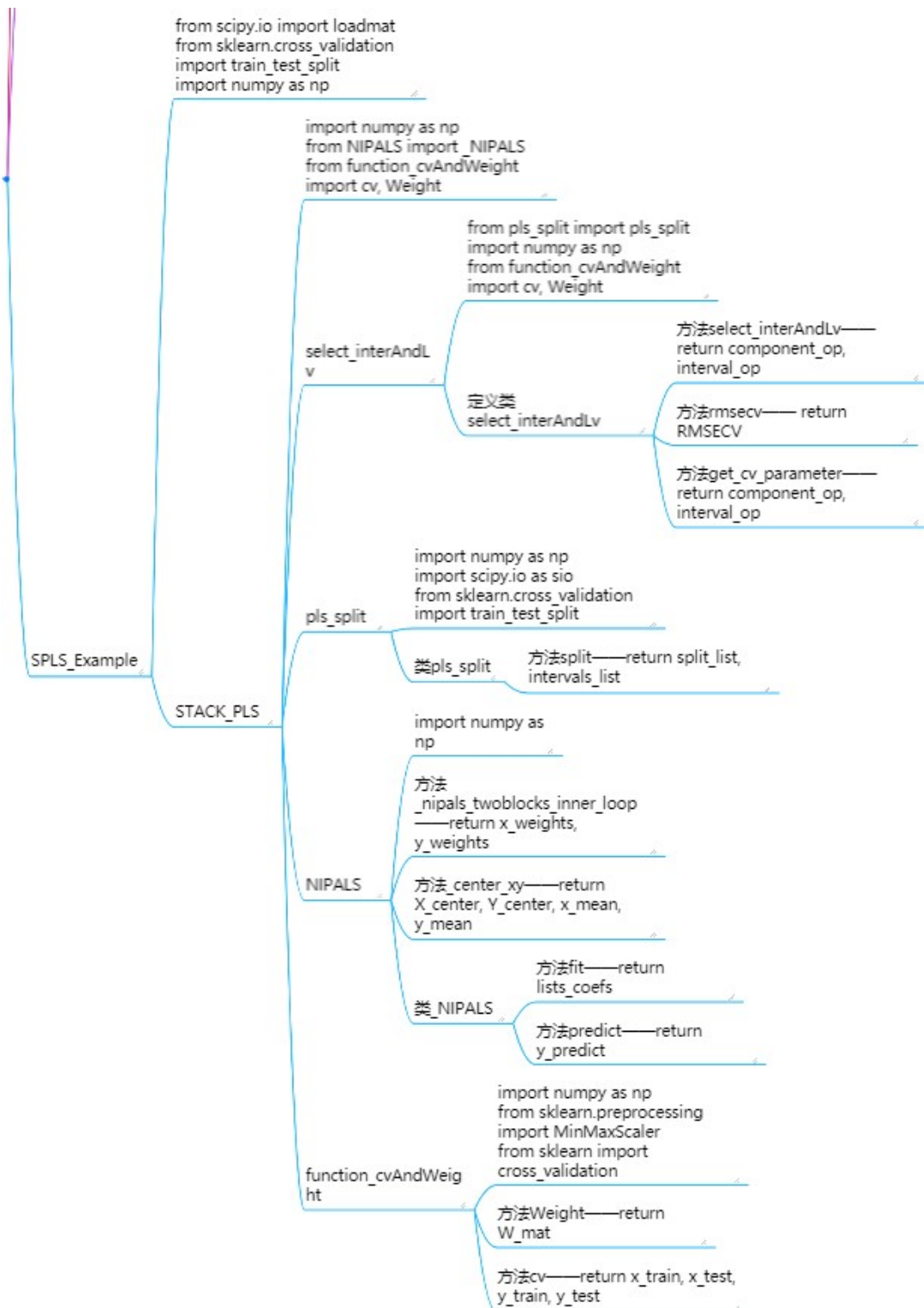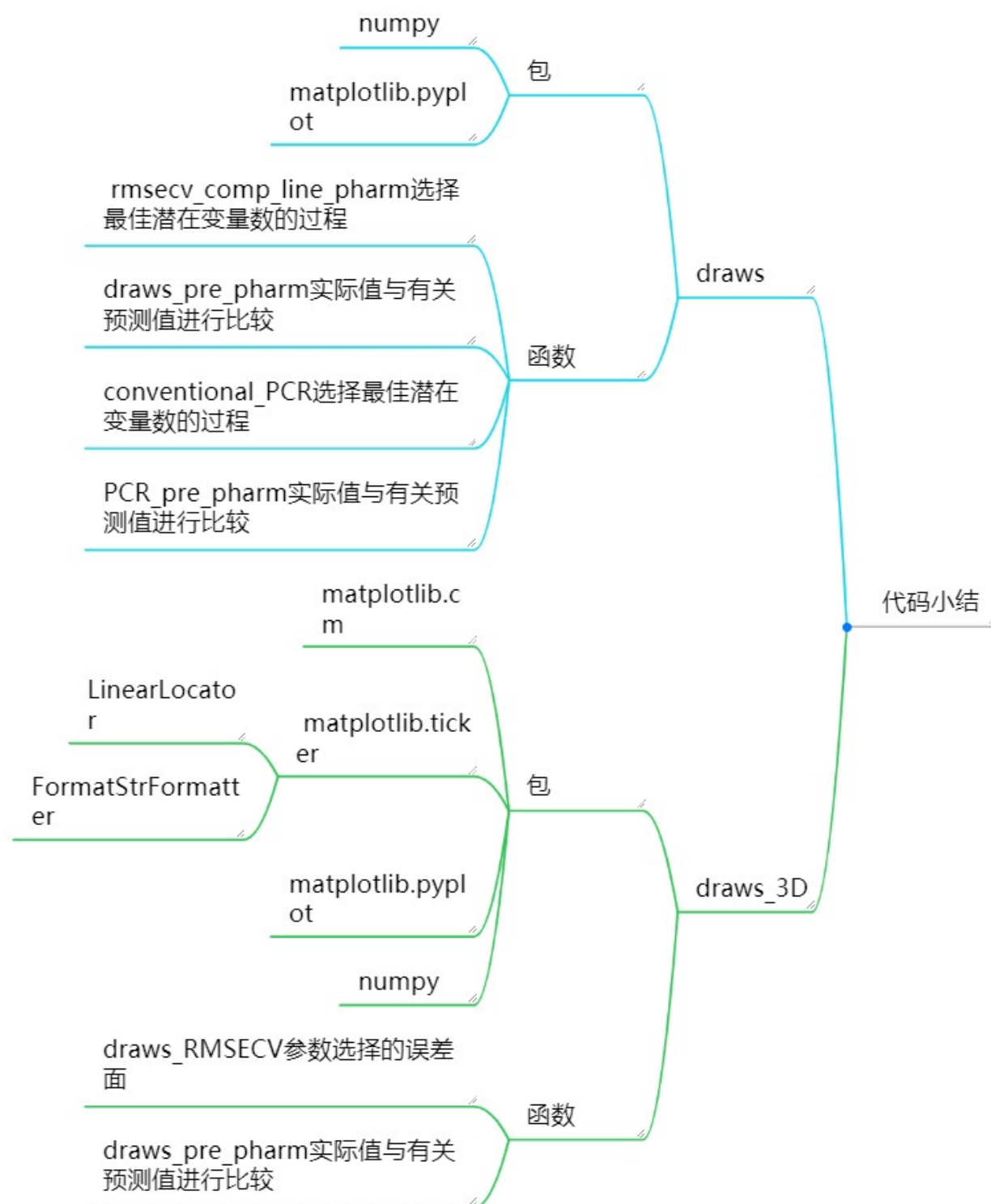# Stack partial least squares

```
from scipy.io import loadmat
from sklearn.cross_validation
import train_test_split
import numpy as np
```

```
import numpy as np
from NIPALS import _NIPALS
from function_cvAndWeight
import cv, Weight
```

```
from pls_split import pls_split
import numpy as np
from function_cvAndWeight
import cv, Weight
```

select_interAndLv

方法select_interAndLv——return component_op, interval_op

定义类 select_interAndLv

方法rmsecv—— return RMSECV

方法get_cv_parameter—— return component_op, interval_op

```
import numpy as np
import scipy.io as sio
from sklearn.cross_validation
import train_test_split
```

pls_split

类pls_split

方法split——return split_list, intervals_list

SPLS_Example

STACK_PLS

```
import numpy as
np
```

方法 _nipals_twoblocks_inner_loop ——return x_weights, y_weights

NIPALS

方法_center_xy——return X_center, Y_center, x_mean, y_mean

方法fit——return lists_coefs

类_NIPALS

方法predict——return y_predict

```
import numpy as np
from sklearn.preprocessing
import MinMaxScaler
from sklearn import
cross_validation
```

function_cvAndWeight

方法Weight——return W_mat

方法cv——return x_train, x_test, y_train, y_test

numpy

matplotlib.pypl
ot

包

rmsecv_comp_line_pharm选择
最佳潜在变量数的过程

draws_pre_pharm实际值与有关
预测值进行比较

conventional_PCR选择最佳潜在
变量数的过程

PCR_pre_pharm实际值与有关预
测值进行比较

函数

draws

matplotlib.c
m

LinearLocato
r

matplotlib.tick
er

FormatStrFormatt
er

包

matplotlib.pypl
ot

numpy

draws_RMSECV参数选择的误差
面

draws_pre_pharm实际值与有关
预测值进行比较

函数

draws_3D

代码小结

Experiments

```python
In [3]: from scipy.io import loadmat
        from sklearn.cross_validation import train_test_split
        from STACK_PLS import Stack_pls
        import numpy as np
        import draws_3D


        if __name__ == '__main__':

            fname = loadmat('E:\Documents\DAY\cornmat.mat')
            X = ('m5','mp5','mp6')
            Y = ('oil', 'starch', 'pro', 'water')
            for a in X: #m5
                rmsecv_list = []

                Y_test = []
                Y_predict = []
                Y_train = []

                for b in Y: #oil
                    x = fname[a]
                    y = fname[b]
                    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=
        0.2, random_state=0)
                    RMSECV = []
                    for i in np.arange(1,16,1):
                        RMSECV_i = []
                        for j in np.arange(2,12,2):
                            demo = Stack_pls(x_train, x_test, y_train, y_test, component
        s=i, folds=5)
                            RMSEP,RMSECV_list,y_predict = demo.stackPls(start=3, end=12,
        intervals=j)

                            RMSECV_i.append(RMSECV_list[0][0])
                        RMSECV.append(RMSECV_i) #15*5

                    rmsecv_list.append(RMSECV)
                    Y_test.append(np.squeeze(y_test).tolist()) #数组降维并转换为list
                    Y_predict.append(np.squeeze(y_predict).tolist())
                    Y_train.append(np.squeeze(y_train).tolist())

                print "RMSECV surface of cross-validation /nfor corndata using SPLS abou
        t the" ,a," instrument"
                draws_3D.draws_RMSECV(Y,rmsecv_list)
                print "The actual value compared with the predicted value about the",a,"
        instrument."
                draws_3D.draws_pre_pharm(Y_test, Y_predict,Y_train)
```
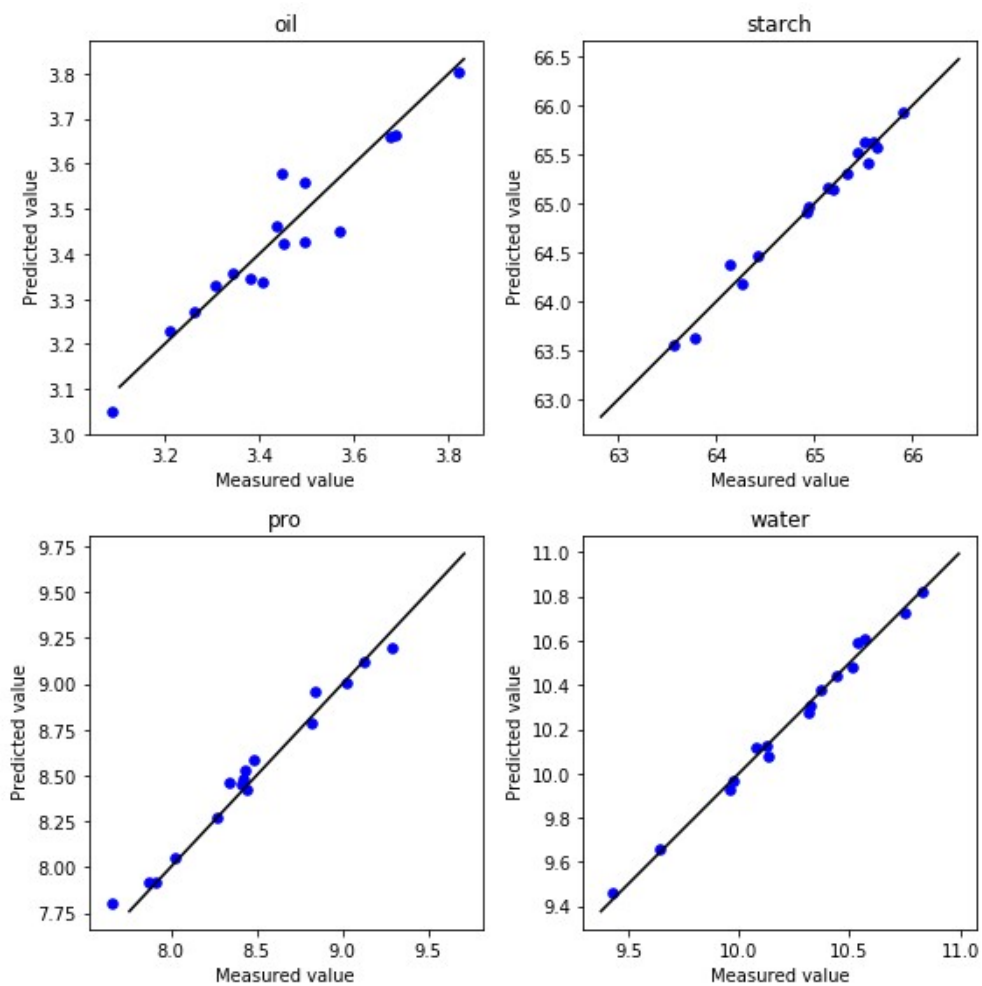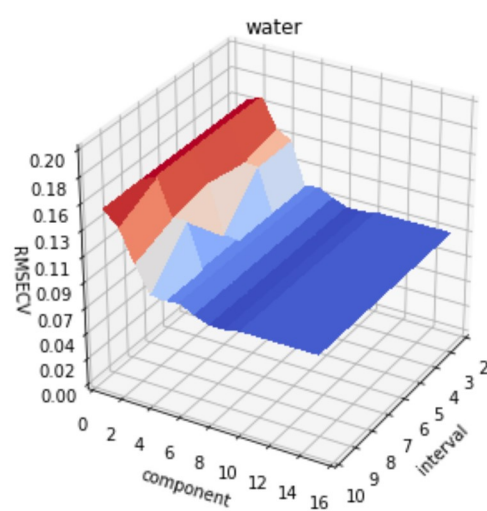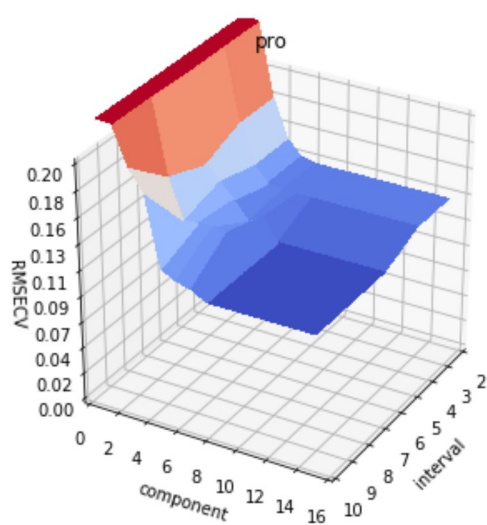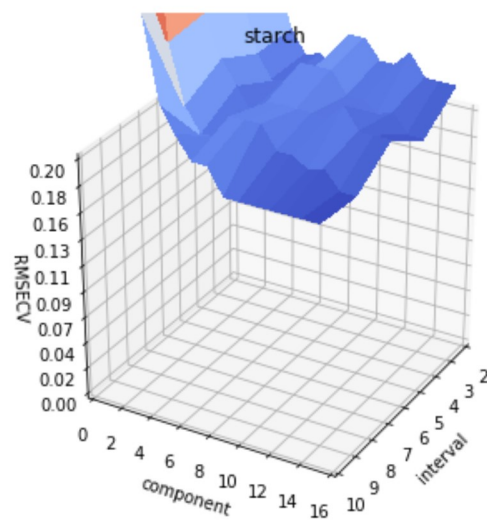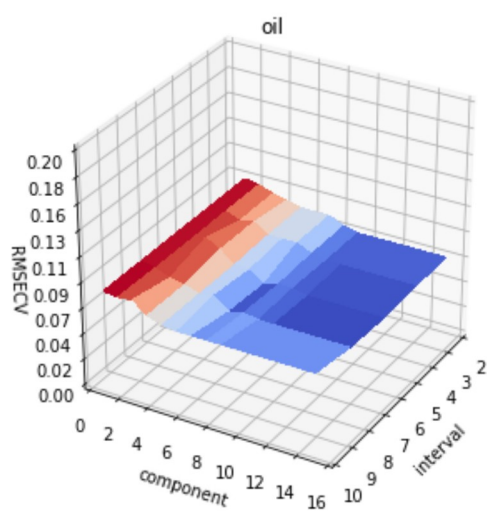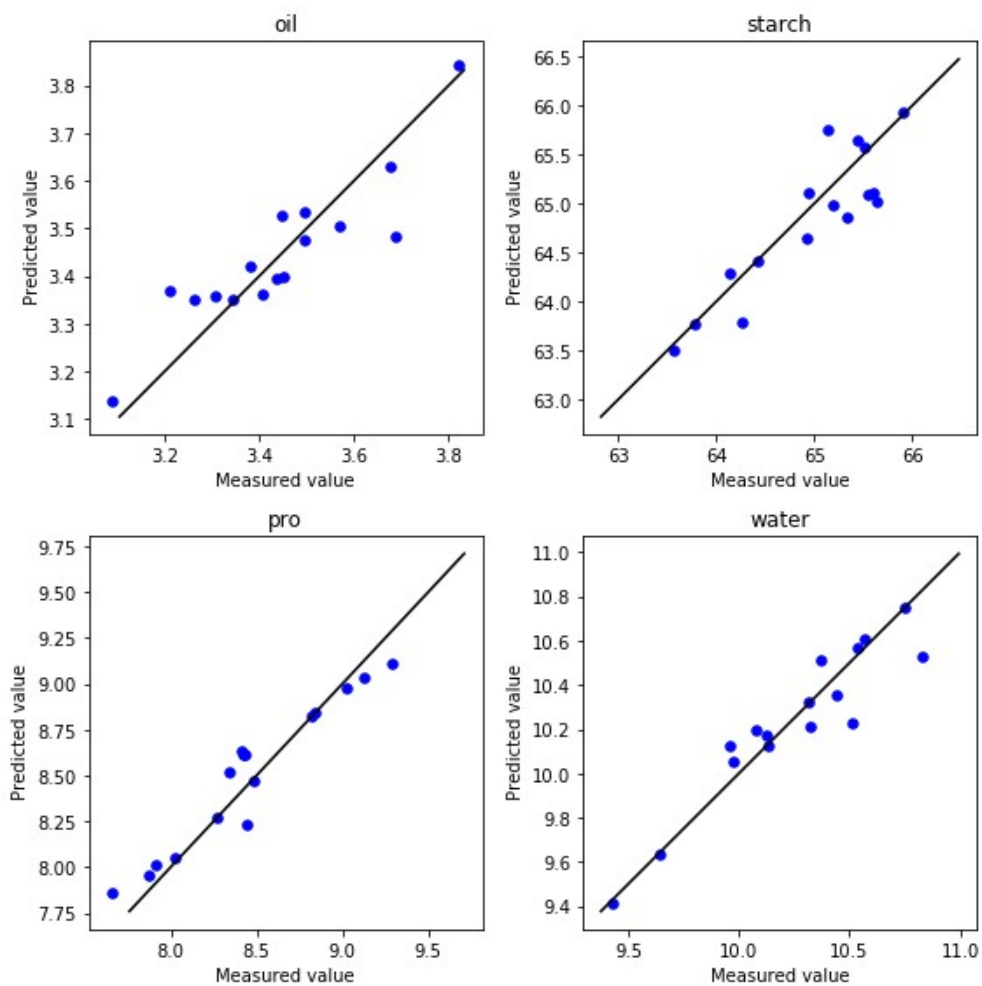
RMSECV surface of cross-validation /nfor corndata using SPLS about the m5 instrument
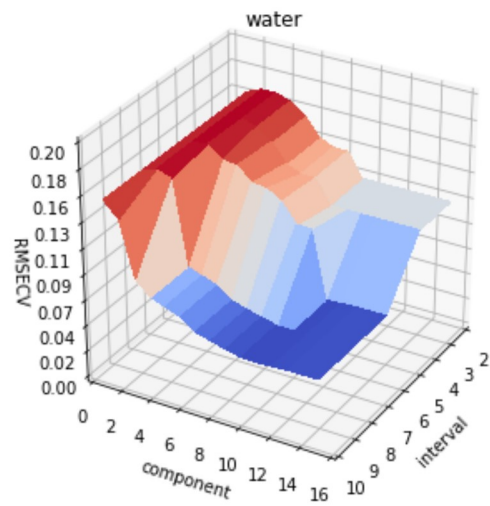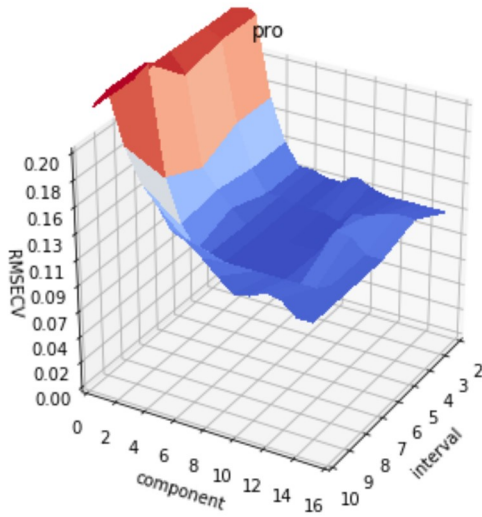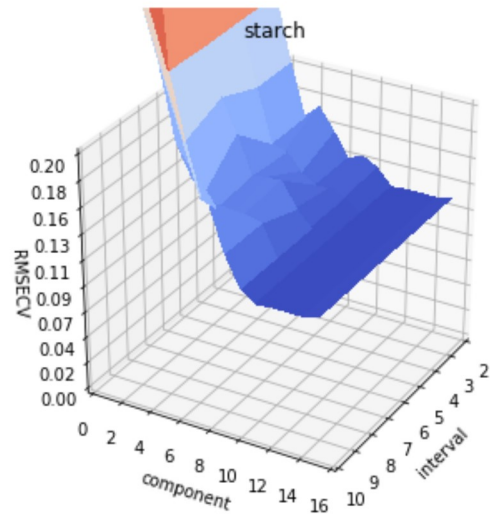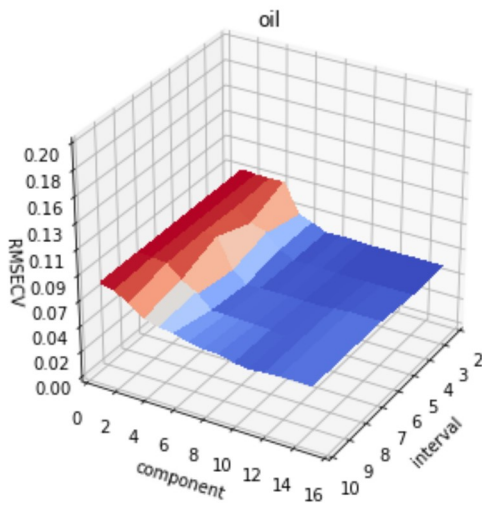
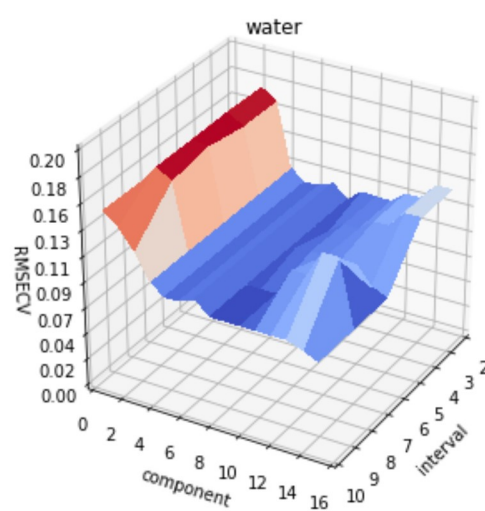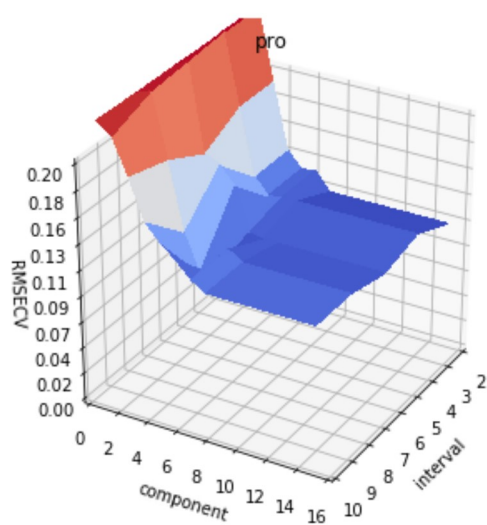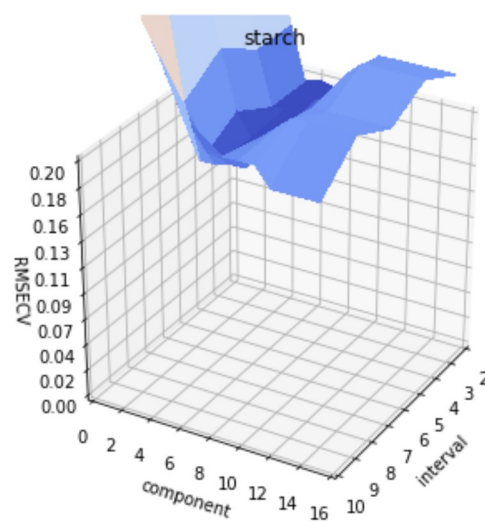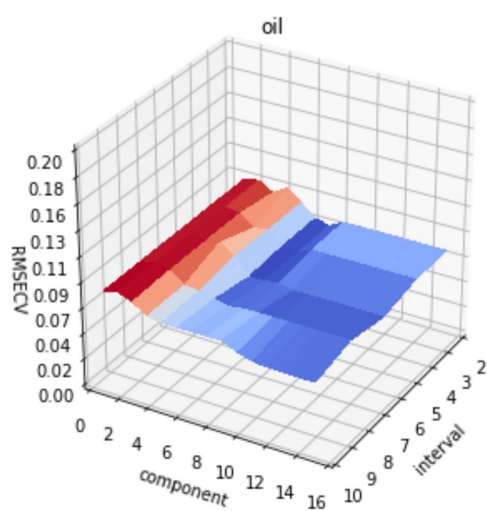The actual value compared with the predicted value about the m5 instrument.

RMSECV surface of cross-validation /nfor corndata using SPLS about the mp5 in
strument

The actual value compared with the predicted value about the mp5 instrument.
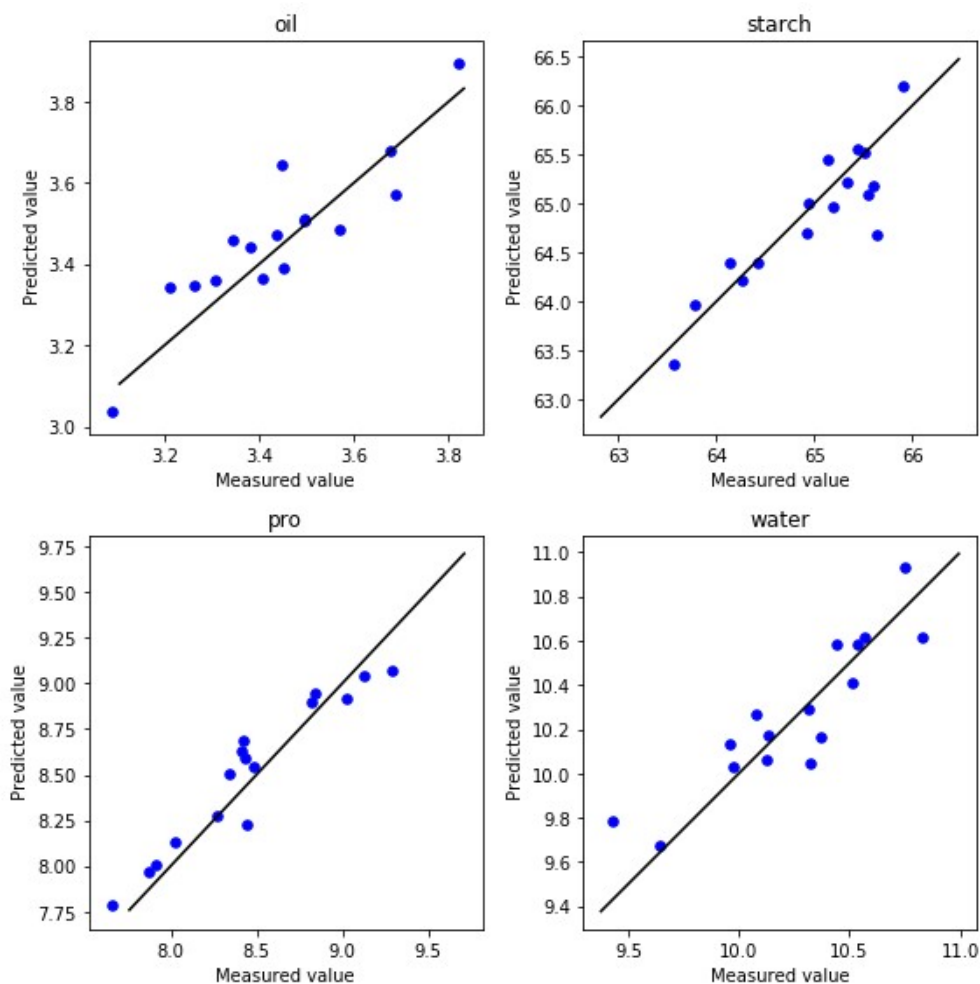
RMSECV surface of cross-validation /nfor corndata using SPLS about the mp6 instrument

The actual value compared with the predicted value about the mp6 instrument.

遇到的问题

- 使用的eclipse暂不支持跑python2版本，但使用jupyter不能导入已经写好的模块，网上查找资料发现是只能导入.py文件，想要在调试好后增加代码把ipynp文件转换。
- 书上用3维图表示的误差，对3维图极其子图画法不熟悉，网上论坛帖子林林总总描述不完善，官网虽然加载慢但是基本找到解决方法，对于传入的坐标数据类型、格式有了基本认识。
- 加循环时老报错，发现是代码没对齐，本质是变量命名不够合理。
- y_tranpredict没能得到，对算法仍不够熟悉
- 跑该数据集吃力，一次运行耗费时间过长，调试时注意加注释，尽量部分整合到一块。