Linear Regression

October 12, 2015

```
In [1]: import numpy as np

In [2]: import urllib

In [3]: import scipy.optimize

In [25]: import random
         from sklearn import datasets, linear_model, metrics

In [5]: def parseData(fname):
          for l in urllib.urlopen(fname):
            yield eval(l)

In [6]: print "Reading data..."
        data = list(parseData("file:///Users/YW/Dropbox/UCSD/FA2015/CSE255/HW/hw1/beer_50000.json"))
        print "done"

Reading data...
done

In [7]: def feature(datum):
          feat = [1]
          feat.append(datum['beer/ABV'])
          return feat

In [8]: X = [feature(d) for d in data]

In [9]: y = [d['review/taste'] for d in data]

In [10]: theta, residuals, rank, s = np.linalg.lstsq(X, y)

In [11]: print 'Theta0:%s, Theta1:%s'%(theta[0], theta[1])

Theta0:3.11521115232, Theta1:0.109055074172

In [37]: X1 = np.matrix(X)
         y1 = np.matrix(y)
         print np.linalg.inv(X1.T * X1) * X1.T * y1.T

[[ 3.11521115]
 [ 0.10905507]]

In [79]: def feature5(datum):
             feat = [1]
             feat.append(datum['beer/ABV'])
             feat.append(datum['beer/ABV']**2)
             feat.append(datum['beer/ABV']**3)
             feat.append(datum['beer/ABV']**4)
             feat.append(datum['beer/ABV']**5)
             return feat
```

```
In [80]: X5 = [feature5(d) for d in data]

In [81]: y5 = [d['review/taste'] for d in data]

In [82]: theta, residuals, rank, s = np.linalg.lstsq(X5, y5)

In [83]: print theta, residuals, rank, s

[  1.40998006e+00   6.71979699e-01  -5.57829661e-02   1.95198865e-03
  -3.03848616e-05   1.73376523e-07] [ 21821.27545336] 6 [  6.93445043e+08   2.17982482e+06   7.70108652
   9.97195416e+01   5.73256063e+00]

In [84]: regr = linear_model.LinearRegression(fit_intercept=False)

In [85]: regr.fit(X5, y5)

Out[85]: LinearRegression(copy_X=True, fit_intercept=False, n_jobs=1, normalize=False)

In [86]: print 'Coefficients:%s\n'% regr.coef_

Coefficients:[  1.40998006e+00   6.71979699e-01  -5.57829661e-02   1.95198865e-03
  -3.03848616e-05   1.73376523e-07]

In [87]: print 'MSE: %s' % np.mean((regr.predict(X5) - y5) ** 2)
         print 'MSE: %.5f' % metrics.mean_squared_error(y5, regr.predict(X5), sample_weight=None)

MSE: 0.436425509067
MSE: 0.43643

In [88]: train = data[:25000]
         test = data[25000:]

In [89]: def features(datum, degree):
             feat = [1]
             for i in xrange(1, degree+1):
                 feat.append(datum['beer/ABV']**i)
             return feat

In [101]: lastTestingError = 0

In [114]: deg = 1

In [115]: while True:
              regr = linear_model.LinearRegression(fit_intercept=False)
              X = [features(d, deg) for d in train]
              y = [d['review/taste'] for d in train]
              X_test = [features(d, deg) for d in test]
              y_test = [d['review/taste'] for d in test]
              regr.fit(X, y)
              thisTestingError = np.mean((regr.predict(X_test) - y_test) ** 2)
              thisTrainError = np.mean((regr.predict(X)-y) ** 2)
              print "Train:", thisTrainError
              print "Test:", thisTestingError
              print "coef:", regr.coef_
              print len(X[0])
              if np.fabs(thisTestingError-lastTestingError) < 0.000001:
                  break
              deg = deg + 1
              lastTestingError = thisTestingError
