Question 1

1). We can use a categorical/multinoulli distribution to describe this scenario.

We will have six parameters:

$\mu_1 \quad \mu_2 \quad \mu_3 \quad \mu_4 \quad \mu_5 \quad \mu_6$

the probabilities for side 1, 2, 3, 4, 5, 6 comes up.

2). If we have a fair dice then

$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \frac{1}{6}$

3). if the die always rolls two then

$\mu_2 = 1, \quad \mu_1 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = 0.$

4) the domain of parameters is $[0,1]$

$\mu_1 \in [0,1] \quad \mu_2 \in [0,1] \quad \mu_3 \in [0,1]$

$\mu_4 \in [0,1] \quad \mu_5 \in [0,1] \quad \mu_6 \in [0,1]$

Question 2.

$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N} \alpha_n \{t_n - w^T\phi(x_n)\}^2$$

$$\frac{\partial}{\partial w} E_D(w) = \frac{1}{2}\sum_{n=1}^{N} 2(t_n - w^T\phi(x_n))(-\phi(x_n)) \cdot \alpha_n$$

$$= \sum_{n=1}^{N} (t_n - w^T\phi(x_n))(-\phi(x_n)) \cdot \alpha_n$$

$$\nabla E_D(w) = \sum_{n=1}^{N} (t_n\alpha_n - w^T\alpha_n\phi(x_n))(-\phi(x_n))^T$$

$$\phi(x_n) = \begin{bmatrix} \phi_0(x_n) \\ \phi_1(x_n) \\ \vdots \\ \phi_{M-1}(x_n) \end{bmatrix} \qquad 0^T = [0, 0, 0 \dots 0]$$

$$\nabla E_D(w) = \left[\frac{\partial}{\partial w_0}\ln(\cdot), \frac{\partial}{\partial w_1}\ln(\cdot) \dots \frac{\partial}{\partial w_n}\ln(\cdot)\right]$$

Set the gradient to 0.

$$0^T = \nabla E_D(w) = \sum_{n=1}^{N} (t_n\alpha_n - w^T\alpha_n\phi(x_n))(-\phi(x_n))^T$$

$$0^T = \sum_{n=1}^{N} -t_n\alpha_n\phi(x_n) + w^T \underbrace{\sum_{n=1}^{N} \alpha_n\phi(x_n)\phi(x_n)^T}$$

$$\boxed{0^T = t^T\alpha^T\Phi - w^T\Phi^T\alpha^T\Phi}$$

Why? Using this part as an example:

$$\underset{M\times 1}{\phi(x_n)}\ \underset{1\times M}{\phi(x_n)^T} = \begin{bmatrix} \phi_0(x_n) \\ \phi_1(x_n) \\ \vdots \\ \phi_{M-1}(x_n) \end{bmatrix} [\phi_0(x_n)\ \phi_1(x_n) \dots \phi_{M-1}(x_n)]$$

$\phi_N(x_n)$

$$= \begin{pmatrix} \phi_0(x_n)\phi_0(x_n) & \phi_0(x_n)\phi_1(x_n) \dots \phi_0(x_n)\phi_N(x_n) \\ \phi_1(x_n)\phi_0(x_n) & \phi_1(x_n)\phi_1(x_n) \dots \\ \vdots & \vdots \end{pmatrix}$$

Treat α_n as a scalar then $\alpha_n \phi(x_n) \phi(x_n)^T$ become

$$\begin{pmatrix} \alpha_n \phi_0(x_n) \phi_0(x_n) & \cdots \\ \alpha_n \phi_1(x_n) \phi_0(x_n) & \cdots \\ \vdots & \end{pmatrix}$$

$\sum_{n=1}^{N} \alpha_n \phi(x_n) \phi(x_n)^T$?

$$\boxed{\alpha_1 \phi_0(x_1) \phi_0(x_1) + \alpha_2 \phi_0(x_2) \phi_0(x_2) + \cdots + \alpha_N \phi_0(x_N) \phi_0(x_N)}$$

$$\boxed{\alpha_1 \phi_1(x_1) \phi_0(x_1) + \alpha_2 \phi_1(x_2) \phi_0(x_2) + \cdots + \alpha_N \phi_1(x_N) \phi_0(x_N)}$$

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \cdots & \phi_{M-1}(x_2) \\ \vdots & & & \\ \phi_0(x_N) & \phi_1(x_N) & \cdots & \phi_{M-1}(x_N) \end{pmatrix} \quad \underline{N \times M}$$

$$\Phi^T = \begin{pmatrix} \phi_0(x_1) & \phi_0(x_2) & \cdots & \phi_0(x_N) \\ \phi_1(x_1) & \phi_1(x_2) & \cdots & \phi_1(x_N) \\ \vdots & \vdots & & \vdots \\ \phi_{M-1}(x_1) & \phi_{M-1}(x_2) & \cdots & \phi_{M-1}(x_N) \end{pmatrix} \quad M \times N.$$

$$\alpha = \begin{pmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 & 0 & \cdots & 0 \\ 0 & 0 & \alpha_3 & \cdots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & & \alpha_N \end{pmatrix} \quad N \times N$$

