# ESC21SUMMER_HW3_woohyunchoi

August 5, 2021

```
[1]: import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression, Ridge, Lasso
     import matplotlib.pyplot as plt
     from sklearn.metrics import mean_squared_error
```

```
[16]: import warnings
      warnings.filterwarnings('ignore')
```

```
[3]: # Data Import
     import ssl
     import pandas as pd
     ssl._create_default_https_context = ssl._create_unverified_context #Github    ␣
      ↪            .                    !
     data = pd.read_csv('https://github.com/YonseiESC/ESC-21SUMMER/blob/main/week3/
      ↪HW_data/data.csv?raw=True')
```

```
[4]: data.head()
     #Age:
     #Experience:
     #Income:
     #Family:
     #CCAvg:
```

```
[4]:    Age  Experience  Income  Family  CCAvg
     0   25           1      49       4    1.6
     1   45          19      34       3    1.5
     2   39          15      11       1    1.0
     3   35           9     100       1    2.7
     4   35           8      45       4    1.0
```

```
[5]: #    ( )
     data.isnull().sum()
```

```
[5]: Age           0
     Experience    0
```

```
Income        0
Family        0
CCAvg         0
dtype: int64
```

[6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Age         2500 non-null   int64
 1   Experience  2500 non-null   int64
 2   Income      2500 non-null   int64
 3   Family      2500 non-null   int64
 4   CCAvg       2500 non-null   float64
dtypes: float64(1), int64(4)
memory usage: 97.8 KB
```

[7]:
```
y = data['Income']
X = data.drop(['Income'], axis = 1)
x_train, x_test, y_train, y_test = train_test_split(X, y, train_size = 0.7,␣
 ↪random_state = 1000)
```

## 0.1 Linear Regression

[8]:
```
reg = LinearRegression()
results1 = reg.fit(x_train, y_train)
```

[9]:
```
reg.coef_
```

[9]:
```
array([-3.07793956,  2.89401562, -3.37220023, 16.09065086])
```

## 0.2 Ridge Regression

[10]:
```
rreg = Ridge(alpha = 0) # alpha = Lambda
rreg.fit(x_train, y_train)
```

[10]:
```
Ridge(alpha=0)
```

[11]:
```
rreg.coef_
```

[11]:
```
array([-3.07793956,  2.89401562, -3.37220023, 16.09065086])
```

[12]:
```
alpha = np.logspace(-3,3,7)
alpha
```

[12]:
```
array([1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e+03])
```

## 0.3 Lasso Regression

[17]:
```
lreg = Lasso(alpha = 0 ) # alpha = Lambda
lreg.fit(x_train, y_train)
```

[17]:
```
Lasso(alpha=0)
```

[18]:
```
lreg.coef_
```

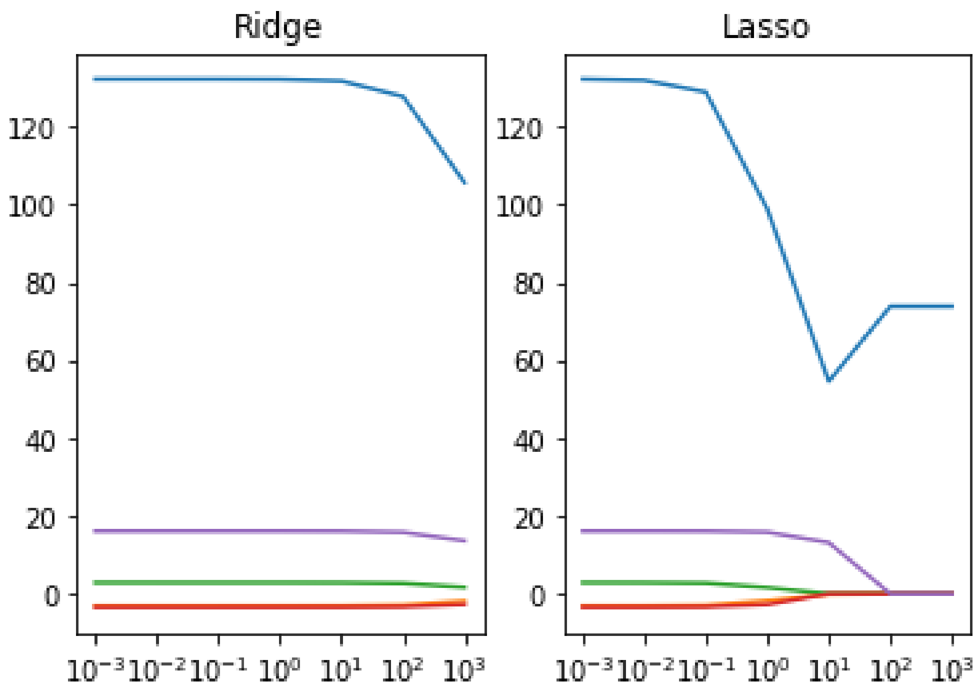
[18]:
```
array([-3.07790231,  2.8939786 , -3.37220244, 16.09065156])
```

[19]:
```
df = []
acc_table = []

for i, a in enumerate(alpha):
        lreg = Lasso(alpha=a).fit(x_train, y_train)
        df.append(pd.Series(np.hstack([lreg.intercept_, lreg.coef_])))
        pred_y = lreg.predict(x_test)

df_lasso = pd.DataFrame(df,index = alpha).T
df_lasso
```

[19]:
```
     0.001       0.010       0.100      1.000      10.000     100.000   \
0  132.261976  131.960877  128.945930  98.937749  54.569493    73.876
1   -3.076625   -3.065044   -2.949074  -1.794975  -0.134206    -0.000
2    2.892703    2.881139    2.765340   1.612913  -0.000000    -0.000
3   -3.371595   -3.366136   -3.311548  -2.765340  -0.000000    -0.000
4   16.090400   16.088142   16.065558  15.839618  13.184919     0.000

   1000.000
0    73.876
1    -0.000
2    -0.000
3    -0.000
4     0.000
```

[20]:
```
import matplotlib.pyplot as plt

ax1 = plt.subplot(121)
plt.semilogx(df_ridge.T)
plt.xticks(alpha)
plt.title("Ridge")
```

```
ax2 = plt.subplot(122)
plt.semilogx(df_lasso.T)
plt.xticks(alpha)
plt.title("Lasso")

plt.show()
```



**Exercise. 3.29**

Suppose we fit a ridge regression with a given shrinkage parameter $\lambda \in \mathbb{R}^+$ on a single variable $x_1$. (Notice that $x_1$ is a $N \times 1$ vector.)

1. (Essential) Show that the coefficient must be $\frac{X^T y}{X^T X + \lambda}$ where $X = x_1$.
2. (Essential) We now include an exact copy $x_2 = x_1$, so our new design matrix would be $X = [x_1 | x_2]$. Using this matrix, re-fit our ridge regression. Show that both coefficients are identical, and derive their value.

1. $L = (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta$

$$\frac{\partial L}{\partial \beta} = -(X^T y - X^T X \beta)^T - (y - X\beta)^T X + 2\lambda\beta^T$$

$$= -2y^T X + 2\beta^T X^T X + 2\lambda\beta^T$$

$$\frac{\partial L}{\partial \beta} = 0 \Rightarrow \beta^T (X^T X + \lambda I) = y^T X$$

In this case, $X = x_1$. $I = 1$. $\Rightarrow \beta^T = (X^T X + \lambda)^{-1} y^T X$

$$\therefore \beta = \frac{X^T y}{X^T X + \lambda}$$

2. $$(X^T X + \lambda I)^{-1} X^T y = \begin{bmatrix} x_1^2 + \lambda & x_1 x_2 \\ x_1 x_2 & x_2^2 + \lambda \end{bmatrix}^{-1} \begin{bmatrix} x_1^T y \\ x_2^T y \end{bmatrix}) \circledcirc$$

symmetric

$$= \frac{1}{(x^2 + \lambda)^2 - x^4} \begin{bmatrix} x^2 + \lambda & -x^2 \\ -x^2 & x^2 + \lambda \end{bmatrix} \begin{bmatrix} xy \\ xy \end{bmatrix}$$

$$= \frac{1}{2x^2\lambda + \lambda^2} \begin{bmatrix} \lambda x y \\ \lambda x y \end{bmatrix}$$

$$\therefore \beta_1 = \beta_2 = \frac{xy}{2x^2 + \lambda}$$