

3次B样条拟合

依赖

- scipy.interpolate 中的 BSpline, splev
- numpy

class **BspFit(dets, c_num=None, alpha=0)**

接口:

- `__init__(self, dets, c_num=None, alpha=0)`
 - dets: ndarray, shape(#bboxes * 4)
 - c_num: 控制点数量, 不指定时为样本点的1/5 和 4中的最大值
 - alpha: 正则项系数, 默认是零。逼近项的系数固定为1
 - 返回: BspFit对象
- `solve(self, alpha=None)`
 - 求解
 - alpha: 正则项系数, 不指定时使用初始化时的正则项系数
 - 返回: tck (节点向量, 控制点, 样条次数组成的3元组)
- `set_alpha(self, alpha)`
 - 设置正则项的权重系数, 也可以在求解的时候或者初始化的时候设置
- `set_c_num(self, n)`
 - 设置控制点数量, 默认为样本点的五分之一, 但不少于4个

待添加的接口

- `set_get_PointList(self, method)`
 - 指定由检测框计算拟合样本点的方式, 目前默认检测框底部中点

其他说明

- 样本点参数化使用的是均匀参数化
- 节点向量均匀选择

示例

- 导入

```
from BspFit import BspFit
```

- 使用:

```
bf = BspFit(dets)
tck = bf.solve(0.0001)    # 0.0001指定的是正则项的系数

# 这里的tck是节点向量, 控制点, 样条次数的三元组

spl = BSpline(*tck)       # 借助scipy.interpolate.BSpline得到样条对象
```

- 示例

```
from BspFit import BspFit
from scipy.interpolate import BSpline
import numpy as np
import pickle
import matplotlib.pyplot as plt

#读取事先准备好的检测框
with open('dets.pkl', 'rb') as f:
    dets = pickle.load(f)

bf = BspFit(dets)

Alpha = [0, 0.001, 0.0001, 0.00001, 0.000001]
Tck = [bf.solve(alpha) for alpha in Alpha]

fig = plt.figure(figsize=(20,20))

ax = fig.add_subplot(321)
ax.invert_yaxis()
ax.plot(np.array(bf._P[:,0]),np.array(bf._P[:,1]))
ax.set_title('raw data')

xx = np.linspace(0,1,1000)

for i in range(len(Alpha)):
    ax = fig.add_subplot(322 + i)
    spl = BSpline(*Tck[i])
    ax.plot(spl(xx)[:,0],spl(xx)[:,1])
    ax.invert_yaxis()
    ax.set_title('alpha = %g'%(Alpha[i]))

plt.show()
```