Ïhý uú

$$\underset{M \times N}{\Phi^T \alpha} = \begin{pmatrix} \alpha_1 \phi_0(x_1) & \alpha_2 \phi_0(x_2) & \cdots & \alpha_N \phi_0(x_N) \\ \alpha_1 \phi_1(x_1) & \alpha_2 \phi_1(x_2) & \cdots & \alpha_N \phi_1(x_N) \\ \vdots & \vdots & & \vdots \\ \alpha_1 \phi_{M-1}(x_1) & \alpha_2 \phi_{M-1}(x_2) & \cdots & \alpha_N \phi_{M-1}(x_N) \end{pmatrix}$$

$$\underset{M \times M}{\Phi^T \alpha \Phi} = \begin{pmatrix} \; \\ \; \\ \; \end{pmatrix}$$

$$\rightarrow \boxed{\alpha_1 \phi_0(x_1) \phi_0(x_1) + \alpha_2 \phi_0(x_2) \phi_0(x_2) + \cdots}$$

$$\rightarrow \boxed{\alpha_1 \phi_1(x_1) \phi_0(x) + \alpha_2 \phi_1(x_2) \phi_0(x_2) + \cdots}$$

Same as $\begin{pmatrix} \alpha_n \phi_0(x_n) \phi_0(x_n) & \cdots \\ \alpha_n \phi_1(x_n) \phi_0(x_n) & \cdots \\ \vdots & \cdots \end{pmatrix}$

$$\left\{ \begin{aligned} (ABC)^T &= C^T B^T A^T \\ (AB)^T &= B^T A^T \end{aligned} \right.$$

Therefore. $0^T = t^T \alpha^T \Phi - w^T \Phi^T \alpha^T \Phi$

$$w^T \Phi^T \alpha^T \Phi = t^T \alpha^T \Phi$$

$$(\Phi^T \alpha \Phi) w = \Phi^T \alpha t$$

$$w = (\Phi^T \alpha \Phi)^{-1} (\Phi^T \alpha t).$$

$$(\Phi^T \alpha t)^T = t^T \alpha^T \Phi$$

$$((\Phi^T \alpha \Phi) w)^T = w^T \Phi^T \alpha^T \Phi$$

Question 3

$E(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2$

$\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2} \|w\|^2$

RMS: $E_{RMS} = \sqrt{2E(w^*)/N}$

1) No, The training set and validation set are both randomly distributed data, from the dataset. There is no guarantees on the relationship between training error and validation error.

2) If the model is overfit the validation error is probably higher than the training error

Good fit: Validation error low, slightly higher than the training error.

Unknown fit: Validation error low, training error high

Under fit: Validation error and training error both high

Generally speaking, training error will almost always underestimate the validation error. But it is possible for the validation error to be less than the training.

2) Yes. Degree 10 polynomial contains Degree 9 polynomial. The unregularized regression gives us the optimal solution which means Degree 10 polynomial mostly fits the data better. In the worst case, the training error

for them two are equal.

But if we change training error to testing error for this question then the answer shuv be "No".

3) No.

In most cases the testing error for regularized regression is lower than unregularized regression since the degree=20 is very high and is highly likely to cause overfitting.

But this is not garuanteed. If we got a weak model then the regularized will underfitting slash the predictive power even more and make the testing error larger compared with the unregularized one.

Ïhý uú

Question 4.

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - w^T\phi(x_n)\}^2 + \frac{1}{2}\sum_{j\in J_1}\lambda_j|w_j|$$

$$+\frac{1}{2}\sum_{j\in J_2}\lambda_j|w_j|^2.$$

Here, $\lambda_{j\in J_2}$ and $\lambda_{j\in J_1}$ are two subsets/subvector of $\lambda_{n\in N}$,

$$\frac{\partial E(w)}{\partial w} = \frac{1}{2}\sum_{n=1}^{N}2(t_n - w^T\phi(x_n))(-\phi(x_n)) + \frac{1}{2}\sum_{j\in J_1}\frac{w_j}{|w_j|}\lambda_j$$

$$+\frac{1}{2}\sum_{j\in J_2}\lambda_j 2w_j$$

$$= \sum_{n=1}^{N}(t_n - w^T\phi(x))(-\phi(x_n)) + \frac{1}{2}\sum_{j\in J_1}\frac{w_j}{|w_j|}\lambda_j$$

$$+\sum_{j\in J_2}\lambda_j w_j.$$

Define matrices;

$$\varphi = \begin{cases}\frac{w_j}{|w_j|}\lambda_j & \text{for } j\in J_1\\ 0 & \text{otherwise}\end{cases}$$

$$r = \begin{cases}\lambda_j & \text{for } j\in J_2\\ 0 & \text{otherwise}\end{cases}$$

I = Identity matrix = $[1, 1, 1, 1 \ldots]^T$.

$$\nabla E(w) = -t^T\Phi + w^T\Phi^T\Phi + \frac{1}{2}\varphi I^T + r.w^T$$

Here Φ is the same design matrix we declared in clas

Ïhý uú

Question 5.

5.1) Niger. 313.7/1000 = 31.37%

Sierra Leone. 185.3/1000 = 18.53%

"na_values = "_" will set the missing features to NA values.

Then we will use nanmean() to find the average value for each column/feature,

np.where (np.isnan) will find the coordinate/indices for the NA values.

and we assign the average value to the missing parts according to their indices we found.

NB: np.where(np.isnan) will return the row and column indices, But np.take(mean_vals, inds[1]) → we only need the column indices since the average is a $\left\{\begin{matrix} 1 \times 40 \\ 1 \times N \end{matrix}\right.$ matrix/vector.

Ïhý uú