Assignment 3

Tuesday, 28 November 2023 08:07

Prathvish Mithare - 7024149

Subhad Kothare Dutta - 7024082

Exercise 3.1

1) For M to be symmetric matrix it must satisfy $M = M^T$

A real symmetric matrix can be decomposed into two components using they real valued symmetric and skew symmetric

$$M = \frac{M + M^T}{2}$$

→ Makes component of symmetric by S.

→ Diagonal matrix with symmetric by D.

2) A singular matrix is a matrix with determinant 0. Thus the matrix is not invertible.

$$|M| = 2(2-1) - 1(-1-0) + 0$$
$$= 2 - 1 = 0$$

Thus M is singular matrix

3) M is a singular matrix thus it at least has one zero eigenvalue

4) $M = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$

Eigenvalue: $|M\lambda I^{-}| = 0$

$$\begin{bmatrix} 2 & -1 & 1 \\ -1 & 6 & -3 \\ 1 & -3 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} = 0$$

$$\begin{bmatrix} 2-\lambda & -1 & 1 \\ -1 & 6-\lambda & -3 \\ 1 & -3 & 3-\lambda \end{bmatrix} = 0$$

$$(2-\lambda)[(6-\lambda)(3-\lambda)-9] + [(-1)(3-\lambda)] + (1)(-3+6-\lambda) = 0$$

Solving for λ we get

$$\lambda = 0, 7, 3$$

Using $\lambda = 0$

we get eigenvector

$$(M - \lambda I)\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\rightarrow v = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ normalizing } \Rightarrow \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}$$

using $\lambda = 7$

we get eigenvector

$$(M - \lambda I)\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\rightarrow v = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \text{ normalizing } \Rightarrow \begin{bmatrix} 1/\sqrt{6} \\ -2/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix}$$

using $\lambda = 3$

we get eigenvector

$$(M - \lambda I)\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\rightarrow v = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \text{ normalizing } \Rightarrow \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{bmatrix}$$

Final Matrix Φ we have

$$M = \Phi \Lambda \Phi^T$$

$$\Phi = \begin{bmatrix} 1/\sqrt{3} & -1/\sqrt{6} & 1/\sqrt{2} \\ 1/\sqrt{3} & 0 & 1/\sqrt{2} \\ 1/\sqrt{3} & 1/\sqrt{6} & -1/\sqrt{2} \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\therefore M = \begin{bmatrix} 1/\sqrt{3} & -1/\sqrt{6} & 1/\sqrt{2} \\ 1/\sqrt{3} & 0 & 1/\sqrt{2} \\ 1/\sqrt{3} & 1/\sqrt{6} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ -1/\sqrt{6} & 0 & 1/\sqrt{6} \\ 1/\sqrt{2} & 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

Exercise 3.2

5)

	Actual		
Predicted	Yes	No	
	Yes	TP	FP
	No	FN	TN

Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy tells us how frequently an classifier was correct predicts. Accuracy is 1 if the positive or correct prediction to total number of predictions.

Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

A precision of a model is its ability only the relevant data. In this case out of the positive predictions, how many are positive. It is useful when to allow false positives. For top, that would result in high precision.

Classifying most examples as false or not spam.

Recall $\frac{TP}{TP + FN}$

The ability of a model to find the relevant cases. It is useful when to allow more false negatives to stop that would have large impact.

Used in medical field for cancer detection etc.

It tries to account the both precision and recall. It does ignore the balance of both precision and recall.

$$F_1 = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}$$

7) $y = ax^2$

$$\text{covariance}(x,y) = \frac{\text{Cov}(x,y)}{\sigma_x \sigma_y}$$

$$\text{cov}(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \quad \text{—— (1)}$$

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad \text{—— (2)}$$

$$\sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \quad \text{—— (3)}$$

$$\bar{x} = \bar{y} = 0$$

Residual square $R = (y - ax)^2$

differentiate with a

$$\frac{\partial R}{\partial a} = \frac{\partial}{\partial a}(y - ax)^2$$

$$= 2(y - ax)(-x)$$

Equate it to zero

$$0 = -2(y - ax)(x)$$

$$yx - ax^2 = 0$$

$$ax^2 = yx$$

$$a = \frac{yx}{x^2}$$

$$\therefore a = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i x_i}{x_i^2} \quad \text{[Considering for all data } i = n\text{]}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \bar{y})(x_i - \bar{x})}{(x_i - \bar{x})^2}$$

$$= \frac{1}{n}\frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\text{cov}(x,y)}{\sigma_x^2} \quad \text{[from equation (1) \& (2)]}$$

$$\text{For } \rho = \sqrt{\frac{\text{cov}(x,y)}{\sigma_x^2}\frac{\text{cov}(x,y)}{\sigma_y^2}}$$

$$= \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}$$

8) $E[(y - \hat{y}_{mse})^2] = E[(y - \hat{y} + \hat{y} - \hat{y}_{mse})^2]$

Adding (in) the given equation

$$E[(y - \hat{y})^2 + (\hat{y} - \hat{y}_{mse})^2 + 2(y - \hat{y})(\hat{y} - \hat{y}_{mse})]$$

$$= E[y^2 + \hat{y}^2 - 2y\hat{y}] + E[\hat{y}^2 + \hat{y}_{mse}^2 - 2\hat{y}\hat{y}_{mse}] + 2E[y\hat{y} - y\hat{y}_{mse} - \hat{y}^2 + \hat{y}\hat{y}_{mse}]$$

$$= E[(y - \hat{y})^2] + E[(\hat{y} - \hat{y}_{mse})^2] + 2E[y\hat{y} - y\hat{y}_{mse} - \hat{y}^2 + \hat{y}\hat{y}_{mse}]$$

$$= E[(y - \hat{y})^2] + E[(\hat{y} - \hat{y}_{mse})^2] \quad \text{[since } E[\hat{y}] = 0, E[y\hat{y}_{mse}] = E[\hat{y}\hat{y}_{mse}] = 0\text{]}$$

$$\text{since } E[\hat{y}^2] = \sigma^2 \text{ (variance)}$$

$$= E[(y - \hat{y})^2] + E[(\hat{y} - \hat{y}_{mse})^2] + \sigma^2$$

$$\therefore = E[(y - \hat{y})^2] + \sigma^2 \quad \text{[} \because E[(\hat{y} - \hat{y}_{mse})^2] = 0\text{]}$$

$$= \text{SMSE}$$

9) $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

$y_1 = 10 + 0.5x_1 + 0.2x_2 + 0.3x_3 = 10.5$

(a) $y_{mse} = 10 + 0.5x_1 + 0.2x_2 + 0.3x_3$

$y_{mse} = 10 + 0.5(-1) + 0.2(2) + 0.3(3)$

$y_{mse} = 10 + 0.5 + 0.4 + 0.9 = 12.8$

$\Delta y = y_{mse} - y_1 = 12.8 - 10.5 = 2.3$ which means there is bias.

→ The predicted output is 2.3 $\frac{\text{unit}}{\text{bias}}$ which is $\frac{\text{output}}{\text{bias}}$.

The unit change is caused $\frac{\text{input change}}{\text{bias}}$.

(b) $y_{mse} = 10 + 0.5x_1 + 0.2(2.5) + 0.3x_3$

$\Delta y = 0.5\Delta(x_1) + 0.2\Delta(x_2) = 0.5\left(\frac{\Delta y}{\Delta x_1}\right) + 0.2\left(\frac{\Delta y}{\Delta x_2}\right)$

→ The predicted output is $0.5\left(\frac{\Delta y(1.5)}{\text{bias}}\right) + 0.2\left(\frac{\Delta y(0)}{\text{bias}}\right)$

The unit change is 1 since $\frac{\Delta y(1.5)}{\text{bias}}$ unit change.

(c) $y_{mse} = 10 + 0.5x_1 + 0.2(2.5) + 0.3x_3$

$\Delta y = 0.5\Delta(x_1) + 0.2\Delta(x_2) = 0.5\left(\frac{\Delta y}{\Delta x_1}\right) + 0.2\left(\frac{\Delta y}{\Delta x_2}\right)$

→ The predicted output is $0.5\left(\frac{\Delta y(1.5)}{\text{bias}}\right) + 0.2\left(\frac{\Delta y(0)}{\text{bias}}\right)$

The unit change is 1 since $\frac{\Delta y(1.5)}{\text{bias}}$ unit change.

(d) β_3 gives the effect of new factor needed.

Exercise 3.3

(a) Cross validation is used method of hold-out method where the amount of data is limited. It checks the models performance on unseen data when the size of the dataset is limited. Cross validation yields out the models ability to learn from different sections of the data and to performance on the unseen set.

The size parameter is which set controls the k-fold cross validation. (k=5)

→ When working with limited data?

→ When the data is limited we would use cross-validation method on train set + better idea about the models performance on different sub-sets of data.

→ Hyperparameter fine tuning?

→ By tuning the models on multiple subsets of the dataset the model learns better to generalize to unseen samples.

(b)

Now we have our procedure to tune with these possible options using k fold cross validation. We will involve 10 students of Training Data the model.

The steps involved are as follows

Step 1: Split the dataset into different subsets according to this step where we find out optimal number of folds to be removed.

Step 2: Train the model on values for the hyperparameters (i.e order of polynomial)

Step 3: Train data is then split into k subsets where here k=5

Step 4: For each of the subsets

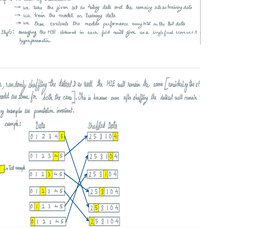
→ In this the given set is taken data and the remaining sets are training data

→ Train the model on training data

→ Now, evaluate the models performance using MSE on the test data

Step 5: Averaging the MSE obtained in each fold will give us a single final score for hyperparameters.

(c)

After randomly shuffling the dataset is to hold the MSE will remain the same because the model has learned from the data. This is because even after shuffling the dataset will remain training samples are population, unbiased.

example: