written1

September 3, 2021

1 STP598 Machine Learning & Deep Learning

1.1 Written Assignment 1

1.1.1 Due 11:59pm Sunday Sept. 12, 2021 on Canvas

1.1.2 name, id

1.2 Question 1

Let C_1, C_2, C_3 be independent events with probabilities $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, respectively.

1. Compute $P(C_1 \cup C_2 \cup C_3)$.
2. Compute $P(C_1^c \cup C_2^c \cup C_3^c)$.

1.3 Question 2

In class we talked about Monty Hall problem (refer to page 34 of lecture 1).

1. Now if there are 4 doors, you pick door 1 and Monty opens door 3 and door 4, will the conclusion change if you switch your choice to door 2? Compute the relative probabilities.

2. Again there are 4 doors, you pick door 1 and Monty only opens 4. Should you change your choice? Write down your analysis.

1.4 Question 3

In the linear regression

$$Y = X\beta + \epsilon, \quad \epsilon \overset{iid}{\sim} (0, \sigma^2) \tag{1}$$

Given data $\{y_i, \mathbf{x}_i\}_{i=1}^n$, assume $n > p$ with p being the number of features. We can have the following estimator for σ^2 (Refer to page 13 of lecture 2 for relevant symbols):

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - (p+1)} = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|^2}{n - (p+1)} = \frac{\mathbf{y}^T(\mathbf{I} - \mathbf{H})\mathbf{y}}{n - (p+1)} \tag{2}$$

1. We know $\mathbb{E}[\mathbf{v}^T \Lambda \mathbf{v}] = \mu^T \Lambda \mu + \mathrm{tr}[\Lambda \Sigma]$ for $\mathbb{E}[\mathbf{v}] = \mu$ and $\mathrm{Cov}[\mathbf{v}] = \Sigma$. Can you prove that $\hat{\sigma}^2$ is an unbiased estimator of σ^2, i.e. $\mathbb{E}[\hat{\sigma}^2] = \sigma^2$?
2. (bonus) Can you further show that $\hat{\sigma}^2/\sigma^2 \sim \chi^2(n - (p+1))$? What condition do you need?

1.5 Question 4

Consider diabetes data in scikit-learn package. Load it as follows.

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets

# Load the diabetes dataset
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True, as_frame=True)

# print the first 5 records
import pandas as pd
diabetes = pd.concat([diabetes_y, diabetes_X],1)
diabetes.head(5)
```

```
<ipython-input-1-82b1f379d4d5>:10: FutureWarning: In a future version of pandas
all arguments of concat except for the argument 'objs' will be keyword-only
  diabetes = pd.concat([diabetes_y, diabetes_X],1)
```

[1]:
```
   target       age       sex       bmi        bp        s1        s2  \
0   151.0  0.038076  0.050680  0.061696  0.021872 -0.044223 -0.034821
1    75.0 -0.001882 -0.044642 -0.051474 -0.026328 -0.008449 -0.019163
2   141.0  0.085299  0.050680  0.044451 -0.005671 -0.045599 -0.034194
3   206.0 -0.089063 -0.044642 -0.011595 -0.036656  0.012191  0.024991
4   135.0  0.005383 -0.044642 -0.036385  0.021872  0.003935  0.015596

         s3        s4        s5        s6
0 -0.043401 -0.002592  0.019908 -0.017646
1  0.074412 -0.039493 -0.068330 -0.092204
2 -0.032356 -0.002592  0.002864 -0.025930
3 -0.036038  0.034309  0.022692 -0.009362
4  0.008142 -0.002592 -0.031991 -0.046641
```

1. Fit linear regression, ridge regression and lasso respectively. The panelty parameters can be determined using cross-validation (`sklearn.linear_model.RidgeCV`, `sklearn.linear_model.LassoCV`). Plot three regression lines $y \sim bmi$ on the same graph.

2. Plot lasso coefficients as a function of the regularization. Refer to `plot_ridge_path.ipynb`.