hw1

September 30, 2024

1 HW 1 - Ishaan Sathaye

1.1 Section A: Theory

(a) $Y = \beta_1 X + \beta_0 + \epsilon$

(b) $Y = \beta * X + 5000 + \epsilon$

(c) $Y = \beta * X^2 + \epsilon$

(d) $Y = log(\beta * X) + \epsilon$

1. Derive the Least Squares Estimator(s) for each model.

(a)

- Residual: $\epsilon = Y - \beta_1 X - \beta_0$
- Minimize Sum of Squared Residual (loss): $l = \sum_{i=1}^{n} (Y_i - (\beta_1 X_i + \beta_0))^2$
- Take partial derivatives with respect to β_0 and set them to zero.
  - $\frac{\partial l}{\partial \beta_0} = -2 \sum_{i=1}^{n} (Y_i - \beta_1 X_i - \beta_0) = 0$
  - $\sum_{i=1}^{n} Y_i = \beta_1 \sum_{i=1}^{n} X_i + n\beta_0$
  - $\beta_0 = \frac{1}{n} (\sum_{i=1}^{n} Y_i - \beta_1 \sum_{i=1}^{n} X_i)$
- Take $\bar{X}$ and $\bar{Y}$ as sample means of X and Y respectively.
- Solving for β_0 gives:
  - $\beta_0 = \bar{Y} - \beta_1 \bar{X}$
- Take partial derivatives with respect to β_1 and set them to zero.
  - $\frac{\partial l}{\partial \beta_1} = -2 \sum_{i=1}^{n} X_i (Y_i - \beta_1 X_i - \beta_0) = 0$
- Simplify:
  - $\sum_{i=1}^{n} X_i Y_i = \beta_1 \sum_{i=1}^{n} X_i^2 + \beta_0 \sum_{i=1}^{n} X_i$
- Substitute β_0 into equation:
  - $\sum_{i=1}^{n} X_i Y_i = \beta_1 \sum_{i=1}^{n} X_i^2 + (\bar{Y} - \beta_1 \bar{X}) \sum_{i=1}^{n} X_i$
- Solve for β_1:
  - $\sum_{i=1}^{n} X_i Y_i - \bar{Y} \sum_{i=1}^{n} X_i = \beta_1 \left(\sum_{i=1}^{n} X_i^2 - \bar{X} \sum_{i=1}^{n} X_i \right)$
- Finally:
  - $\beta_1 = \frac{\sum_{i=1}^{n} X_i Y_i - \bar{Y} \sum_{i=1}^{n} X_i}{\sum_{i=1}^{n} X_i^2 - \bar{X} \sum_{i=1}^{n} X_i}$

(b)

- $\beta_0 = 5000$
- All previous steps remain the same for β_1.
- Substitute β_0 into equation:

  - $\sum_{i=1}^{n} X_i Y_i = \beta_1 \sum_{i=1}^{n} X_i^2 + 5000 \sum_{i=1}^{n} X_i$
- Solve for β_1:
  - $\sum_{i=1}^{n} X_i Y_i - 5000 \sum_{i=1}^{n} X_i = \beta_1 \left(\sum_{i=1}^{n} X_i^2\right)$
- Finally:
  - $\beta_1 = \frac{\sum_{i=1}^{n} X_i Y_i - 5000 \sum_{i=1}^{n} X_i}{\sum_{i=1}^{n} X_i^2}$

(c)

- $\beta_0 = 0$
- Take partial derivatives with respect to β_1 and set them to zero.
  - $\frac{\partial l}{\partial \beta_1} = -2 \sum_{i=1}^{n} X_i^2 (Y_i - \beta_1 X_i^2) = 0$
- Simplify:
  - $\sum_{i=1}^{n} X_i^2 Y_i = \beta_1 \sum_{i=1}^{n} X_i^4$
- Solve for β_1:
  - $\beta_1 = \frac{\sum_{i=1}^{n} X_i^2 Y_i}{\sum_{i=1}^{n} X_i^4}$

(d)

- $\beta_0 = 0$
- Rewrite as:
  - $Y = \beta_1 log(X) + \epsilon$
- Take partial derivatives with respect to β_1 and set them to zero.
  - $\frac{\partial l}{\partial \beta_1} = -2 \sum_{i=1}^{n} Y_i - log(\beta_1) - log(X_i) = 0$
- Solve for $log(\beta_1)$:
  - $log(\beta_1) = \frac{\sum_{i=1}^{n} Y_i - log(X_i)}{n}$
- Finally solve for β_1:
  - $\beta_1 = e^{\frac{\sum_{i=1}^{n} Y_i - log(X_i)}{n}}$

2. Derive the Method of Moments Estimator(s) for each model. You do *not* need to prove that the mean of observations is the UMVUE.

(a)

- Residual: $\epsilon = Y - \beta_1 X - \beta_0$
- Take the expectation of the residual:
  - $E[\epsilon] = E[Y] - \beta_1 E[X] - \beta_0$
- Since $E[\epsilon] = 0$:
  - $E[Y] = \beta_1 E[X] + \beta_0$
- Estimate expectations using sample means:
  - $\bar{Y} = \beta_1 \bar{X} + \beta_0$
- Solve for β_0:
  - $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$
- Take the expectation of Y:
  - $E[Y] = \beta_1 E[X] + \beta_0$
- Second moment is based on covariance:
  - $Cov(X, Y) = \beta_1 Var(X)$
- Solve for β_1:
  - $\hat{\beta}_1 = \frac{Cov(X,Y)}{Var(X)}$

(b)

- $\beta_0 = 5000$
- All previous steps remain the same for $ _1.
- Estimate expectations using sample means:
  - $\bar{Y} = \beta_1 \bar{X} + 5000$
- Solve for β_1:
  - $\hat{\beta}_1 = \frac{\bar{Y} - 5000}{\bar{X}}$

(c)

- $\beta_0 = 0$
- Take the expectation of the residual:
  - $E[\epsilon] = E[Y] - \beta_1 E[X^2]$
- Since $E[\epsilon] = 0$:
  - $E[Y] = \beta_1 E[X^2]$
- Estimate expectations using sample means:
  - $\bar{Y} = \hat{\beta}_1 \bar{X^2}$
- Solve for β_1:
  - $\hat{\beta}_1 = \frac{\bar{Y}}{X^2}$

(d)

- $\beta_0 = 0$
- Expectation of Y:
  - $E[Y] = E[log(\beta X) + \epsilon]$
  - $E[Y] = E[log(\beta) + log(X) + \epsilon]$
- Since $E[\epsilon] = 0$:
  - $E[Y] = log(\beta) + E[log(X)]$
- Estimate expectations using sample means:
  - $\bar{Y} = log(\hat{\beta}) + \bar{X}$
- Solve for β:
  - $\hat{\beta} = e^{\bar{Y} - \bar{X}}$

3. Derive the Maximum Likelihood Estimator(s) for each model.

(a) $Y = \beta_1 X + \beta_0 + \epsilon$

- Likelihood function: $L(\beta_0, \beta_1) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(Y_i - \beta_1 X_i - \beta_0)^2}{2\sigma^2}\right)$
- Log-likelihood function: $l(\beta_0, \beta_1) = -\frac{n}{2} log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \beta_1 X_i - \beta_0)^2$
- Take partial derivatives with respect to β_0 and set them to zero.
  - $\frac{\partial l}{\partial \beta_0} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (Y_i - \beta_1 X_i - \beta_0) = 0$
  - $\sum_{i=1}^{n} Y_i = \beta_1 \sum_{i=1}^{n} X_i + n\beta_0$
  - $\beta_0 = \frac{1}{n} (\sum_{i=1}^{n} Y_i - \beta_1 \sum_{i=1}^{n} X_i)$
- Take $\bar{X}$ and $\bar{Y}$ as sample means of X and Y respectively.
- Solving for β_0 gives:
  - $\beta_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$
- Take partial derivatives with respect to β_1 and set them to zero.
  - $\frac{\partial l}{\partial \beta_1} = \frac{1}{\sigma^2} \sum_{i=1}^{n} X_i (Y_i - \beta_1 X_i - \beta_0) = 0$
- Simplify:
  - $\sum_{i=1}^{n} X_i Y_i = \beta_1 \sum_{i=1}^{n} X_i^2 + \beta_0 \sum_{i=1}^{n} X_i$
- Substitute β_0 into equation:

  - $\sum_{i=1}^{n} X_i Y_i = \beta_1 \sum_{i=1}^{n} X_i^2 + (\bar{Y} - \beta_1 \bar{X}) \sum_{i=1}^{n} X_i$
- Solve for β_1:
  - $\sum_{i=1}^{n} X_i Y_i - \bar{Y} \sum_{i=1}^{n} X_i = \beta_1 \left(\sum_{i=1}^{n} X_i^2 - \bar{X} \sum_{i=1}^{n} X_i\right)$
- Finally:
  - $\hat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i Y_i - \bar{Y} \sum_{i=1}^{n} X_i}{\sum_{i=1}^{n} X_i^2 - \bar{X} \sum_{i=1}^{n} X_i}$

(b)

- $\beta_0 = 5000$
- All previous steps remain the same for $ _1.
- Substitute β_0 into equation:
  - $\sum_{i=1}^{n} X_i Y_i = \beta_1 \sum_{i=1}^{n} X_i^2 + 5000 \sum_{i=1}^{n} X_i$
- Solve for β_1:
  - $\sum_{i=1}^{n} X_i Y_i - 5000 \sum_{i=1}^{n} X_i = \beta_1 \left(\sum_{i=1}^{n} X_i^2\right)$
- Finally:
  - $\hat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i Y_i - 5000 \sum_{i=1}^{n} X_i}{\sum_{i=1}^{n} X_i^2}$

(c)

- $\beta_0 = 0$
- Take partial derivatives with respect to β_1 and set them to zero.
  - $\frac{\partial l}{\partial \beta_1} = \frac{1}{\sigma^2} \sum_{i=1}^{n} X_i^2 (Y_i - \beta_1 X_i^2) = 0$
- Simplify:
  - $\sum_{i=1}^{n} X_i^2 Y_i = \beta_1 \sum_{i=1}^{n} X_i^4$
- Solve for β_1:
  - $\hat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i^2 Y_i}{\sum_{i=1}^{n} X_i^4}$

(d)

- $\beta_0 = 0$
- Rewrite as:
  - $Y = \beta_1 log(X) + \epsilon$
- Take partial derivatives with respect to β_1 and set them to zero.
  - $\frac{\partial l}{\partial \beta_1} = \frac{1}{\sigma^2} \sum_{i=1}^{n} Y_i - log(\beta_1) - log(X_i) = 0$
- Solve for $log(\beta_1)$:
  - $log(\hat{\beta}_1) = \frac{\sum_{i=1}^{n} Y_i - log(X_i)}{n}$
- Finally solve for β_1:
  - $\hat{\beta}_1 = e^{\frac{\sum_{i=1}^{n} Y_i - log(X_i)}{n}}$

1.2 Section B: Coding

[61]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

# load data
data = pd.read_csv('AmesHousing.csv')
data = data[['SalePrice', 'Lot Area']]
```

```
data.head()
```

[61]:

	SalePrice	Lot Area
0	215000	31770
1	105000	11622
2	172000	14267
3	244000	11160
4	189900	13830

1. Use either the LinearRegression function in scikit-learn to fit the model from Part A. Do not do any cross-validation or tuning for this; simply fit the model on the full data, and print out the coefficient(s) that were estimated.

[32]:
```
model = LinearRegression()
model.fit(data[['Lot Area']], data['SalePrice'])
print('Intercept:', model.intercept_)
print('Slope:', model.coef_)
```

```
Intercept: 153373.89290717372
Slope: [2.70224462]
```

2. Write a function that estimates the coefficients by random choice and brute force.

[33]:
```
def slr_guess_and_check(x, y, num_candidates=1000):
    min_loss = float('inf')
    for _ in range(num_candidates):
        b0 = np.random.uniform(1000, 200000)
        b1 = np.random.uniform(-10, 10)
        yhat = b0 + b1*x
        loss = np.mean((y - yhat)**2)
        if loss < min_loss:
            min_loss = loss
            best_b0 = b0
            best_b1 = b1
    return best_b0, best_b1
```

3. Apply slr_guess_and_check to the AMES Housing data. Compare the results to the "correct" coefficients that you got in B1. Make a plot, table, or summary statement to describe how close your function tends to get to the right answers. (Since your function uses randomization, you will need to run it many times to comment on how close it "tends" to get.)

[45]:
```
b0, b1 = slr_guess_and_check(data['Lot Area'], data['SalePrice'])
print('Intercept:', b0)
print('Slope:', b1)
```

```
Intercept: 148741.56333684328
Slope: 2.7793548532319825
```

Model	β_0	β_1
A	153373.89290717372	2.70224462
B	148741.56333684328	2.7793548532319825

The function tends to get very close to the right answers after running multiple times. It overshoots sometimes but stays in the right range for both β_0 and β_1.

4. Now write a function that "updates" b0 and b1.

[48]:
```
def slr_improvement_method(x, y, num_iterations=1000):
    b0 = 0
    b1 = 0
    yhat = b0 + b1*x
    loss = np.mean((y - yhat)**2)
    for _ in range(num_iterations):
        b0_c = np.random.uniform(1000, 200000)
        yhat = b0_c + b1*x
        loss_c = np.mean((y - yhat)**2)
        if loss_c < loss:
            b0 = b0_c
            loss = loss_c
        b1_c = np.random.uniform(-10, 10)
        yhat = b0 + b1_c*x
        loss_c = np.mean((y - yhat)**2)
        if loss_c < loss:
            b1 = b1_c
            loss = loss_c
    return b0, b1
```

[54]:
```
b0, b1 = slr_improvement_method(data['Lot Area'], data['SalePrice'])
print('Intercept:', b0)
print('Slope:', b1)
```

```
Intercept: 154324.68369496462
Slope: 2.6279768052587453
```

Model	β_0	β_1
A	153373.89290717372	2.70224462
B	153380.35319410317	2.7070442508083747

This function is very accurate and stays in a very tight range around the correct values for both β_0 and β_1. Even after running multiple times there are no significant deviations from the correct values.

1.3 Section C: Concepts

1. Plug in appropriate summary statistics from the AMES data to get the least squares estimators for the three models in Section A. Are they similar, or not? Give an intuitive explanation for this.

[60]:
```
# Model 1: Y = B1X + B0
x = data['Lot Area']
y = data['SalePrice']
b1 = np.sum(x*y) - np.mean(y)*np.sum(x)
b1 /= np.sum(x**2) - np.mean(x)*np.sum(x)
b0 = np.mean(y) - b1*np.mean(x)
print('Model 1:')
print('Intercept:', b0)
print('Slope:', b1)
print()

# Model 2: Y = BX + 5000
b1 = np.sum(x*y) - 5000*np.sum(x)
b1 /= np.sum(x**2)
print('Model 2:')
print('Intercept:', 5000)
print('Slope:', b1)
print()

# Model 3: Y = BX^2
b1 = np.sum(x**2*y)
b1 /= np.sum(x**4)
print('Model 3:')
print('Intercept:', 0)
print('Slope:', b1)
```

```
Model 1:
Intercept: 153373.89290717372
Slope: 2.7022446157281115

Model 2:
Intercept: 5000
Slope: 11.824643640102646

Model 3:
Intercept: 0
Slope: -0.09872158273462628
```

Since the first model is a simple linear regression, the least squares estimator is the same as from the results from Section B. However for the second and third model they are not similar to Model since the intercept has been hardcoded as 5000 and this affects calculating β_1. The third model is a quadratic model and the least squares estimator is extremely different from the other 2 models and this is because the model is not linear and the relationship between X and Y is not linear.

2. Consider models (a) and (b) from Section A. Mathematically, which one will have the smaller optimized squared error loss? Why? Do you expect this to be true for any chosen loss function?

Model (a) will have a smaller optimized squared error loss because the intercept is not hardcoded and the model is more flexible. Model (b) has a fixed intercept of 5000 and this will not be able to capture the relationship between X and Y as well as Model (a). Having 2 free parameters will allow model (a) to fit better thus having a smaller optimized squared errors loss. This is true for the squared error loss function but may not be true for other loss functions.

3. Discuss your summaries/plots from B3 and B5. How did these two (somewhat silly) approaches to estimation differ? Why?

The brute force approach with random choice was not accurate and differed greatly than the improved brute force approach. The difference was that improved approach had a way to update the coefficients based on the error and keep minimizing the error. The brute force approach was random and did not have a way to update the coefficients based on the error and thus was not accurate.

In the following problems, consider a possible loss function that we could use instead of squared error loss. For each, start the process of deriving the estimator for the model $Y = \beta * X + \epsilon$. When you hit a reason it is impossible to continue, stop, and then describe in words why you were not able to continue.

4. $l(\beta) = \max_i |Y_i - \hat{Y}_i|$

- The loss function is not differentiable and so the derivative cannot be taken and set to zero to find the estimator for β.

5. $l(\beta) = \sum_{i=1}^{n} |Y_i - \hat{Y}_i|$

- The loss function is not differentiable and so the derivative cannot be taken and set to zero to find the estimator for β.

6. $l(\beta) = \sum_{i=1}^{n} Y_i * log(\hat{Y}_i)$

- Take the partial derivative with respect to β:
  - $\frac{\partial l}{\partial \beta} = \sum_{i=1}^{n} Y_i * \frac{1}{\hat{Y}_i} * \frac{\partial \hat{Y}_i}{\partial \beta}$
- The derivative of $log(\hat{Y}_i)$ is not defined and so the derivative cannot be taken and set to zero to find the estimator for β.
- It is not defined because $log(\hat{Y}_i)$ is not differentiable at 0.