Python 离散小波分析

Dezeming Family

2022 年 5 月 15 日

DezemingFamily 系列书和小册子因为是电子书，所以可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列书，可以从我们的网站 [https://dezeming.top/] 找到最新版。对书的内容建议和出现的错误欢迎在网站留言。

目录

一 离散小波变换函数 **1**

1 1 dwt 函数 1

1 2 wavedec 函数 1

二 离散小波变换测试 **1**

2 1 数据准备 1

2 2 离散小波变换 2

三 离散小波逆变换 **3**

3 1 函数介绍 3

3 2 程序测试 3

参考文献 **4**

一　离散小波变换函数

1 1　dwt 函数

我们可以通过下面的函数来查看有哪些小波可以用于离散小波变换：

```
wavlist = pywt.wavelist(kind='discrete')
print(wavlist)
```

得到：

```
['bior1.1', 'bior1.3', 'bior1.5', 'bior2.2', 'bior2.4', 'bior2.6', '
    bior2.8', 'bior3.1', 'bior3.3', 'bior3.5', 'bior3.7', 'bior3.9', '
    bior4.4', 'bior5.5', 'bior6.8', 'coif1', 'coif2', 'coif3', 'coif4', '
    coif5', 'coif6', 'coif7', 'coif8', 'coif9', 'coif10', 'coif11', '
    coif12', 'coif13', 'coif14', 'coif15', 'coif16', 'coif17', 'db1', '
    db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'db11'
    , 'db12', 'db13', 'db14', 'db15', 'db16', 'db17', 'db18', 'db19', '
    db20', 'db21', 'db22', 'db23', 'db24', 'db25', 'db26', 'db27', 'db28'
    , 'db29', 'db30', 'db31', 'db32', 'db33', 'db34', 'db35', 'db36', '
    db37', 'db38', 'dmey', 'haar', 'rbio1.1', 'rbio1.3', 'rbio1.5', '
    rbio2.2', 'rbio2.4', 'rbio2.6', 'rbio2.8', 'rbio3.1', 'rbio3.3', '
    rbio3.5', 'rbio3.7', 'rbio3.9', 'rbio4.4', 'rbio5.5', 'rbio6.8', '
    sym2', 'sym3', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'sym9', 'sym10
    ', 'sym11', 'sym12', 'sym13', 'sym14', 'sym15', 'sym16', 'sym17', '
    sym18', 'sym19', 'sym20']
```

一共 106 个。

pywt.dwt 用于离散小波变换，参数列表为：

```
[cA,cD] = dwt(data, wavelet, mode='symmetric', axis=-1)
```

wavelet 表示小波名称。mode 表示数据扩展的方式 [4]：因为计算机处理的数据是有限位数据，采用级联滤波器组算法做离散小波变换之前，需要对数据进行一些额外的外扩。比如填充 0，即 mode='zero'；还比如根据数组的第一位和最后一位的值分别往两边用常量外扩，即 mode='constant'。symmetric 表示对称外扩。我们之前介绍过补 0 的方法，但是注意，这可能会引入一个跳变的过程，给小波分析带来不利影响，因此会采样其他的填补方式，例如 symmetric。

1 2　wavedec 函数

如果有时候想一次多变换几个 level，则可以使用 wavedec 函数，该函数参数列表是：

```
[cA_n, cD_n, cD_n-1, ..., cD2, cD1] = pywt.wavedec(data, wavelet, mode='
    symmetric', level=None, axis=-1)
```

level 就是进行分解的次数，cD_n 中的 n 等于 level 值。

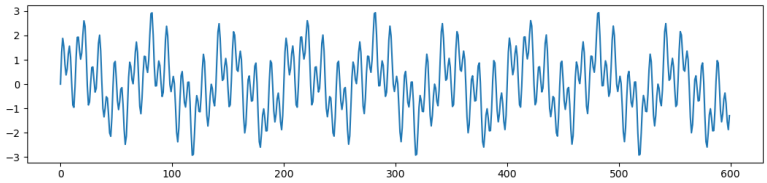
二　离散小波变换测试

2 1　数据准备

我们采用的数据是：

```
data = []
for i in range(600):
    aa = np.sin(0.3 * np.pi * i) + np.sin(0.13 * np.pi * i) + np.sin(0.03 *
        np.pi * i) + 0.2 * np.sin(0.01 * np.pi * i) + 0.01 * np.sin(0.001 *
        np.pi * i)
    data.append(aa)
plt.subplot(511)
plt.plot(data)
```

生成图像为：

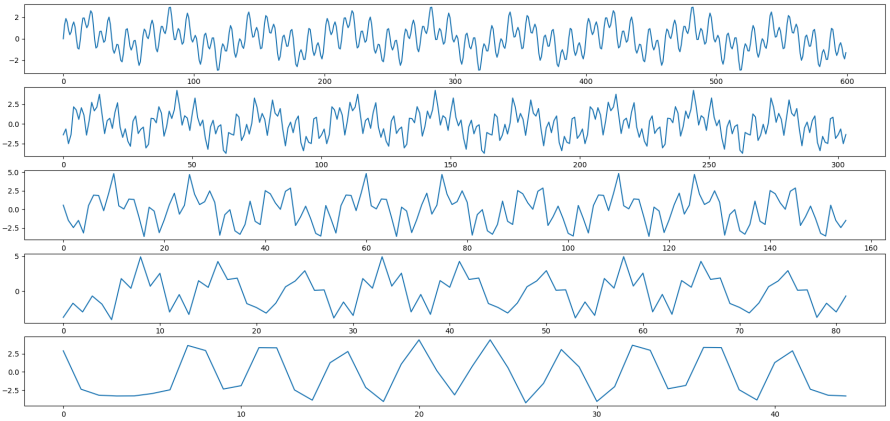


2 2 离散小波变换

我们先用 dwt 函数，通过循环的方式来分解四次：

```
wavename = 'db5'
cA = []
cA.append(data)
cD = []
for i in range (0,4):
    cA1, cD1 = pywt.dwt(cA[i], wavename, mode='periodic')
    cA.append(cA1)
    cD.append(cD1)
    imageIndex = '51' + str(i+2)
    plt.subplot(int(imageIndex))
    x1 = range(len(cA1))
    plt.plot(x1, cA1)
plt.show()
```

得到结果为：



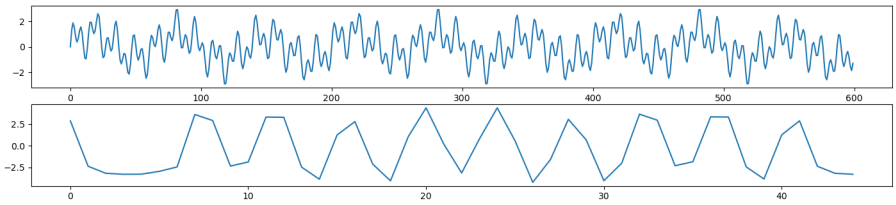
可以看到下标不断减半，而随着变换的进行，数据的大致轮廓被提取了出来，也可以可视化 cD 来查看细节部分，这里不再介绍。

也可以使用 wavedec 来进行变换：

```
wavename = 'db5'
```

```
[cA_4, cD_4, cD_3, cD_2, cD1] = pywt.wavedec(data, wavename, level=4, mode='
    periodic')
x1 = range(len(cA_4))
plt.subplot(512)
plt.plot(x1, cA_4)
plt.show()
```

可以看到得到的结果是一样的：



三　离散小波逆变换

3 1　函数介绍

pywt 的 idwt 函数参数列表如下：

```
rec = def idwt(cA, cD, wavelet, mode='symmetric', axis=-1)
```

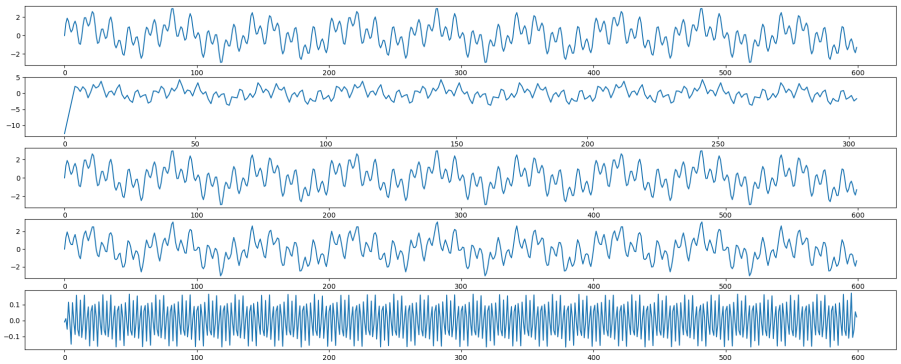
定义都很清楚明确，这里就不解释了。

3 2　程序测试

测试代码如下：

```
(cA, cD) = pywt.dwt(data, 'db5', mode='smooth')
plt.subplot(512)
plt.plot(cA)
data2 = pywt.idwt(cA, cD, 'db5', mode='smooth')
plt.subplot(513)
plt.plot(data2)
cA_2 = pywt.idwt(cA, None, 'db5', mode='smooth')
plt.subplot(514)
plt.plot(cA_2)
cD_2 = pywt.idwt(None, cD, 'db5', mode='smooth')
plt.subplot(515)
plt.plot(cD_2)
plt.show()
```

'smooth' 模式表示根据信号两端的导数来进行扩张，进行 idwt 以后的信号恢复到原信号长度。我们也可以单独对 cA 或者 cD 做离散小波逆变换。图示如下：



参考文献

[1] https://github.com/PyWavelets/pywt

[2] https://pywavelets.readthedocs.io/en/latest/

[3] https://pywavelets.readthedocs.io/en/latest/ref/index.html

[4] https://pywavelets.readthedocs.io/en/latest/ref/signal-extension-modes.html#ref-modes