Lab8

September 5, 2020

0.1 Without regularization

[36]:
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.datasets import make_regression
```

[2]:
```
# Load the diabetes dataset
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
```

[5]:
```
x_train,x_test,y_train,y_test=train_test_split(diabetes_X,diabetes_y,random_state=0)
```

[12]:
```
regr=linear_model.LinearRegression()
regr.fit(x_train,y_train)
y_pred=regr.predict(x_test)
```

[13]:
```
print('Regularization coefficients:',regr.coef_)
```

```
Regularization coefficients: [ -43.26774487 -208.67053951  593.39797213
302.89814903 -560.27689824
  261.47657106   -8.83343952  135.93715156  703.22658427   28.34844354]
```

[17]:
```
print("Mean square error: %.2f"% mean_squared_error(y_test,y_pred))
```

```
Mean square error: 3180.20
```

[18]:
```
print('Coefficient of determination: %.2f'
      % r2_score(y_test, y_pred))
```

```
Coefficient of determination: 0.36
```

0.2 With regularization - Ridge

[30]:
```
regr=linear_model.Ridge()
regr.fit(x_train,y_train)
y_pred=regr.predict(x_test)
```

[27]:
```
print('Regularization coefficients:',regr.coef_)
```

```
Regularization coefficients: [  21.19927911  -60.47711393  302.87575204
179.41206395    8.90911449
  -28.8080548  -149.30722541  112.67185758  250.53760873   99.57749017]
```

[29]:
```
print("Mean square error: %.2f"% mean_squared_error(y_test,y_pred))
```

```
Mean square error: 3192.33
```

[31]:
```
print('Coefficient of determination: %.2f'
      % r2_score(y_test, y_pred))
```

```
Coefficient of determination: 0.36
```

0.3 With Regularization - Lasso

[32]:
```
regr=linear_model.Lasso()
regr.fit(x_train,y_train)
y_pred=regr.predict(x_test)
```

[33]:
```
print('Regularization coefficients:',regr.coef_)
```

```
Regularization coefficients: [  0.          -0.         442.67992538   0.
0.
   0.          -0.           0.         330.76014648   0.        ]
```

[34]:
```
print("Mean square error: %.2f"% mean_squared_error(y_test,y_pred))
```

```
Mean square error: 3583.42
```

[35]:
```
print('Coefficient of determination: %.2f'
      % r2_score(y_test, y_pred))
```

```
Coefficient of determination: 0.28
```

0.4 Create synthetic dataset

[37]:
```
X,y=make_regression(n_samples=1000,n_features=1,random_state=0)
```

[38]:
```
x_train,x_test,y_train,y_test=train_test_split(X,y,random_state=0)
```

[39]:
```
regr=linear_model.Lasso()
regr.fit(x_train,y_train)
y_pred=regr.predict(x_test)
```

[40]:
```
print('Regularization coefficients:',regr.coef_)
```

```
Regularization coefficients: [81.17093529]
```

[41]:
```
print("Mean square error: %.2f"% mean_squared_error(y_test,y_pred))
```

```
Mean square error: 0.99
```

[42]:
```
print('Coefficient of determination: %.2f'
      % r2_score(y_test, y_pred))
```

```
Coefficient of determination: 1.00
```

[44]:
```
plt.scatter(x_test, y_test,  color='black')
plt.plot(x_test, y_pred, color='blue', linewidth=3)

plt.xticks(())
plt.yticks(())

plt.show()
```