```

```
Train: 0.483983105115
Test: 0.423776528023
coef: [ 2.99503282  0.11690802]
2
Train: 0.471743067557
Test: 0.427256126
coef: [ 2.62007309  0.20716481 -0.00496806]
3
Train: 0.457832195847
Test: 0.432820707084
coef: [  1.57847740e+00   5.19869113e-01  -2.97470415e-02   3.97626061e-04]
4
Train: 0.451641221709
Test: 0.438301732573
coef: [  7.17629022e-01   8.26164532e-01  -6.32189636e-02   1.67076516e-03
  -1.40075070e-05]
5
Train: 0.451335575196
Test: 0.439820774965
coef: [  1.16389773e+00   6.12905604e-01  -2.85943943e-02  -6.28209940e-04
   4.41325901e-05  -4.79191916e-07]
6
Train: 0.450769302041
Test: 0.443012263638
coef: [  1.94901771e+00   1.79935890e-01   5.75772252e-02  -8.45991886e-03
   3.82283840e-04  -7.06992367e-06   4.63100297e-08]
7
Train: 0.45074188066
Test: 0.442575342636
coef: [  1.70847546e+00   3.47096902e-01   1.39022021e-02  -2.95198356e-03
   2.49609809e-05   4.72513678e-06  -1.40946836e-07   1.12958340e-09]
8
Train: 0.450231148043
Test: 0.442487701726
coef: [  3.93690397e-01   1.40863997e+00  -3.13749111e-01   4.74923395e-02
  -4.20781201e-03   2.03296270e-04  -5.28225580e-06   6.93371692e-08
  -3.60135053e-10]
9
Train: 0.449716240495
Test: 0.439253459889
coef: [  6.57779065e-01   7.66518411e-01   4.06647965e-02  -3.84794380e-02
   6.77043490e-03  -5.85586427e-04   2.71796542e-05  -6.83946306e-07
   8.76795980e-09  -4.47490401e-11]
10
Train: 0.485239771582
Test: 0.457827607951
coef: [  8.10035383e-03   2.95349476e-02   7.80426486e-02   1.25171126e-01
  -4.56697918e-02   6.52647339e-03  -4.84697958e-04   2.00717535e-05
  -4.64486475e-07   5.59715969e-09  -2.72729507e-11]
11
Train: 0.627028248862
Test: 0.646353376223
coef: [  3.13393784e-04   2.12106023e-03   5.06678428e-03   1.51375880e-02
   2.59611730e-02  -9.65927443e-03   1.37038921e-03  -1.00082775e-04
```

```
   4.07376218e-06  -9.28854537e-08   1.10590758e-09  -5.33743373e-12]
12
Train: 2.85408190325
Test: 4.53394584743
coef: [  1.20060415e-08  -1.67445314e-06   5.25536588e-07   3.21145550e-06
   1.74858508e-05   7.57760222e-05   1.95196988e-04  -5.09012896e-05
   4.96757208e-06  -2.38620849e-07   6.03395828e-09  -7.69501605e-11
   3.89503838e-13]
13
Train: 8.18203669655
Test: 10.0715024042
coef: [  3.95168107e-14   1.06781904e-10   3.10753427e-12   3.01578879e-11
   2.67364021e-10   2.24185996e-09   1.68928241e-08   1.03194964e-07
   3.87685919e-07  -6.75444710e-08   4.24511502e-09  -1.25235102e-10
   1.76192140e-12  -9.53340770e-15]
14
Train: 9.554158671
Test: 11.0448045432
coef: [  2.16639158e-16  -5.08056120e-11  -2.29606780e-14   1.97454960e-13
   1.89012696e-12   1.75238256e-11   1.52805342e-10   1.19077076e-09
   7.48268931e-09   2.87797828e-08  -4.94168737e-09   3.08053486e-10
  -9.04048899e-12   1.26734339e-13  -6.83975029e-16]
15
Train: 10.6493250888
Test: 11.7727856209
coef: [  1.17714518e-18  -4.45631245e-13  -4.58771397e-15   1.23856013e-15
   1.24714430e-14   1.23881558e-13   1.18594897e-12   1.06296521e-11
   8.47778730e-11   5.43078453e-10   2.12212634e-09  -3.59970437e-10
   2.22832356e-11  -6.50993035e-13   9.09743964e-15  -4.89877188e-17]
16
Train: 14.658067654
Test: 14.6303906263
coef: [  2.58346426e-25  -1.12064050e-15  -2.07631352e-19   6.14616717e-22
   5.33209390e-21   6.67769289e-20   8.47113777e-19   1.08018529e-17
   1.36403547e-16   1.65985013e-15   1.85096217e-14   1.70915296e-13
   9.99592835e-13  -9.93492947e-14   3.61772635e-15  -5.74964411e-17
   3.36561166e-19]
17
Train: 15.0695872189
Test: 15.0707338786
coef: [  8.24007104e-28   1.23742293e-16  -3.04283198e-21  -3.21208823e-23
   1.84267302e-23   2.36419159e-22   3.08338023e-21   4.06979631e-20
   5.38745687e-19   7.03731870e-18   8.81727942e-17   1.00719518e-15
   9.47871416e-15   5.62466260e-14  -5.58960316e-15   2.03534267e-16
  -3.23472985e-18   1.89347485e-20]
18
Train: 15.3488085966
Test: 15.3919509124
coef: [  2.42412870e-30  -7.77039051e-18   4.74890909e-21   1.43152594e-25
   6.09204213e-26   7.72181044e-25   1.03250682e-23   1.40202437e-22
   1.92188975e-21   2.63248076e-20   3.54182580e-19   4.54793549e-18
   5.29779276e-17   5.06144231e-16   3.03755177e-15  -3.01830340e-16
   1.09902445e-17  -1.74663973e-19   1.02240519e-21]
19
```

```
Train: 15.8358242625
Test: 15.9550205998
coef: [  1.09721348e-44  -3.80173473e-28  -1.68836187e-32  -5.46659687e-36
   1.45912708e-38   2.64070845e-37   7.73757971e-36   2.23663703e-34
   6.35375978e-33   1.76457920e-31   4.75650672e-30   1.23161080e-28
   3.01607501e-27   6.81360469e-26   1.35910058e-24   2.18837256e-23
   2.21986060e-22  -1.46300726e-23   3.16900977e-25  -2.25541643e-27]
20
Train: 15.8358344939
Test: 15.9550264602
coef: [  1.16144638e-47   8.03196982e-30   8.59344924e-34   1.28367662e-36
  -2.96853875e-39   2.89867746e-40   8.58237099e-39   2.51480251e-37
   7.26966628e-36   2.06518724e-34   5.73554315e-33   1.54605131e-31
   4.00322961e-30   9.80346876e-29   2.21469970e-27   4.41763235e-26
   7.11310207e-25   7.21543628e-24  -4.75984657e-25   1.03181543e-26
  -7.34803958e-29]
21
Train: 15.8358386458
Test: 15.9550286554
coef: [  1.23480922e-50   1.43384821e-30  -5.93199965e-36   1.78108556e-38
  -5.26064528e-41   3.13096731e-43   9.43944138e-42   2.79559030e-40
   8.19179384e-39   2.36806648e-37   6.72730260e-36   1.86834530e-34
   5.03624712e-33   1.30404895e-31   3.19347321e-30   7.21436936e-29
   1.43904051e-27   2.31708659e-26   2.35041756e-25  -1.55172618e-26
   3.36590504e-28  -2.39825616e-30]
22
Train: 15.8358403672
Test: 15.955029486
coef: [  1.31263710e-53  -5.38892493e-32  -3.23662641e-36   3.15709069e-39
   2.62616297e-42   4.17109278e-46   1.02960295e-44   3.07504109e-43
   9.10714282e-42   2.66863746e-40   7.71445794e-39   2.19155799e-37
   6.08652461e-36   1.64066299e-34   4.24821298e-33   1.04034086e-31
   2.35023179e-30   4.68797439e-29   7.54838817e-28   7.65695804e-27
  -5.05828127e-28   1.09778541e-29  -7.82520950e-32]
23
```

```
In [108]: print X[0]
          print X[1]
          print regr.coef_
```

```
[1, 5.0, 25.0, 125.0, 625.0, 3125.0, 15625.0, 78125.0, 390625.0, 1953125.0, 9765625.0, 48828125.0, 2441
[1, 6.2, 38.440000000000005, 238.32800000000003, 1477.6336000000001, 9161.32832, 56800.23558400001, 352
[  1.31263710e-53  -5.38892493e-32  -3.23662641e-36   3.15709069e-39
   2.62616297e-42   4.17109278e-46   1.02960295e-44   3.07504109e-43
   9.10714282e-42   2.66863746e-40   7.71445794e-39   2.19155799e-37
   6.08652461e-36   1.64066299e-34   4.24821298e-33   1.04034086e-31
   2.35023179e-30   4.68797439e-29   7.54838817e-28   7.65695804e-27
  -5.05828127e-28   1.09778541e-29  -7.82520950e-32]
```

```
In [138]: print lastTestingError
          print thisTestingError
```

```
15.8358415497
15.8358416122
```

```
In [ ]:
```

In [ ]: