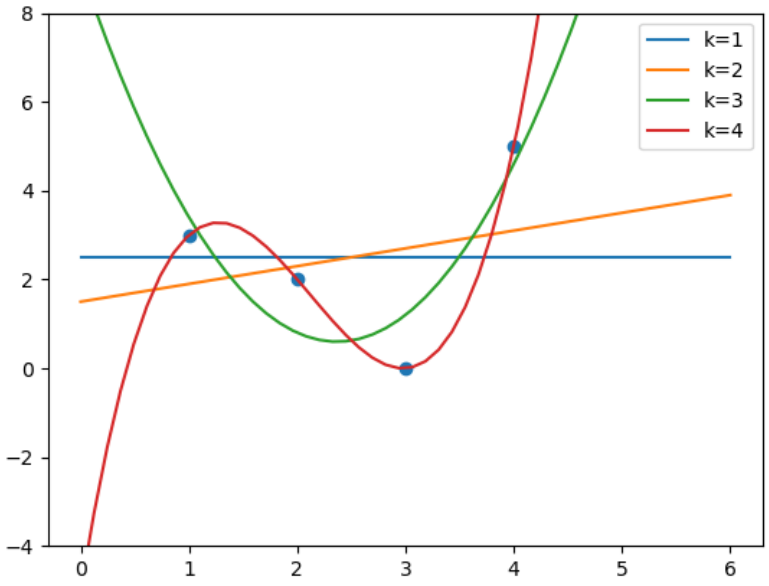

Figure 1.1. Fitting of polynomial regression up to degree 4 for given data points

b) Equations of polynomials $k = 1, 2, 3$

$$y = 2.5$$
$$y = 0.4x + 1.5$$
$$y = 1.5x^2 - 7.1x + 9$$

c)

k	MSE
1	3.25
2	3.05
3	0.8
4	0

Question 2

ai)

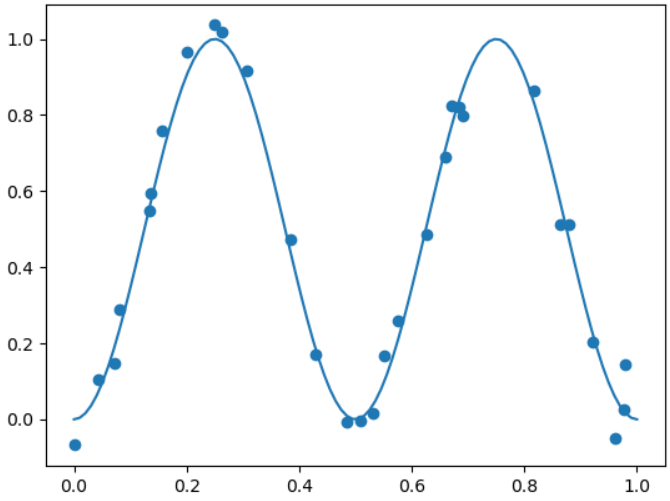

Figure 1.2. The function $sin^2(2\pi x)$ and datapoints generated from it with white noise of 0.07 standard deviation added

aii)

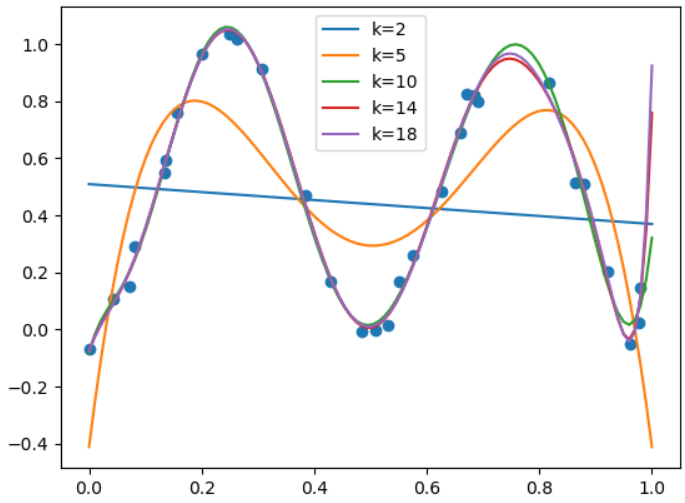

Figure 1.3. Datapoints generated from $sin^2(2\pi x)$ with white noise of 0.07 standard deviation added. Polynomials of degree 2,5,10,14,18 fitted and plotted

b)

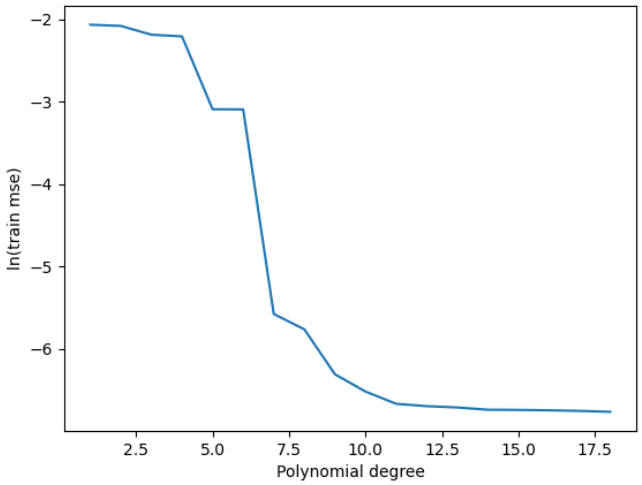

Figure 1.4. Natural log of the training mean squared error for polynomial regression of degree 1 to 18 on datapoints generated from $sin^2(2\pi x)$ with white noise of 0.07 standard deviation added

c)

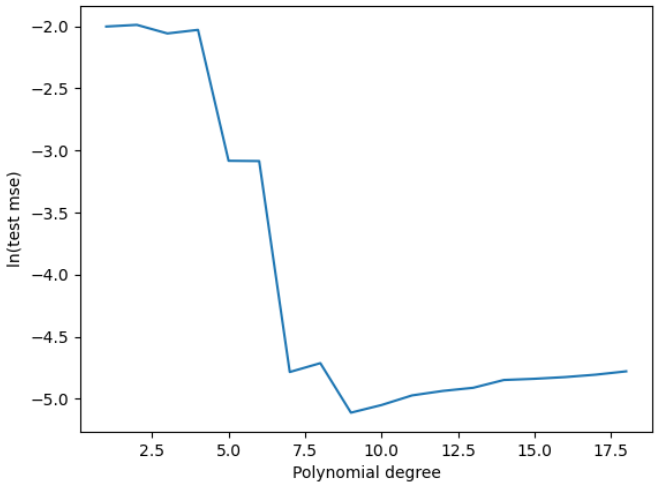

Figure 1.5. Natural log of the test mean squared error for polynomial regression of degree 1 to 18 on datapoints generated from $sin^2(2\pi x)$ with white noise of 0.07 standard deviation added

d)

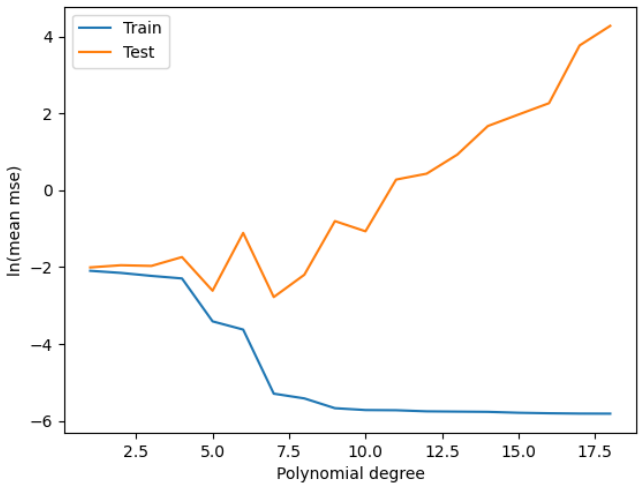

Figure 1.6. Natural log of the mean train and test mean squared error for polynomial regression of degree 1 to 18 on datapoints generated from $sin^2(2\pi x)$ with white noise of 0.07 standard deviation added over 100 datasets.

Question 3

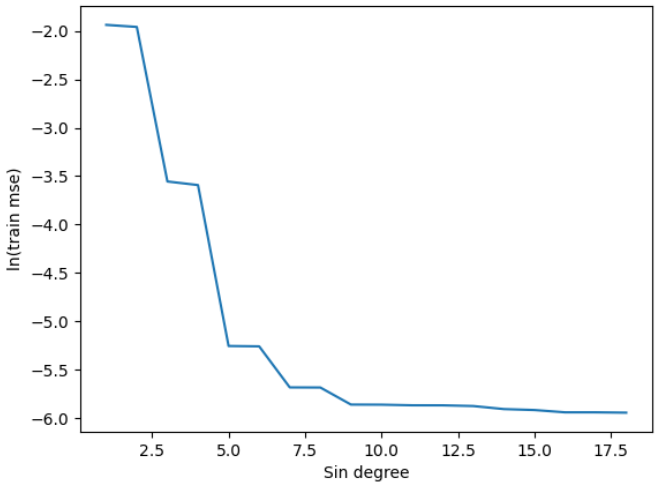

Figure 1.7. Natural log of the mean train mean squared error for linear regression with sin feature maps of frequency 1 to 18 on datapoints generated from $sin^2(2\pi x)$ with white noise of 0.07 standard deviation added over 100 datasets.

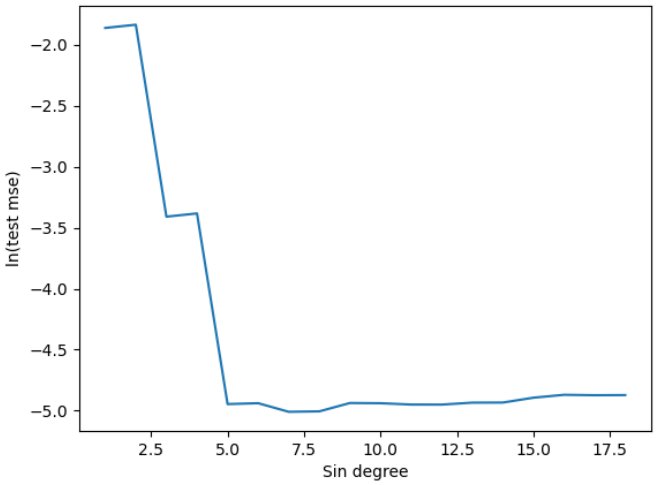

Figure 1.8. Natural log of the mean test mean squared error for linear regression with sin feature maps of frequency 1 to 18 on datapoints generated from $sin^2(2\pi x)$ with white noise of 0.07 standard deviation added over 100 datasets.

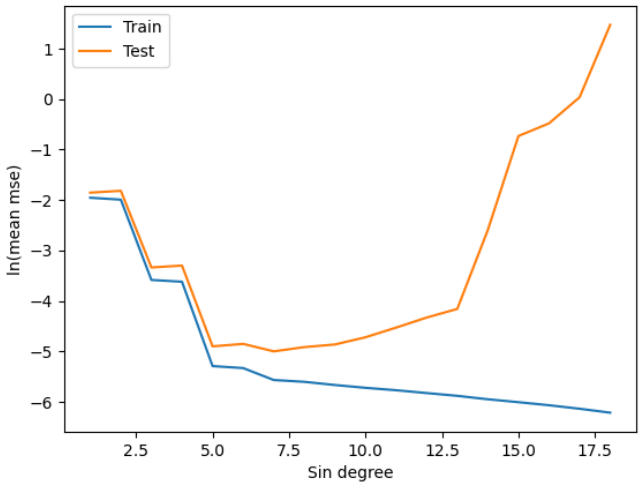

Figure 1.9. Natural log of the mean train and test mean squared error for linear regression with sin feature maps of frequency 1 to 18 on datapoints generated from $sin^2(2\pi x)$ with white noise of 0.07 standard deviation added over 100 datasets.

Question 4

a) Naive regression Train MSE 84.47, Test MSE 84.54.

b) When the feature transformation is to a bias term 1 instead of to a feature, then the weight computed is the average of all the training y values, and prediction is simply that average. This can be shown in the equation for solving for the weights of a linear equation

$$\mathbf{w} = (\mathbf{1}_n^T \mathbf{1}_n)^{-1} \mathbf{1}_n^T y = (n)^{-1} \sum_{i=1}^{n} y_i = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Any prediction is then

$$\mathbf{w}^T \phi_1(x) = (\frac{1}{n} \sum_{i=1}^{n} y_i) \times 1$$

c) d)

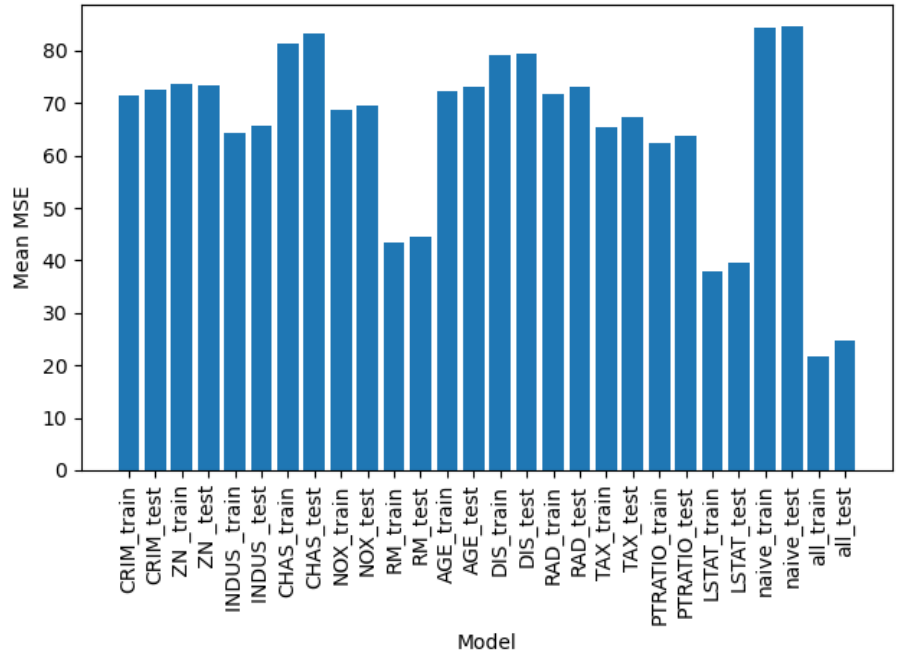

Figure 1.10. Train and test MSE of naive, single feature, and all features linear regression on filtered boston dataset

CRIM train	71.53
CRIM test	72.67
ZN train	73.70
ZN test	73.38
INDUS train	64.30
INDUS test	65.74
CHAS train	81.31
CHAS test	83.37
NOX train	68.82
NOX test	69.64
RM train	43.35
RM test	44.47
AGE train	72.23
AGE test	73.12
DIS train	79.19
DIS test	79.49
RAD train	71.77
RAD test	73.18
TAX train	65.33
TAX test	67.30
PTRATIO train	62.35
PTRATIO test	63.86
LSTAT train	38.04
LSTAT test	39.64
naive train	84.47
naive test	84.54
all train	21.82
all test	24.62

Table showing the train and test MSE of naive, single feature, and all features linear regression on the boston filtered dataset.

Question 5

a) b)

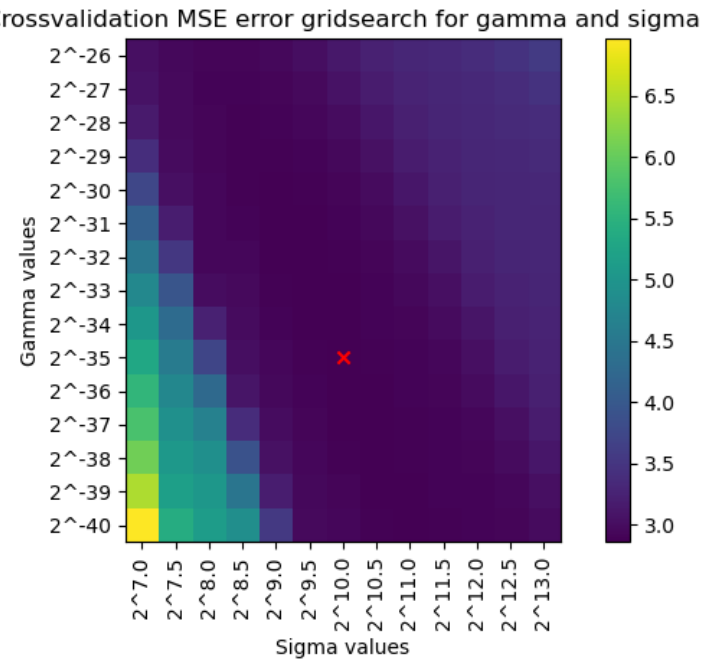

Figure 1.11. Mean MSE of 5-fold cross validation using gaussian kernel ridge regression for each combination of gamma and sigma plotted as a heatmap on boston filtered data. The red x indicates the gamma-sigma value combination with lowest mean MSE

c)

Gaussan Kernel Ridge Regression - $\gamma = 2^{-35}, \sigma = 2^{10}$

Train MSE - 7.30, Test MSE - 12.45 d)

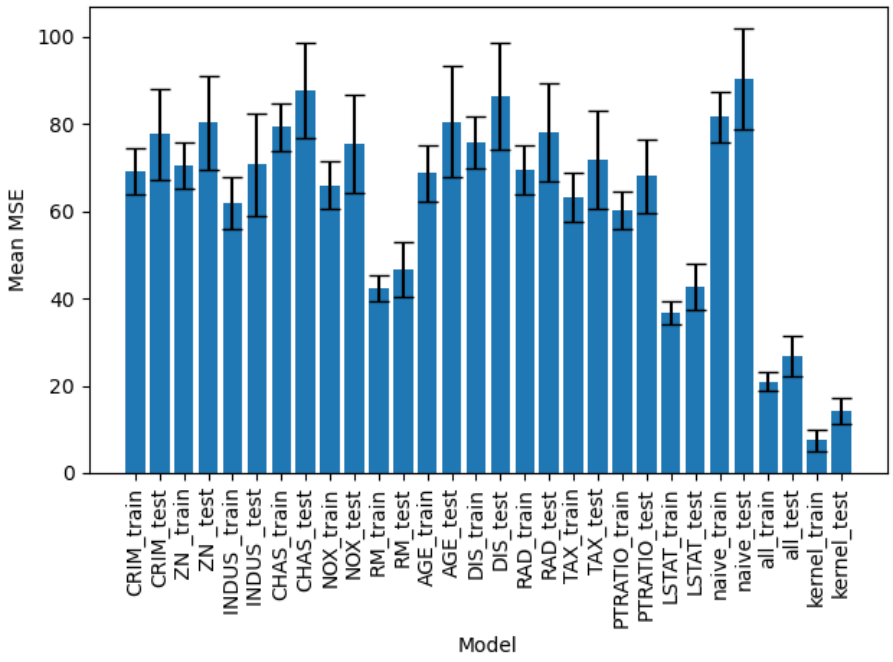

Figure 1.12. Train and test MSE of naive linear regression, single feature linear regression, all features linear regression, and all features gaussian kernel ridge regression over 20 runs on filtered boston dataset. Error bars represent standard deviation of the mean over 20 runs.

Model	Train MSE	Train MSE sd	Test MSE	Test MSE sd
CRIM	69.23	5.29	77.61	10.42
ZN	70.33	5.30	80.25	10.81
INDUS	61.87	5.86	70.74	11.74
CHAS	79.22	5.42	87.61	11.02
NOX	65.98	5.61	75.36	11.22
RM	42.37	3.06	46.61	6.25
AGE	68.67	6.35	80.52	12.87
DIS	75.80	5.97	86.36	12.22
RAD	69.34	5.65	78.08	11.38
TAX	63.12	5.52	71.80	11.14
PTRATIO	60.23	4.23	67.98	8.57
LSTAT	36.64	2.58	42.57	5.26
naive	81.61	5.72	90.25	11.49
all	20.97	2.06	26.84	4.59
kernel	7.51	2.45	14.11	2.92

Table showing train and test mean MSE with standard deviation over 20 runs for naive linear regression, single feature linear regression, all features linear regression, and all feautures gaussian kernel linear regression on the boston filtered dataset.

1.1 Part II

Question 6

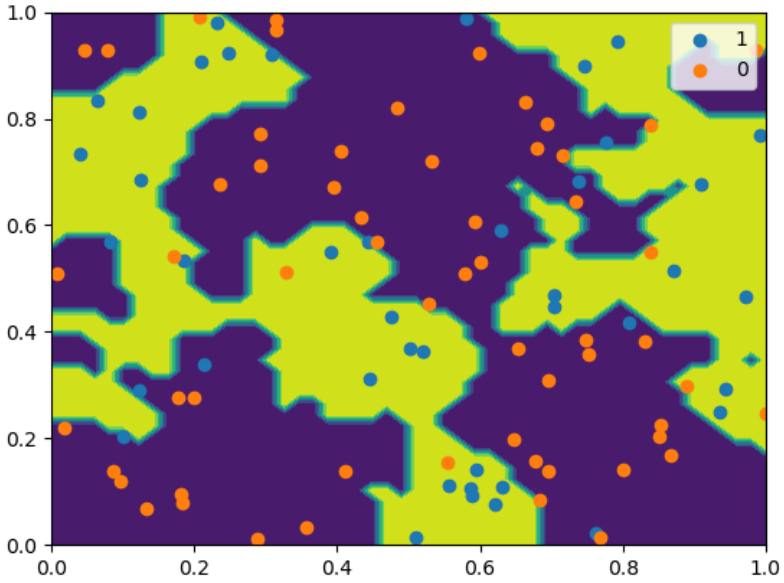

Figure 1.13. Visualising the decision boundaries of 100 points uniformly distributed in $[0, 1]^2$ with binary labels generated with probability 0.5 per class.

Question 7

The graph of mean misclassification rate against values of k is initially high, and then rapidly falls to bottom out at around $k = 10$. It then gradually continues to increase as k increases. The initial high misclassification rate is likely due to the large effect a single erroneous datapoint will have on the mean when k is small. Similarly when k increases, the chances of including datapoints not in the neighbourhood of the test data point increases and introduces noise.

Another interesting pattern is the oscillation of the misclassification rate, where the rate is higher when k is even. This is likely because when k is even there can be ties and then the class is decided arbitrarily. The oscillation reduces in magnitude as k increases in size because the chances of a tie occurring become lower.

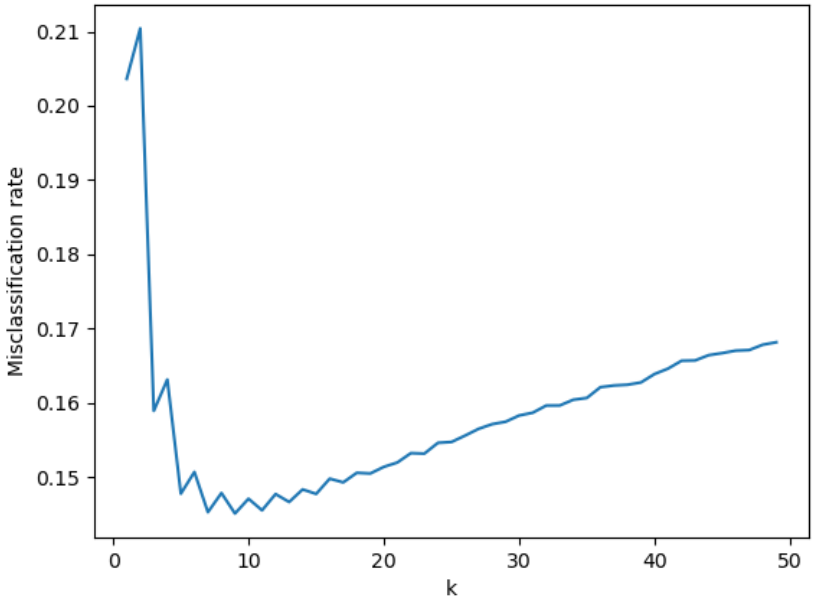

Figure 1.14. Mean misclassification rate against k for training set size of 4000 and test set size of 1000

Question 8

The graph of mean optimal k against training set size is positively correlated. This is likely because as training set size increases, the density of the neighbourhoods increases. As such, increasing k will allow more accurate sampling of the region around a test point without risking introducing noise from regions beyond it. This gives a better approximation of the local conditional mean and hence better performance.

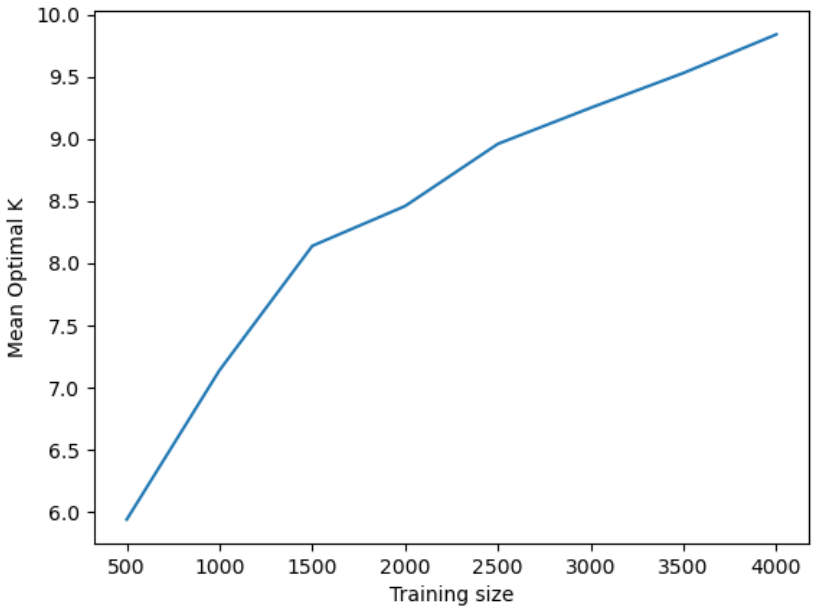

Figure 1.15. Comparing mean optimal k for varying size of training set

1.2 Part III

Question 9

(a) For $c \geq 0, K_c$ is a positive semidefinite kernel.
Proof
The kernel K_c is the sum of 2 kernels

$$K_1 = c, K_2 = \sum_{i=1}^{n} x_i z_i$$

The sum of 2 kernels is also a kernel, so if we prove that K_1 and K_2 are kernels then K_c will also be a kernel.

Proving K_1
K_1 corresponds to a kernel with a 1 dimensional feature map $\phi(\mathbf{x}) = \sqrt{c}, \mathbf{x} \in \mathbb{R}^n$

$$K_1(\mathbf{x}, \mathbf{z}) = \sqrt{c}\sqrt{c} = c$$

When $c \geq 0$ then $K_1 \geq 0, \forall \mathbf{x}, \mathbf{z} \in \mathbb{R}^n$

Proving K_2
K_2 is the standard euclidean inner product for $\mathbb{R}^n$ and is $K_2 \geq 0, \forall \mathbf{x} \in \mathbb{R}^n$ since $x^2 \geq 0, \forall x \in \mathbb{R}$.

Proving K_c
Therefore K_c is a positive semidefinite kernel since $K_c = K_1 + K_2$ and $K_1 \geq 0$ when $c \geq 0$ and $K_2 \geq 0, \forall \mathbf{x} \in \mathbb{R}^n$

(b) Training a linear regression model with least square loss, using kernel K_c on dataset $\mathcal{S}$ with $|\mathcal{S}| = n$ we is the same as performing a linear regression with a transform on the inputs $\phi(x) = \begin{bmatrix} x \\ \sqrt{c} \end{bmatrix}$ in the primal version. This is equivalent to adding a bias term, which means that our bias term becomes $b\sqrt{c}$ so the magnitude of c doesn't matter as long as it is non-zero, since b can be learned to account for that difference as long as $c \neq 0$

Question 10
The problem formulation for this question is that our gaussian kernel regression must satisfy the following equation.

$$y_{nn}(\sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{t})) > 0$$

$$y_{nn}(\alpha_{nn} K(\mathbf{x}_{nn}, \mathbf{t}) + \sum_{i=1, i\neq nn}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{t})) > 0$$

where $y_{nn} \in \{-1, 1\}$ is the label of the nearest neighbour, $\alpha_i, \alpha_{nn} \in \mathbb{R}, \mathbf{x}_i, \mathbf{t}, \mathbf{x}_{nn} \in \mathbb{R}^d, m, d \in \mathbb{Z}^+$.

For the gaussian kernel to act as a 1 nearest neighbour classifier, we need to have the term $\alpha_{nn} K(\mathbf{x}_{nn}, \mathbf{t})$ outweigh the sum of the remaining terms.

For any given training set S and test point $\mathbf{t}$, let the l_2 distance between the nearest neighbour of $\mathbf{t}$ as

$$||\mathbf{x}_{nn} - \mathbf{t}||^2 = d_{nn}^2$$

We can then express the l_2 distance between the test point $\mathbf{t}$ and any other data point in the training set as

$$||\mathbf{x}_i - \mathbf{t}||^2 = (d_{nn} + \delta_i)^2$$

where $\delta_i = ||x_i - t|| - ||x_{nn} - t||$

To compute the alpha values for the regression we take the inverse of the pairwise kernel matrix as shown below

$$\alpha = K^{-1} y$$

Where $K_{i,j} = e^{-\beta||x_i - x_j||^2}$. This suggests that $diag(K) = \mathbf{1}_m$ since the distance between a point and itself is 0, and anything to the power of 0 is 1. The off diagonals take values within $(0, 1]$ due to the range of the exponential function. This also means that as we increase β the off diagonal components shrink towards 0, as any distance will be amplified by the large

β, meaning that $K \approx I$. Since $I^{-1} = I$ then $K^{-1} \approx I$ as β becomes larger. This means that we can approximate individual α_i as follows

$$\alpha_i \approx y_i + \sum_{j=1, j\neq i}^{m} K^{-1}_{i,j} y_j$$

This means that $\alpha_i = \gamma_i y_i$ where $\gamma_i \approx 1$ as β gets larger. Substituting this definition of α_i we have

$$y_{nn}(\gamma_{nn} y_{nn} K(\mathbf{x}_{nn}, t) + \sum_{i=1, i\neq nn}^{m} \gamma_i(-y_{nn}) K(\mathbf{x}_i, t)) > 0$$

We substitute in the equation for the gaussian kernel.

$$\gamma_{nn} e^{-\beta d^2_{nn}} - \sum_{i=1, i\neq nn}^{m} \gamma_i e^{-\beta(d_{nn}+\delta)^2} > 0$$

We can now factor out $e^{-\beta d^2_{nn}}$

$$e^{-\beta d^2_{nn}}(\gamma_{nn} - \sum_{i=1, i\neq nn}^{m} \gamma_i e^{-\beta(2d_{nn}\delta+\delta^2)}) > 0$$

From this equation we can see that $e^{-\beta d^2_{nn}} > 0$, so if the remaining term is larger than 0, then the inequality holds. The remaining term consists of a term $\gamma_{nn} \to 1$ as $\beta \to \infty$ and the subtraction of a number of terms dependent on an expontential of β. Therefore for large enough β, these terms approach 0, which we summarise with ϵ.

$$1 - \epsilon \approx 1 > 0$$

Therefore there exists some β where this inequality holds and we can make the gaussian kernel regression act as a 1-nn classifier.

Question 11

(a) Based on the question definition we have

$$\mathcal{E}_p(f) = \int_{\mathcal{X}\times\mathcal{Y}} \mathbf{1}_{f(x)\neq y} dp(x,y)$$

$$= \int_{f(\mathcal{X})\neq\mathcal{Y}} 1 dp(x,y) + \int_{f(\mathcal{X})=\mathcal{Y}} 0 dp(x,y)$$

$$= \int_{f(\mathcal{X})\neq\mathcal{Y}} 1 dp(x,y) \geq 0$$

This shows that $inf_{f:\mathcal{X}\to\mathcal{Y}}\mathcal{E}_p(f) \geq 0$. Now we show that for any $p_i((x,y))$ then $\mathcal{E}_{p_i}(f_i) = 0$

$$\mathcal{E}_{p_i}(f_i) = \int_{f_i(\mathcal{X})\neq\mathcal{Y}} 1 dp(x,y) + \int_{f_i(\mathcal{X})==\mathcal{Y}} 0 dp(x,y)$$

For every pair of $(x,y) \sim p_i$ the probability of drawing it is greater than 0 only if $f(x) = y$, hence in the equation above $|f_i(\mathcal{X}) = \mathcal{Y}| = 2n$ and $|f_i(\mathcal{X}) \neq \mathcal{Y}| = 0$. Therefore this leaves us with only the second integral.

$$\mathcal{E}_{p_i}(f_i) = \int_{f_i(\mathcal{X})=\mathcal{Y}} 0 dp(x,y) = 0$$

Since $\mathcal{E}_{p_i}(f_i)$ is a special instance of $\mathcal{E}_p(f)$ and we know $inf_{f:\mathcal{X}\to\mathcal{Y}}\mathcal{E}_p(f) \geq 0$. Then $\mathcal{E}_{p_i}(f_i) = inf_{f:\mathcal{X}\to\mathcal{Y}}\mathcal{E}_p(f) = 0$.

(b) Based on the definitions in the question, for a given $S \sim p_i$ we create labels for $x \in S$ using f_i such that $S^i = (x_1, f_i(x_1)), (x_2, f_i(x_2)), \ldots, (x_n, f(x_n))$. However, S is first drawn from C^n. On the LHS of the equation we have

$$\max_{i=1,\ldots,T} \mathbb{E}_{S\sim p_i^n}\mathcal{E}_{p_i}(A(S))$$

By optimising over T, we seek to find the labelling function which maximises the expected risk over all k possible datasets of size n which are drawn with equal probability and labelled with f_i. Which means that for any given $i \in T$

$$\mathbb{E}_{S\sim p_i^n}\mathcal{E}_{p_i}(A(S)) = \sum_{j=1}^{k} \frac{1}{k}\mathcal{E}_{p_i}(A(S_j^i)) = \frac{1}{k}\sum_{j=1}^{k}\mathcal{E}_{p_i}(A(S_j^i))$$

If we take the expectation over all possible labelling functions T which are drawn with equal probability rather than the maximum, then we have

$$\mathbb{E}_{f_i\sim\mathcal{Y}^C}\mathbb{E}_{S\sim p_i^n}\mathcal{E}_{p_i}(A(S)) = \sum_{i=1}^{T}\frac{1}{T}\sum_{j=1}^{k}\frac{1}{k}\mathcal{E}_{p_i}(A(S_j^i)) = \frac{1}{Tk}\sum_{i=1}^{T}\sum_{j=1}^{k}\mathcal{E}_{p_i}(A(S_j^i))$$

On the RHS we can also take the expectation over all possible k datasets with equal probability and obtain the following equation

$$\mathbb{E}_{S\sim C^n}\frac{1}{T}\sum_{i=1}^{T}\mathcal{E}_{p_i}(A(S_j^i)) = \sum_{j=1}^{k}\frac{1}{k}\frac{1}{T}\sum_{i=1}^{T}\mathcal{E}_{p_i}(A(S_j^i)) = \frac{1}{Tk}\sum_{i=1}^{T}\sum_{j=1}^{k}\mathcal{E}_{p_i}(A(S_j^i))$$

Since the LHS and RHS have the same terms over T and k, and by definition the minimum of a set of scalars is less than or equal to the average of them which is in turn less than or equal to the maximum of them, we have

$$\max_{i=1,\dots,T}\mathbb{E}_{S\sim p_i^n}\mathcal{E}_{p_i}(A(S)) \geq \frac{1}{Tk}\sum_{i=1}^{T}\sum_{j=1}^{k}\mathcal{E}(A(S_j^i)) \geq \min_{j=1,\dots,k}\frac{1}{T}\sum_{i=1}^{T}\mathcal{E}_{p_i}(A(S_j^i))$$

Which implies

$$\max_{i=1,\dots,T}\mathbb{E}_{S\sim p_i^n}\mathcal{E}_{p_i}(A(S)) \geq \min_{j=1,\dots,k}\frac{1}{T}\sum_{i=1}^{T}\mathcal{E}_{p_i}(A(S_j^i))$$

(c) On the LHS the expression seeks to evaluate the expected error of a classifier trained on a particular dataset S_j across the set of all possible labelling functions Y^C. Assuming that $A(S_j^i)$ learns to predict the elements of S_j^i with some error rate, the classifier has not been trained on any elements in ${S'}_j^i$ and hence will only be able to label these by chance. Therefore we can write the expected error of all classifiers learned on S_j over all possible labelling functions as the sum of 2 sources of error.

$$\frac{1}{T}\sum_{i=1}^{T}[\mathcal{E}_{p_i}(A(S_j^i)(S_j^i)) + \mathcal{E}_{p_i}(A(S_j^i)({S'}_j^i))]$$

$$= \frac{1}{T}\sum_{i=1}^{T}[\frac{1}{2n-p}\sum_{r=1}^{2n-p}\mathbf{1}_{A(S_j^i)(x_r)\neq f_i(x_r)} + \frac{1}{p}\sum_{r=1}^{p}\mathbf{1}_{A(S_j^i)(v_r)\neq f_i(v_r)}]$$

$$= \frac{1}{T}\sum_{i=1}^{T}\frac{1}{2n-p}\sum_{r=1}^{2n-p}\mathbf{1}_{A(S_j^i)(x_r)\neq f_i(x_r)} + \frac{1}{T}\sum_{i=1}^{T}\frac{1}{p}\sum_{r=1}^{p}\mathbf{1}_{A(S_j^i)(v_r)\neq f_i(v_r)}$$

On the RHS, if we take the expectation over all $v_r \in {S'}_j^i$ instead of the minimum, then we have

$$\frac{1}{2}\sum_{r=1}^{p}\frac{1}{p}\frac{1}{T}\sum_{i=1}^{T}\mathbf{1}_{A(S_j^i)(v_r)\neq f_i(v_r)}$$

Which is the same as half of the 2nd error term on the LHS. Since both of these are expected risk terms which we previously showed are non-negative, then we have shown that the LHS is larger than the RHS, since the minimum is less than the expectation for any given set of scalars.

(d) The set of functions $f_i \in Y^C$ was computed by taking all possible combinations of labelling functions of size $2n$. The rationale for this being that for each $x \in C$ there are 2 possible classifications, so we can have $2 \times 2 \times \cdots \times 2$ up to 2^{2n} unique functions. If we fix a particular $v_r = 1$ or $v_r = -1$ then the number of combinations for the remaining parts of the string can be computed in the same way with $2 \times 2 \times \cdots \times 2$ up to $2n-1$ times, giving us $2^{n-1} = \frac{T}{2}$ combinations. Since the set where $v_r = 1$ and $v_r = -1$ is of the same size, they are in effect the same set, and therefore we can pair up each element in both sets.

(e) Markov's inequality is given as

$$Pr(Z \geq a) \leq \frac{\mathbb{E}(Z)}{a}$$

Therefore

$$Pr(Z > 1-a) = 1 - Pr(Z \leq 1-a) \leq \frac{\mathbb{E}(Z)}{1-a}$$

If we rearrange the 2nd expression we can have

$$Pr(Z \leq 1-a) = Pr(a \leq 1-Z) = Pr(1-Z \geq a) \leq \frac{\mathbb{E}(1-Z)}{a}$$

$$Pr(Z > 1-a) = 1 - \frac{\mathbb{E}(1-Z)}{a} = \frac{a - \mathbb{E}(1) + \mathbb{E}(Z)}{a} = \frac{\mathbb{E}(Z) - (1-a)}{a}$$

(f) We seek to prove

$$\mathbb{P}_{S\sim p^n}[\mathcal{E}_p(A(S)) > \frac{1}{8}] \geq \frac{1}{7}$$

From part c, we have

$$\frac{1}{T}\sum_{i=1}^{T} \mathcal{E}_{p_i}(A(S_j^i)) \geq \frac{1}{2} \min_{r=1,\ldots,p} \frac{1}{T} \mathbf{1}_{A(S_j^i)(v_r) \neq f_i(v_r)}$$

By substituting part d into this equation we obtain

$$\frac{1}{T}\sum_{i=1}^{T} \mathcal{E}_{p_i}(A(S_j^i)) \geq \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

To obtain $\mathbb{E}_{S\sim p^n}\mathcal{E}_p(A(S))$ we need to take the expectation over all k possible datasets of size n so we have

$$\mathbb{E}_{S\sim p^n}\mathcal{E}_p(A(S)) = \frac{1}{kT}\sum_{j=1}^{k}\sum_{i=1}^{T} \mathcal{E}_{p_i}(A(S_j^i)) \geq \frac{1}{k}\sum_{j=1}^{k}\frac{1}{4} = \frac{1}{4}$$

We can then substitute in this lower bound into the equation from part e

$$\mathbb{P}_{S\sim p^n}(\mathcal{E}_p(A(S)) > 1 - \frac{7}{8}) \geq \frac{\mathbb{E}_{S\sim p^n}\mathcal{E}_p(A(S)) - \frac{1}{8}}{\frac{7}{8}} \frac{\frac{1}{4} - \frac{1}{8}}{\frac{7}{8}} = \frac{\frac{1}{8}}{\frac{7}{8}} = \frac{1}{7}$$

(g i) Let $\mathcal{X}$ be an infinite set which is mapped to 2 classes in $\mathcal{Y}$ by some distribution $p_{\mathcal{X}\times\mathcal{Y}}$. For any subset C of $\mathcal{X}$ of size $2n$, and a learning algorithm A which learns a function $f : \mathcal{X} \to \mathcal{Y}$, if A is trained on n or less samples from C, there exists a function f such that $\mathcal{E}_{p^n}(f) = 0$ which is the minimum error. However, the average error of the classifier learned on n or less samples from C when evaluated on all possible datasets of size n from C is at least $\frac{1}{4}$. Furthermore, the probability that such a classifier has an error larger than $\frac{1}{8}$ is at least $\frac{1}{7}$ or greater.

(g ii) It implies that the space of functions $\mathcal{Y}^{\mathcal{X}}$ is not learnable. Based on part a, we know that the $inf_{f:\mathcal{X}\to\mathcal{Y}}\mathcal{E}(A(S)) = 0$. So by setting $\epsilon = \frac{1}{8}$ we can fill in the equation as

$$\mathbb{P}_{S\sim p^n}(\mathcal{E}_p(A(S)) - 0 \leq \frac{1}{8}) \geq 1 - \delta$$

This expression is the complement of the expression in part f.

$$1 - \mathbb{P}_{S\sim p^n}(\mathcal{E}_p(A(S)) > \frac{1}{8}) = \mathbb{P}_{S\sim p^n}(\mathcal{E}_p(A(S)) \leq \frac{1}{8})$$

Since we previously saw that

$$\mathbb{P}_{S\sim p^n}(\mathcal{E}_p(A(S)) > \frac{1}{8}) \geq \frac{1}{7}$$

Then

$$\mathbb{P}_{S\sim p^n}(\mathcal{E}_p(A(S)) \leq \frac{1}{8}) \leq \frac{6}{7}$$

So if we set $0 \leq \delta < \frac{1}{7}$, then we obtain a contradiction, implying that the space of functions is not learnable.

(g iii) The implication is that for binary classification, unless one has knowledge of the data generation process beyond the information found within the training set, no learning algorithm is able to learn a classifier that generalises beyond the training set. As such when building machine learning models for binary classification, ideally have some information about the data generation process and can use that to choose which functions are more likely to represent the underlying process and hence to search over them rather than all possible functions.