VG441 Problem Set 1

Pan, Chongdan ID:516370910121

May 31, 2020

1 Problem 1

$\theta^T X^T X\theta = (X\theta)^T(X\theta) = (X\theta)^2$
Assume $X \in \mathbb{R}^{m\times n}, \theta \in \mathbb{R}^{n\times 1}$ then $= (X\theta)^2 = \sum_{i=1}^{m}(\sum_{j=1}^{n} x_{ij}\theta_j)^2$

$$\frac{\mathrm{d}\theta^T X^T X\theta^2}{\mathrm{d}\theta} = \frac{\mathrm{d}(X\theta)^2}{\mathrm{d}\theta} = \begin{bmatrix} \frac{\partial(X\theta)^2}{\partial\theta_j} \\ \vdots \\ \frac{\partial(X\theta)^2}{\partial\theta_n} \end{bmatrix} = \begin{bmatrix} 2\sum_{i=1}^{m}(\sum_{j=1}^{n} x_{ij}x_{i1}\theta_j) \\ \vdots \\ 2\sum_{i=1}^{m}(\sum_{j=1}^{n} x_{ij}x_{in}\theta_j) \end{bmatrix}$$

$$2X^TX\theta = 2\begin{bmatrix} \sum_{i=1}^{m} x_{i1}x_{i1} & \cdots & \sum_{i=1}^{m} x_{i1}x_{in} \\ \vdots & \vdots & \vdots \\ \sum_{i=1}^{m} x_{in}x_{i1} & \cdots & \sum_{i=1}^{m} x_{in}x_{in} \end{bmatrix}\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} = \begin{bmatrix} 2\sum_{i=1}^{m}(\sum_{j=1}^{n} x_{ij}x_{i1}\theta_j) \\ \vdots \\ 2\sum_{i=1}^{m}(\sum_{j=1}^{n} x_{ij}x_{in}\theta_j) \end{bmatrix}$$

Therefore, the derivative of $\theta^T X^T X\theta$ with respect to θ is $2X^TX\theta$

2 Problem 2

- The average value for salary is 5875.So before the first iteration:

Age	Home Owner	Car Owner	Having kids	Salary	F0	PR0
40	YES	YES	YES	10000	5875	4125
20	NO	NO	NO	500	5875	-5375
50	YES	NO	YES	8000	5875	2125
30	YES	NO	NO	5000	5875	-875

The deviance is 5118750.
For **Home Owner** node: deviance is 12666666.67
For **CAR Owner** node: deviance is 28500000
For **Having Kids** node:deviance is 12125000
For **Age** $\leq$ **25** node: deviance is 12666666.67
For **Age** $\leq$ **35** node: deviance is 12125000
For **Age** $\leq$ **45** node: deviance is 45166666

So we set **Having Kids** as the highest node, **Home Owner** as the left lower node,**CAR Owner** as the right lower node

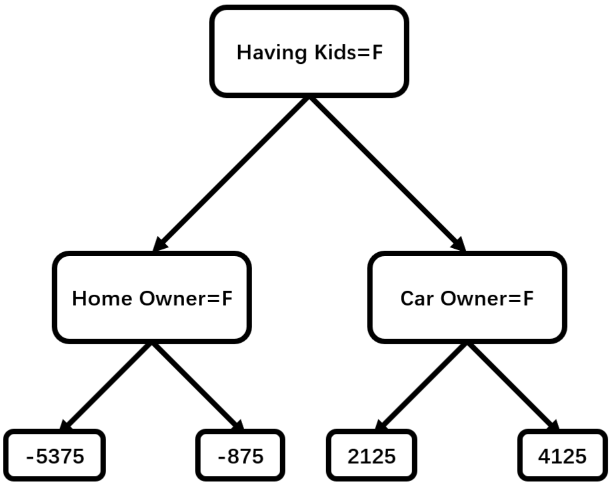

Figure 1: First decision tree for GBM

Age	Home Owner	Car Owner	Having kids	Salary	F0	PR0	F1	PR1	F2	PR2
40	YES	YES	YES	10000	5875	4125	6287.5	3712.5	6658.75	3341.25
20	NO	NO	NO	500	5875	-5375	5337.5	-4837.5	4853.75	-4353.75
50	YES	NO	YES	8000	5875	2125	6087.5	1912.5	6278.5	1721.25
30	YES	NO	NO	5000	5875	-875	5787.5	-787.5	5708.75	-708.75

- The average value for salary is 5875.So before the first iteration:

Age	Home Owner	Car Owner	Having kids	Salary	F0	PR0
40	YES	YES	YES	10000	5875	4125
20	NO	NO	NO	500	5875	-5375
50	YES	NO	YES	8000	5875	2125
30	YES	NO	NO	5000	5875	-875

SS with $\lambda = 1$ is 994050000
For **Home Owner** node: Gain is 21667968.75
For **CAR Owner** node: Gain is 12761718.75
For **Having Kids** node:Gain is 26041666.67
For **Age** $\leq$ **25** node: Gain is 21667968.75
For **Age** $\leq$ **35** node: Gain is 26041666.67
For **Age** $\leq$ **45** node: Gain is 3386718.75

So we set **Having Kids** as the highest node, **Home Owner** as the left lower node,**CAR Owner** as the right lower node
The **Home Owner** node has Gain with $1807292 > \gamma$, so it'll remain.
The **CAR Owner** node has Gain with $-2255208 < \gamma$, so it'll be deleted.
After pruning, the XGBoost tree becomes:

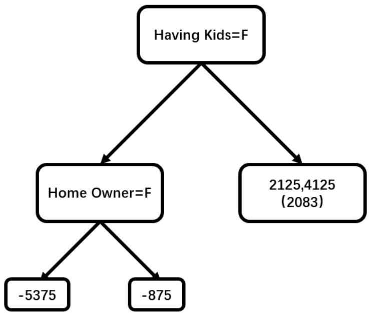

Figure 2: First XGBoost Tree

Age	Home Owner	Car Owner	Having kids	Salary	F0	PR0	F1	PR1	F2	PR2
40	YES	YES	YES	10000	5875	4125	6083.3	3916.7	6277.75	3722.25
20	NO	NO	NO	500	5875	-5375	5606.3	-5106.3	5351	-4851
50	YES	NO	YES	8000	5875	2125	6083.3	1916.7	6277.75	1722.25
30	YES	NO	NO	5000	5875	-875	5831.3	-831.3	5790	-790

3 Problem 3

- The Linear Regression model leads to $MSE = 3.74 \times 10^9, R^2 = 0.66$

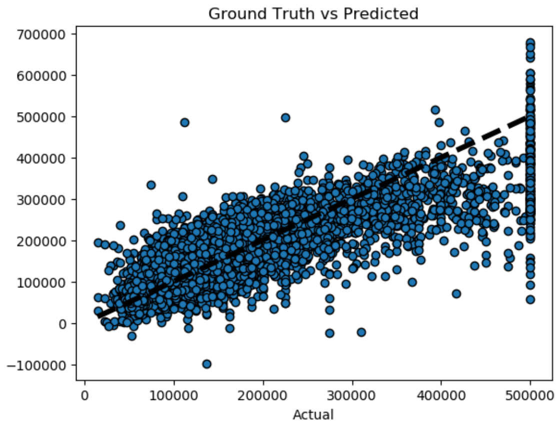

Figure 3: Actual Value vs Linear Regression Predicted Value

- The GBM model leads to $MSE = 2.01 \times 10^9, R^2 = 0.81$

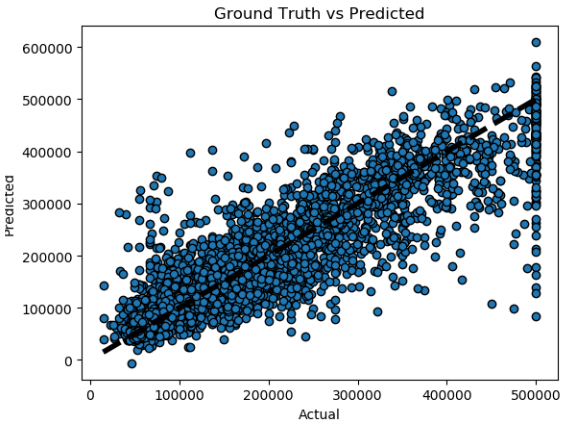

Figure 4: Actual Value vs Gradient Boosting Predicted Value

- The XGBoost model leads to $MSE = 1.85 \times 10^9, R^2 = 0.83$

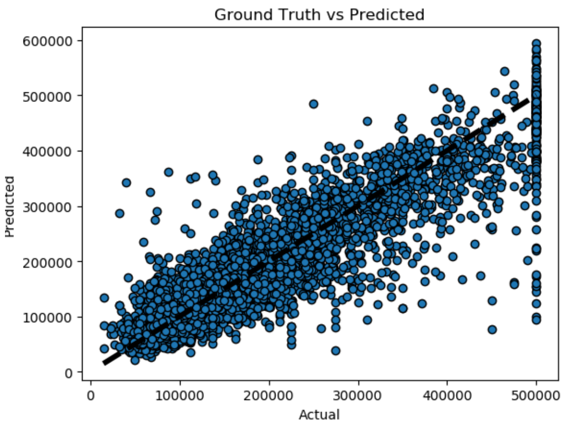

Figure 5: Actual Value vs XGBoost Predicted Value

From the figure, MSE, R^2 we get that the XGBoost Model can achieve the best prediction result, and Linear Regression Model's result is worst.

Python Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

```
from sklearn.model_selection import cross_val_predict

df=pd.DataFrame(pd.read_csv(r"D:\PANDA\Study\VG441\Homework\Problem Set 1\Cal_Housing.csv"))

class_mapping={'NEAR BAY':0, 'INLAND':1}
df['ocean_proximity']=df['ocean_proximity'].map(class_mapping) #
df=df.dropna(axis=0,how='any',inplace=False) # nan

X=df[['longitude','latitude','housing_median_age','total_rooms','total_bedrooms','population','ho
Y=df[['median_house_value']]
X_train,X_test,Y_train,Y_test=train_test_split(X, Y, test_size=0.8)
```

Linear Regression Model

```
model=linear_model.LinearRegression()
```

GBM Model

```
params = {'n_estimators': 500, 'max_depth': 4, 'min_samples_split': 2, 'learning_rate': 0.05, 'lo
model = ensemble.GradientBoostingRegressor(**params)
```

XGBoost Model

```
params = {'n_estimators': 500, "objective":"reg:linear",'colsample_bytree': 0.5,'learning_rate':
                'max_depth': 5, 'alpha': 1}
model = xgb.XGBRegressor(**params)
```

Model Fit

```
model.fit(X_train, Y_train)
model_score = model.score(X_train,Y_train)
Y_predicted = model.predict(X_test)
```

Result and Visualization

```
print("Mean squared error: %.2f"% mean_squared_error(Y_test, Y_predicted))
print('R2 sq: ',r2_score(Y_test, Y_predicted))
fig, ax = plt.subplots()
ax.scatter(Y_test, Y_predicted, edgecolors=(0, 0, 0))
ax.plot([Y_test.min(), Y_test.max()], [Y_test.min(), Y_test.max()], 'k--', lw=4)
ax.set_xlabel('Actual')
ax.set_ylabel('Predicted')
ax.set_title("Ground Truth vs Predicted")
plt.show()
```