06 Implement SGD

January 24, 2019

In [1]:
```
import warnings
warnings.filterwarnings("ignore")
from sklearn.datasets import load_boston
from random import seed
from random import randrange
from csv import reader
from math import sqrt
from sklearn import preprocessing
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
from sklearn.linear_model import SGDRegressor
from sklearn import preprocessing
from sklearn.metrics import mean_squared_error
```

In [2]:
```
boston = load_boston()
```

In [3]:
```
print(boston.data.shape)
```

```
(506, 13)
```

In [4]:
```
print(boston.feature_names)
```

```
['CRIM' 'ZN' 'INDUS' 'CHAS' 'NOX' 'RM' 'AGE' 'DIS' 'RAD' 'TAX' 'PTRATIO'
 'B' 'LSTAT']
```

In [5]:
```
X = load_boston().data
Y = load_boston().target
```

In [6]:
```
scaler = preprocessing.StandardScaler().fit(X)
X = scaler.transform(X)
```

In [619]:
```
clf = SGDRegressor(alpha=0.01)
clf.fit(X, Y)
print(mean_squared_error(Y, clf.predict(X)))
```

```
22.774307243197338
```

```
In [620]: X.shape
```

```
Out[620]: (506, 13)
```

```
In [621]: Y.shape
```

```
Out[621]: (506,)
```

```
In [644]: def SGD(X, Y, learning_rate=0.01, n_iter = 1000, batch_size = 100):
              W = np.zeros(X.shape[1])
              b = 0.0
              r = learning_rate
              rt_power = 0.25
              for i in range(1,n_iter+1):
                  idx = np.random.randint(0, len(X),batch_size)
                  x_k = X[idx]
                  y_k = Y[idx]
                  N = float(batch_size)
                  error = y_k - (np.dot(x_k, W) - b)
                  W -= r * (-2/N) * x_k.T.dot(error)
                  b -= r * np.sum(error)
                  r = learning_rate / pow(i, rt_power)
              return W,abs(b)
```

```
In [645]: W, b= SGD(X,Y)
```

```
In [646]: W
```

```
Out[646]: array([-0.68561721,  0.55167202, -0.39747528,  0.77377464, -1.0242332 ,
                  3.1512419 , -0.1606775 , -2.11335569,  0.8508525 , -0.5005695 ,
                 -1.83639621,  0.88158787, -3.43708822])
```

```
In [647]: b
```

```
Out[647]: 22.489897905395324
```

```
In [648]: pred = X.dot(W) + b
```

```
In [649]: MSE = mean_squared_error(Y, pred)
```

```
In [650]: MSE
```

```
Out[650]: 22.734997202702246
```

```
In [652]: x = PrettyTable(["Implimentation", "learning_rate", "MSE"])

          x.add_row(["sk_learn" , "0.01", 22.774])
          x.add_row(["from scrach","0.01", 22.734])

          print(x.get_string(title="SGD Model"))
```

```
+----------------+---------------+--------+
| Implimentation | learning_rate |  MSE   |
+----------------+---------------+--------+
|    sk_learn    |      0.01     | 22.774 |
|  from scrach   |      0.01     | 22.734 |
+----------------+---------------+--------+
```

In [ ]: