Study by example: sklearn - linear_model - ElasticNetCV

In [1]:

```
%matplotlib inline
import time
print('Session...\n\t\t', time.strftime("%a, %b %d, %Y at %H:%M:%S", tim
e.localtime()))
t_start_all = time.time()
```

```
Session...
                    Tue, May 30, 2017 at 12:13:44
```

In [2]:

```
#  http://scikit-learn.org/stable/modules/linear_model.html
from IPython.display import display
import matplotlib.pyplot as plt

import os
import sys

import numpy as np
import pandas as pd

from sklearn import linear_model

#         1) I used the function ElasticNetCV
#         2) I used l1_ratio=[0.1, 0.5, 1]
#         3) I used alphas=[0.0125, 0.025, 0.05, .125, .25, .5, 1., 2.,
 4.]
#         4) I used cv=4

import knpackage.data_cleanup_toolbox as datacln
```

In [3]:

```
#             train and test - data directory
data0_dir = '../../../Google Drive/zz_keg/AminInMay/Data_GDSC'
os.listdir(data0_dir)
```

Out[3]:

```
['features_test.csv',
 'features_train.csv',
 'response_test.csv',
 'response_train.csv']
```

In [4]:

```
#            load the training and test data:
features_train_df = pd.read_csv(os.path.join(data0_dir, 'features_train.
csv'), sep=',', index_col=0, header=0)
response_train_df = pd.read_csv(os.path.join(data0_dir, 'response_train.
csv'), sep=',', index_col=0, header=0)
#                 clean training data:
features_train_df, response_train_df, s = datacln.check_input_value_for_
gene_prioritazion(
    features_train_df, response_train_df)

features_test_df = pd.read_csv(os.path.join(data0_dir, 'features_test.cs
v'), sep=',', index_col=0, header=0)
response_test_df = pd.read_csv(os.path.join(data0_dir, 'response_test.cs
v'), sep=',', index_col=0, header=0)

features_test_df, response_test_df, _ = datacln.check_input_value_for_ge
ne_prioritazion(
    features_test_df, response_test_df)


print('\nTEST:\n\tfeatures:\t', features_test_df.shape, '\n\tresponse:
\t', response_test_df.shape)

display(response_train_df)
display(response_test_df)
```

```
TEST:
        features:       (13042, 119)
        response:       (1, 119)
```

	23132-87	5637	639-V	647-V	697	786-0	8-MG-BA	8505C
17-AAG	-1.563772	-2.85766	-1.644401	3.670938	0.451354	-1.872032	-2.356787	-0.090621

1 rows × 480 columns

	RPMI-8866	RS4-11	RT-112	RVH-421	RXF393	S-117	SAS	SBC-1	S
17-AAG	0.261312	4.431245	-1.963627	-1.654628	-2.003833	-1.178236	-2.06439	3.80875	-1

1 rows × 119 columns

In [5]:

```
#               find the buggy parts of the data
a = response_train_df.as_matrix();       print((a != a).sum(), 'Nan valu
es in response_train_df');    bad_b = 0

for r in features_train_df.index.tolist():
    b = features_train_df.loc[r].values
    if (b != b).sum() != 0: bad_b += 1

print(bad_b, 'Nan values in features_train_df')
```

```
0 Nan values in response_train_df
0 Nan values in features_train_df
```

In [6]:

```
#                 clean training data:
features_train_df, response_train_df, s = datacln.check_input_value_for_
gene_prioritazion(
    features_train_df, response_train_df)

features_test_df, response_test_df, _ = datacln.check_input_value_for_ge
ne_prioritazion(
    features_test_df, response_test_df)

print('TRAIN:\n\tfeatures:\t', features_train_df.shape, '\n\tresponse:
\t', response_train_df.shape)
print('\nTEST:\n\tfeatures:\t', features_test_df.shape, '\n\tresponse:
\t', response_test_df.shape)
```

```
TRAIN:
        features:        (13042, 480)
        response:        (1, 480)

TEST:
        features:        (13042, 119)
        response:        (1, 119)
```

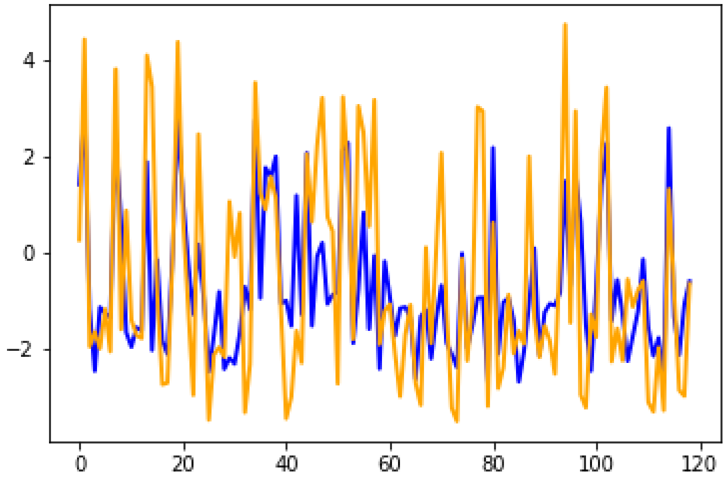
In [7]:

```
# reg_moE = linear_model.ElasticNetCV(l1_ratio, alphas, cv=4)
reg_moE = linear_model.ElasticNetCV()
t0 = time.time()
mo_predict = reg_moE.fit(features_train_df.transpose().values,
                         response_train_df.values[0]).predict(features_te
st_df.transpose().values)

fit_time = time.time() - t0
print('training time = %0.2f'%(fit_time))
```

```
training time = 78.81
```

In [8]:

```
plt.plot(mo_predict, color='blue', linewidth=2, label='Elastic net coeff
icients')

plt.plot(response_test_df.values[0], color='orange', linewidth=2,
label='test data')
plt.show()
```



In [10]:

```
print('Total time all cells =', time.time() - t_start_all)
%whos
```

```
Total time all cells = 5154.410343885422
Variable            Type            Data/Info
---------------------------------------------
a                   ndarray         1x480: 480 elems, type `float64`, 3
840 bytes
b                   ndarray         480: 480 elems, type `float64`, 384
0 bytes
bad_b               int             0
data0_dir           str             ../../../Google Drive/zz_keg/AminIn
May/Data_GDSC
datacln             module          <module 'knpackage.data_c<...>data_
cleanup_toolbox.py'>
display             function        <function display at 0x101432158>
features_test_df    DataFrame                        RPMI-886<...>13042
rows x 119 columns]
features_train_df   DataFrame                         23132-8<...>13042
rows x 480 columns]
fit_time            float           78.81080412864685
linear_model        module          <module 'sklearn.linear_m<...>inear
_model/__init__.py'>
mo_predict          ndarray         119: 119 elems, type `float64`, 952
bytes
np                  module          <module 'numpy' from '/Li<...>kage
s/numpy/__init__.py'>
os                  module          <module 'os' from '/Libra<...>3.5/l
ib/python3.5/os.py'>
pd                  module          <module 'pandas' from '/L<...>ages/
pandas/__init__.py'>
plt                 module          <module 'matplotlib.pyplo<...>es/ma
tplotlib/pyplot.py'>
r                   str             FLJ20152
reg_moE             ElasticNetCV    ElasticNetCV(alphas=None,<...>', to
l=0.0001, verbose=0)
response_test_df    DataFrame               RPMI-8866    RS4-<...>n\n[1
rows x 119 columns]
response_train_df   DataFrame               23132-87     5637<...>n\n[1
rows x 480 columns]
s                   str             Passed input value validation.
sys                 module          <module 'sys' (built-in)>
t0                  float           1496164433.875156
t_start_all         float           1496164424.244874
time                module          <module 'time' (built-in)>
```

In [ ]:

In [ ]: