6月11日工作随笔

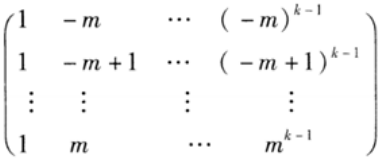
1、实现了测试数据产生函数：test\_data\_gen(num\_pi,num\_ex,num\_gap,num\_bk)，num\_pi可以设置测试数据长度，num\_ex可以设置异常数据的个数，num\_gap可以设置块缺失数据的个数，num\_bk可以设置单个缺失数据的个数。

2、与邹舒畅讨论了对时间序列清洗的步骤，确定先进行异常值滤除，再进行缺失值插值。

3、对data\_smooth程序（邹舒畅已实现）中使用的Savitzky-Golay平滑滤波方法进行了学习，其主要思想是了：利用n个多项式k元线性方程组对窗口为n的数据点进行最小二乘求解。

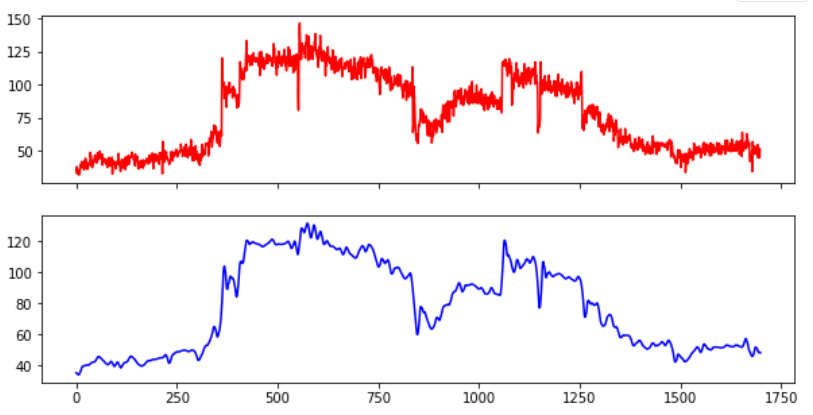
预测值：，式中Y为n个时间序列值

矩阵

矩阵X=，式中2m+1=n;

程序通过设置窗口n来改变滤除异常值的效果。

1. 学习了Yahoo大规模时间序列数据自动异常检测架构，文中提出了利用多种异常检测方法并用，并通过AdaBoost模型对多种结果进行组合加权表决产生。
2. 对小波变换库Pywavelet进行了简单的使用，效果如下：



6月12日工作随笔

1. 搜索了3D结构线条模型制作工具：Maya和Houdini，同时还尝试了Wings3D
2. 研究了小波变换库的具体功能：

创建一个小波对象wavelet = pywt.Wavelet('mode+k')

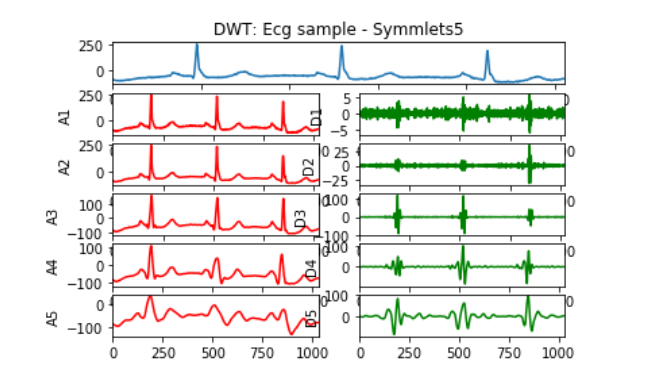
小波族mode有：['haar', 'db', 'sym', 'coif', 'bior', 'rbio', 'dmey']可选

每一个小波族又可以通过pywt.wavelist(‘mode’)来选择具体的小波名称k

使用dwt()函数用于完成一阶、1D的离散小波变换：

(cA, cD) = dwt(data, wavelet, mode=’sym’)

cA为近似分量，cD为细节分量，要想滤除干扰只需要保留近似分量。将数据序列进行小波分解，每一层分解的结果是上次分解得到的低频信号再分解成低频和高频两个部分。如此进过N层分解后源信号X被分解为：X = D1 + D2 + ... + DN + AN 其中D1,D2,...,DN分别为第一层、第二层到等N层分解得到的高频信号，AN为第N层分解得到的低频信号。



恢复数据Data=idwt(cA,cD,wavelet,’sym’)

多阶次离散小波分解wavedec(data,wavelet)

coeffs = pywt.wavedec(data, db4,’sym’,level=2)#

# 高频系数置零

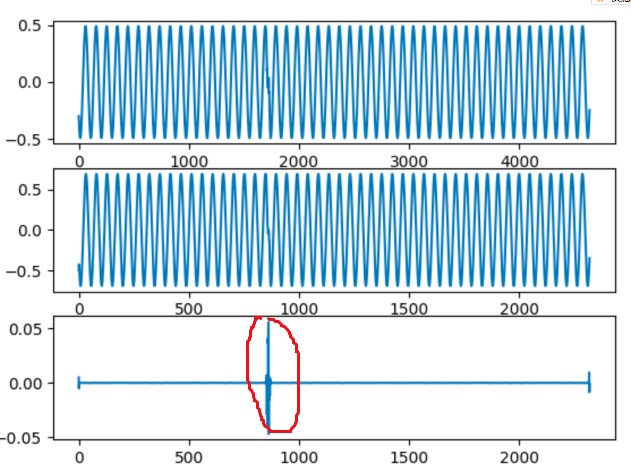
coeffs[len(coeffs)-1] \*= 0

coeffs[len(coeffs)-2] \*= 0

# 重构

meta = pywt.waverec(coeffs, db4)

小波变换最重要的特点：可以定位到异常数据的位置

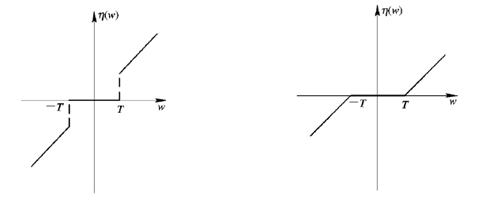


一般来说， 一维信号的降噪过程可以分为 3个步骤

      信号的小波分解。选择一个小波并确定一个小波分解的层次N，然后对信号进行N层小波分解计算。

      小波分解高频系数的阈值量化。对第1层到第N层的每一层高频系数（三个方向）， 选择一个阈值进行阈值量化处理．

     这一步是最关键的一步，主要体现在阈值的选择与量化处理的过程，在每层阈值的选择上matlab提供了很多自适应的方法， 这里不一一介绍，量化处理方法主要有硬阈值量化与软阈值量化。下图是二者的区别：



     上面左图是硬阈值量化，右图是软阈值量化。采用两种不同的方法，达到的效果是，硬阈值方法可以很好地保留信号边缘等局部特征，软阈值处理相对要平滑，但会造成边缘模糊等失真现象。

      信号的小波重构。根据小波分解的第 N层的低频系数和经过量化处理后的第1层到第N 层的高频系数，进行信号的小波重构。

小波阀值去噪的基本问题包括三个方面：小波基的选择，阀值的选择，阀值函数的选择。  
（1）****小波基的选择****：通常我们希望所选取的小波满足以下条件：正交性、高消失矩、紧支性、对称性或反对称性。但事实上具有上述性质的小波是不可能存在的，因为小波是对称或反对称的只有Haar小波，并且高消失矩与紧支性是一对矛盾，所以在应用的时候一般选取具有紧支的小波以及根据信号的特征来选取较为合适的小波。  
（2）****阀值的选择****：直接影响去噪效果的一个重要因素就是阀值的选取，不同的阀值选取将有不同的去噪效果。

（3）****阀值函数的选择****：阀值函数是修正小波系数的规则，不同的反之函数体现了不同的处理小波系数的策略。最常用的阀值函数有两种：一种是硬阀值函数，另一种是软阀值函数。还有一种介于软、硬阀值函数之间的Garrote函数。  
另外，对于去噪效果好坏的评价，常用信号的信噪比（SNR）与估计信号同原始信号的均方根误差（RMSE）来判断。

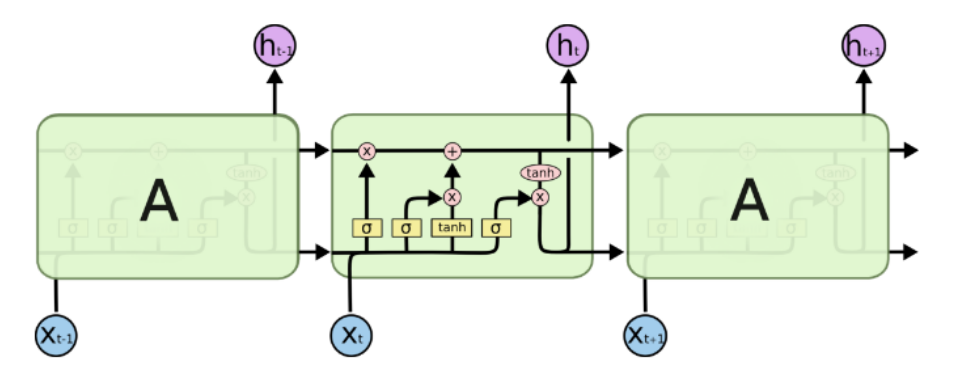
pywt.threshold（data，value，mode ='soft'，substitute = 0 ）

# data：数据集，value：阈值，mode：比较模式默认soft，substitute：替代值，默认0，float类型

mode : {‘soft’, ‘hard’, ‘greater’, ‘less’}

6月13日工作随笔

由于RNN也有梯度消失的问题，因此很难处理长序列的数据，大牛们对RNN做了改进，得到了RNN的特例LSTM（Long Short-Term Memory），它可以避免常规RNN的梯度消失，因此在工业界得到了广泛的应用。其结构如下：



在每个序列索引位置t时刻向前传播的除了和RNN一样的隐藏状态h(t)，还多了另一个隐藏状态，如图中上面的长横线。这个隐藏状态我们一般称为细胞状态(Cell State)，记为C(t)。

1、TensorFlow中可以通过来BasicRNNCell/BasicLSTMCell/GRUCell/RNNCell/LSTMCell来定义，我们使用了：BasicLSTMCell

cell=tf.nn.rnn\_cell.BasicLSTMCell(rnn\_unit,state\_is\_tuple=True)#定义一个cell

2、然后要对cell的state状态进行初始化：

state=cell.zero\_state(batch\_size,dtype=tf.float32)#初始化全零

3、然后我们调用 dynamic\_rnn() 来让我们构建好的网络运行起来

output\_rnn,states=tf.nn.dynamic\_rnn(cell,input\_rnn,initial\_state=state,dtype=tf.float32, time\_major=False)

4当然进入cell之前肯定有输入隐层：

input\_rnn=tf.matmul(input\_hi,w\_in)+b\_in #input\_rnn作为BasicLSTMCell的一个输入

5 Cell的output\_rnn给输出隐层来做预测：

Predict=tf.matmul(output,w\_out)+b\_out

整个程序最主要的是数据有几次维度的变化，对原始数据变换成一个个batch的三维数据batch\_size\*time\_step\*input\_size(60\*20\*6),但是进入输入隐层又要转换为二维数据,因为我们定义的输入隐层的权重W为input\_size\*rnn\_unit(6\*20)，输入隐层将计算后的数据给cell时又要转换成三维数据batch\_size\*time\_step\*rnn\_unit(60\*20\*10)，cell的输出给输出隐层做预测时又要将数据转换为二维数据，因为我们的输出隐层的权重W为rnn\_unit\*one(10\*1)。

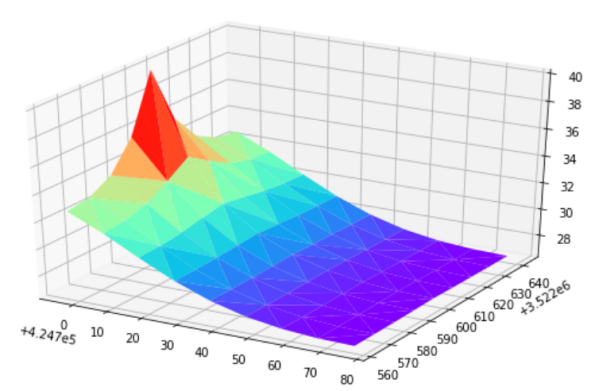
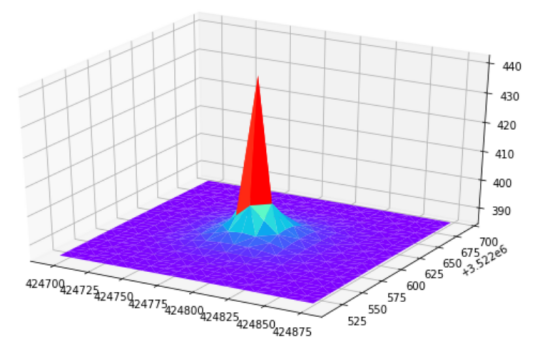
Cell中rnn\_unit的个数太多会 overfitting，太少效果不好，可以调节看变化。

time\_step比较难理解（时序持续长度为20，即每做一次预测，需要先输入20次数据），其实在做文本预测时就是一句话的长度，所以三维数据为：batch\_size\*sequence\_lenth\*Input\_Dimension

可以参考：<https://www.jianshu.com/p/b9cd38804ac6>

6月14日工作随笔

1. 根据输入的XYZ三列数据生成三维图，由于Z与XY没有函数关系式，Python中只有ax.plot\_trisurf(X, Y, Z, cmap='rainbow')函数画三维图,效果如下：



1. 与邹舒畅商量了改善LSTM预测模型的方法：
2. 改变参数time\_step为288，但是这个可能会导致模型训练很慢，暂时不考虑
3. 增加rnn\_unit的个数为128个
4. 数据需要预处理：清洗（使用清洗后的数据，训练次数为200次后loss小于0.01）
5. 特征选择，剔除掉可能对预测结果用处不大的特征
6. 使用了随机森林回归模型对特征（错位1天的前值、温度、天气、季节、DW、是否节假日）进行了选择，在进行特征选择时，对数据进行了不同长度的测试，结果如下：

1天数据量特征相关性检测：[(0.919, 'ex\_Value'), (0.101, 'Weather'), (0.085, 'Temperature'), (0.057, 'DW'), (-0.005, 'Season'), (-0.02, 'IsHoliday')]

7天数据量特征相关性检测：[(0.888, 'ex\_Value'), (-0.001, 'Season'), (-0.002, 'Temperature'), (-0.003, 'Weather'), (-0.004, 'DW'), (-0.006, 'IsHoliday')]

30天数据量特征相关性检测：[(0.716, 'ex\_Value'), (0.13, 'Temperature'), (0.124, 'Weather'), (0.023, 'DW'), (0.02, 'IsHoliday'), (-0.001, 'Season')]

180天数据量特征相关性检测：[(0.736, 'ex\_Value'), (0.162, 'Temperature'), (0.103, 'Season'), (0.031, 'Weather'), (0.009, 'IsHoliday'), (0.001, 'DW')]

360天数据量特征相关性检测：[(0.747, 'ex\_Value'), (0.102, 'Temperature'), (0.037, 'Season'), (0.021, 'Weather'), (0.005, 'IsHoliday'), (0.001, 'DW')]

从检测结果可以看出：错位1天的前值、温度、天气、季节这四个特征与我们的Y值相关性较高。

要学习的内容：数据特征提取、卷积分类、LSTM回归、对抗生成神经网络、遗传算法、强化学习

6月15日工作随笔

1. 对GAN生成对抗网络模型的理论进行了学习：G是一个生成图片的网络，它接收一个随机的噪声z，通过这个噪声生成图片，记做G(z)。D是一个判别器，判别一张图片是不是“真实的”。当G和D两者都是多层感知机的时候，即称为对抗网络（adversarial nets），该算法的主要难度在于训练，并且两个网络不能同时训练。后面会主要结合GAN\_smooth代码进行学习。

2、对rnn\_unit为128的LSTM预测模型的训练，总共进行10000次迭代，预计到还有两天能训练结束。

3、对自己一周的工作进行了总结，完成了工作周报。

4、通过下午开会学习了智慧水务的相关知识，对最低服务水头、变压供水、全局最不利点、漏损与控压的关系等知识进行了学习，对智慧水务有了进一步的认识。

6月20日工作随笔

1. 对输配水自动化半实物仿真模型的算法部分内容主要从以下五个方面进行了填充：

一、考核指标：

实现对半实物仿真模型的数据进行优化处理。

实现对半实物仿真模型模拟的问题进行预测。

实现对半实物仿真模型的智能调度和智能控制。

二、预期取得的成果：

通过各种AI算法对半实物仿真模型模拟的问题实现智能预测、智能调度、智能处置和智能控制等功能。

三、主要研究内容：

通过AI算法实现智能预测和智能处置半实物仿真模型模拟的以下问题：

（1）远端供水压力最小点问题

（2）近端压力过高问题

（3）水管太细导致用户用水困难

（4）水源端泵的压力过高问题

（5）管网不受控的漏水问题

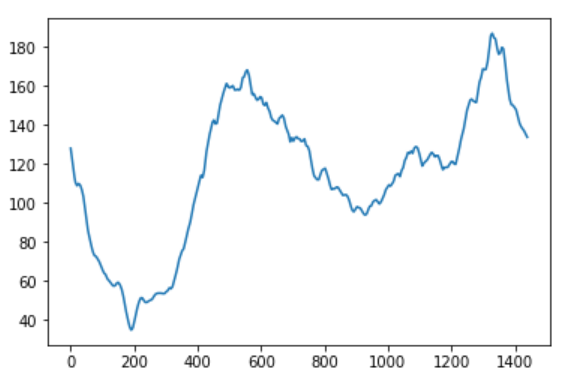
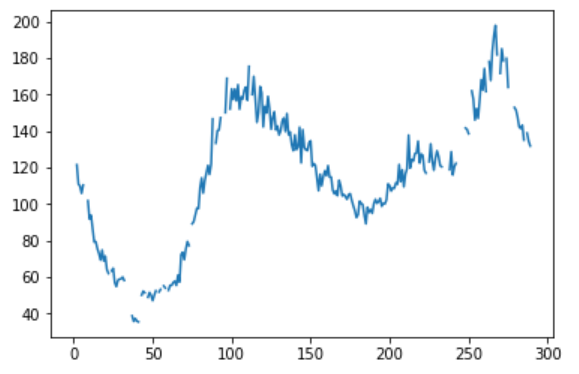
四、技术特点与创新性：

使用了AI算法的理念对当前水务管理中的问题进行解决，同时又通过半实物仿真模型实现对AI算法的可靠性进行了验证和迭代更新。

五、关键技术：传统调度算法滞后的解决方案，本项目利用神经网络、深度学习、遗传算法和模式识别等AI技术实现智能预测、智能调度、智能处置、智能控制和智能服务全方位的水务管理。

技术难点：神经网络、深度学习等AI算法开发需要海量的数据进行模型训练和测试。

2、压力分布3D仿真模型所需的数据处理，由于从网站导出的数据为5分钟间隔的289个数据，对数据进行数据清洗后采用采样的方法将原数据变为1分钟间隔的1440个数据。



1. 学习了ValveCare：减压阀控制策略研究的内容，其主要进行研究的内容如下：
2. 对得到的压力数据和流量数据进行清洗操作；
3. 根据dP（减压阀后压力P2-最不利点压力P3）和减压阀前入口流量Q进行了如下策略分析：

（1）固定阀后压力P2的方式

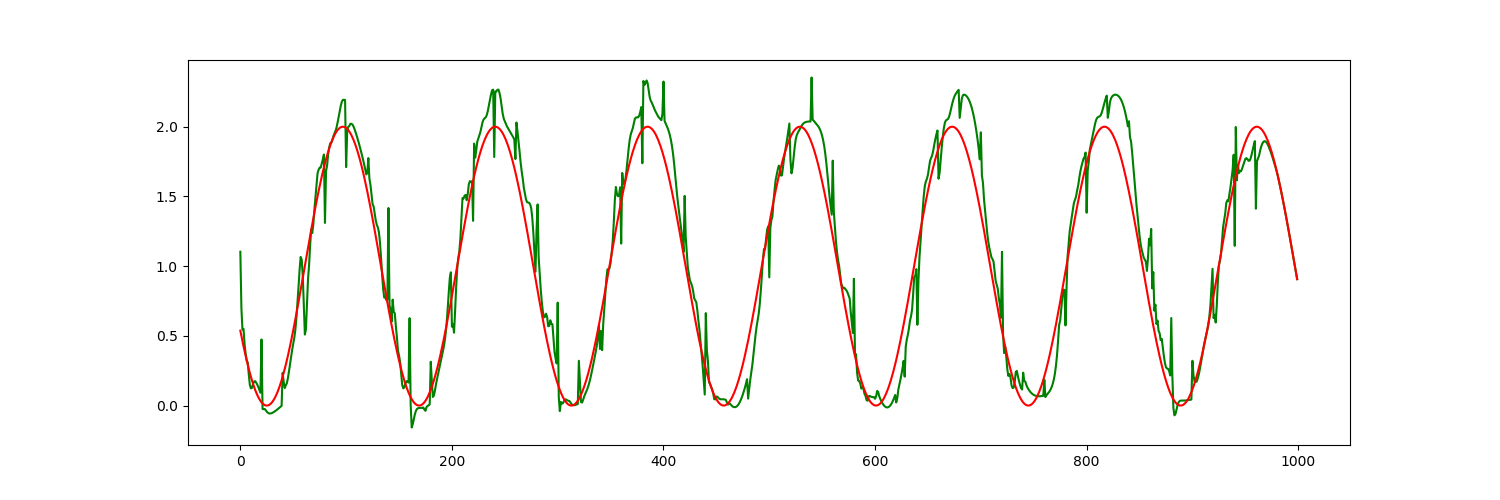
（2）多点压力分段控制的方式（即：Kmeans后的dP与分段时间进行控制）；

（3）dP-Q进行最小二乘拟合设定P2的方式；

（4）dP-Q进行BP神经网络拟合，并通过最不利点压力P3回传进行在线更新的方式。

6月21日工作随笔

1. 对ValveCare（减压阀控制策略研究）的逻辑结构进行了Visio画图表示。
2. 对LSTM模型用正弦波信号进行了测试，主要是用产生的正弦波序列替代了之前的的数据列，用训练后的模型参数进行了预测测试，结果如下：



与邹舒畅讨论出现毛刺的原因是模型中一些其他非连续特征值导致的。下一步决定改造预测模型进行测试。

3、讨论了一个简单的非线性预测神经网络中遇到的问题，图1中可看出红色曲线不能很好的拟合原数据，原因是最后输出层采用了tanh激励函数对预测结果输出导致的截断，修改后拟合结果如图2所示，所以提示我们在设计模型时根据数据选择合适的激励函数。

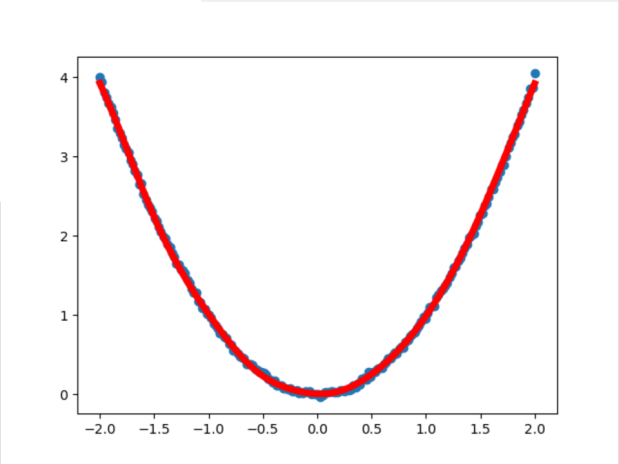
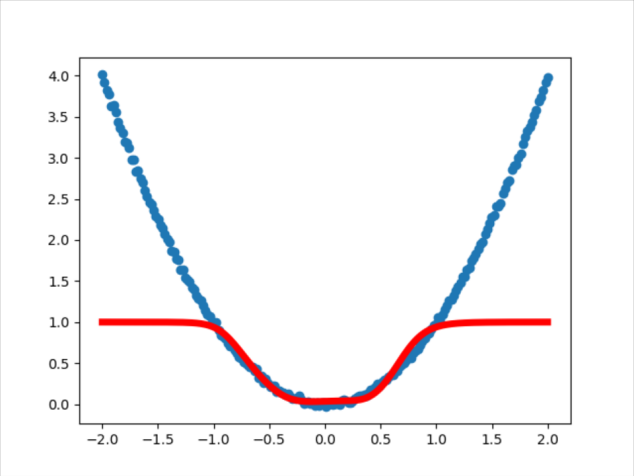


图1 图2

6月22日工作随笔

1. 将LSTM预测算法进行了改造，训练和测试均为正弦波数据，降低特征维度后，预测数据还是带有毛刺，计划改造网络的结构为：rnn\_unit的个数为10，input\_size为2。
2. 学习了三种异常值检测方法one-class SVM算法、Robust covariance和局部异常因子LOF算法（作为技术储备）。

一、One-Class SVM：

One-Class SVM，它有能力捕获数据集的形状,因此对于强非高斯数据有更加优秀的效果。它的训练集不能包含异常样本，否则的话，可能在训练时影响边界的选取。

二、Robust covariance：

基于协方差的稳健估计，假设数据是高斯分布的，那么在这样的案例中执行效果将优于One-Class SVM。

1. LOF：

通过比较每个点p和其邻域点的密度来判断该点是否为异常点，如果点p的密度越低，越可能被认定是异常点。至于密度，是通过点之间的距离来计算的，点之间距离越远，密度越低，距离越近，密度越高，完全符合我们的理解。而且，因为lof对密度的是通过点的第k邻域来计算，而不是全局计算，因此得名为“局部”异常因子。

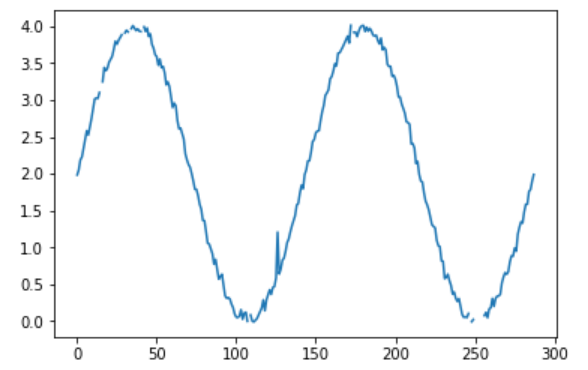
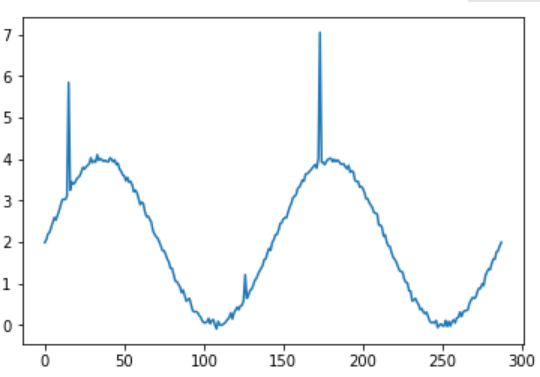


图1 原始数据 图2 SVM

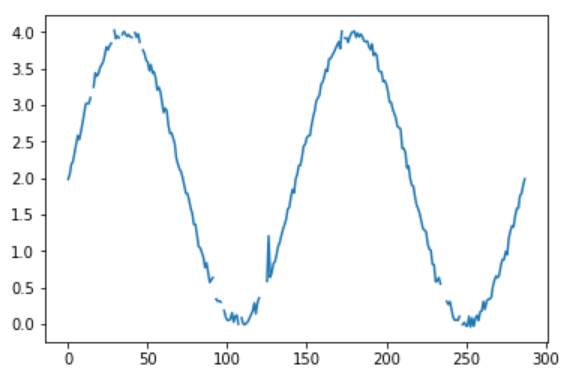
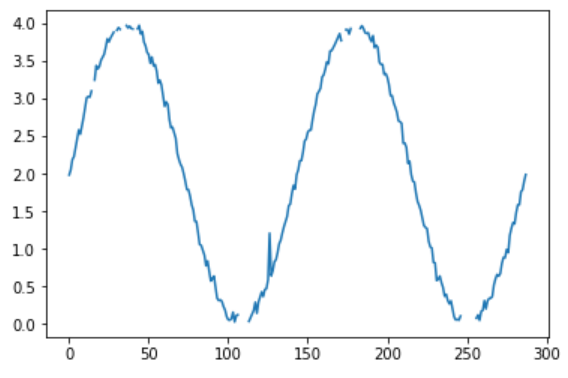


图3 Robust covariance 图4 LOF

1. 和叶玲梳理了MeterCare、ValveCare、PumpCare、PatternCare的逻辑关系图初稿。

6月25~6月26日工作随笔

1. 开会对输配水自动化半实物仿真模型的管网CAD原理图进行了原理和功能梳理，主要模拟了细、破、近、远、殇、爆六种管网问题。
2. 根据输配水自动化中梳理的管网问题，对Care系列逻辑规划图进行了功能梳理，主要是PatternCare、MeterCare、ValveCare、PumpCare、DMACare，这些整体合成了一个PressureCare：全网压力管理。
3. 在水务管理中最重要的是能耗的管理和漏损的管理，降低管网的输送压力可以有效的降低水务管理中的能耗和漏损（爆管情况）。

6月27日工作随笔

1. 主要进行了Care系列的逻辑规划图的Visio绘制，使自己对整个Care系列有了一个较为整体的理解。
2. 利用早上上班前的时间学习了熵理论：

动态符号化时间序列，动态符号化是指对长度为的时间序列取长度为的矩形窗，对窗内序列取均值， 窗内大于均值的点置于1，小于均值的点置为0，这样将一长度为的非“0，1”序列变换为个长度为的“0，1”符号序列，然后对*N-L*+1个序列分别计算其复杂度或熵，从而得到复杂度随时间的动态变化。

（1）信息熵是从平均意义上描述信源的总体特征的，它表示信源输出的平均信息量，它表征信源的平均不确定性

（2）近似熵（Approximate Entropy ）主要的是从衡量时间序列复杂性的角度来度量信号中产生新模式的概率大小，产生新模式的概率越大，序列的复杂性越大，相应的近似熵也越大。

样本熵（SampleEn）在算法上相对于近似熵算法的改进

（3）多尺度熵（Multiscale Entropy ），它计算时间序列在多个尺度上的样本熵值，体现了时一间序列在尺度上的无规则度。若熵值在尺度上单调递减，则序列在尺度上白相似性较低，结构简单，属于随机时间序列；若熵值在尺度上越大，序列自相似性大，复杂度大；若一个时间序列的熵值在绝大部分尺度上大于另一个时间序列的熵值，则说明前者要比后者复杂

6月28日工作随笔

主要对近似熵功能进行了测试：

(1)进行了信噪比相关的测试，定义SNR=信号幅度/噪声幅度均值，图1的SNR=20，图2的SNR=80，可以明显看出图1中熵值均大于1.2，处于复杂度较高的状态；图2中熵值总体较小，且熵值的变化能够较好的反应新模式的产生。

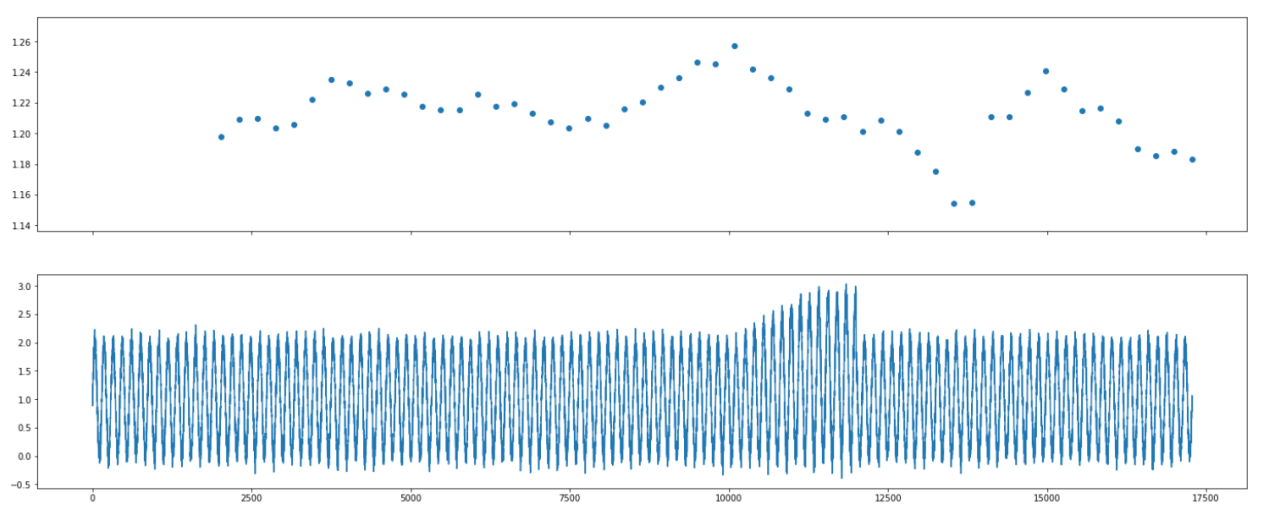


图1

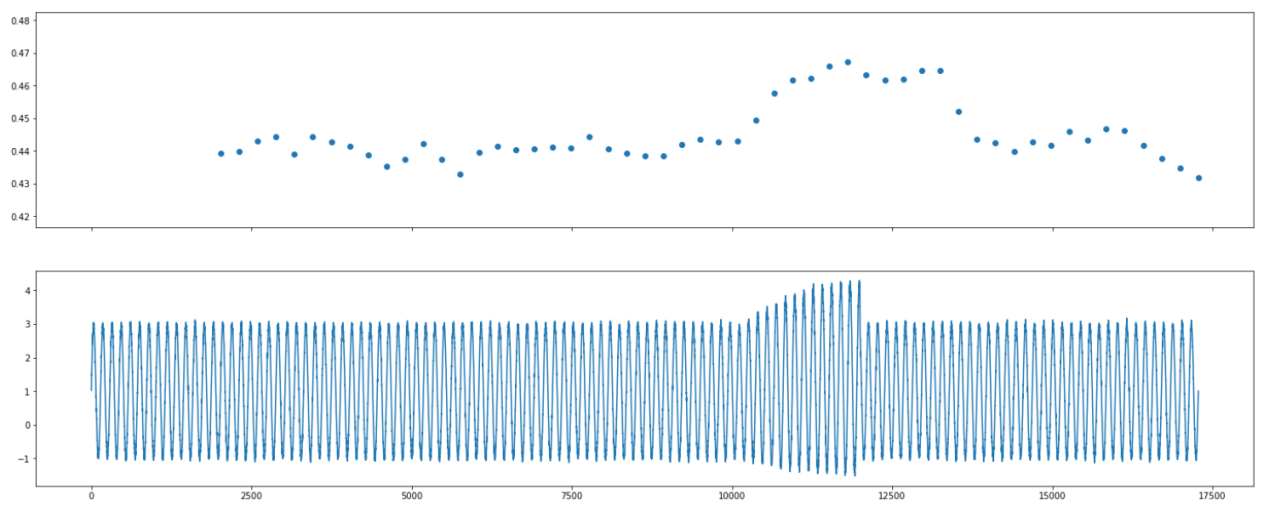
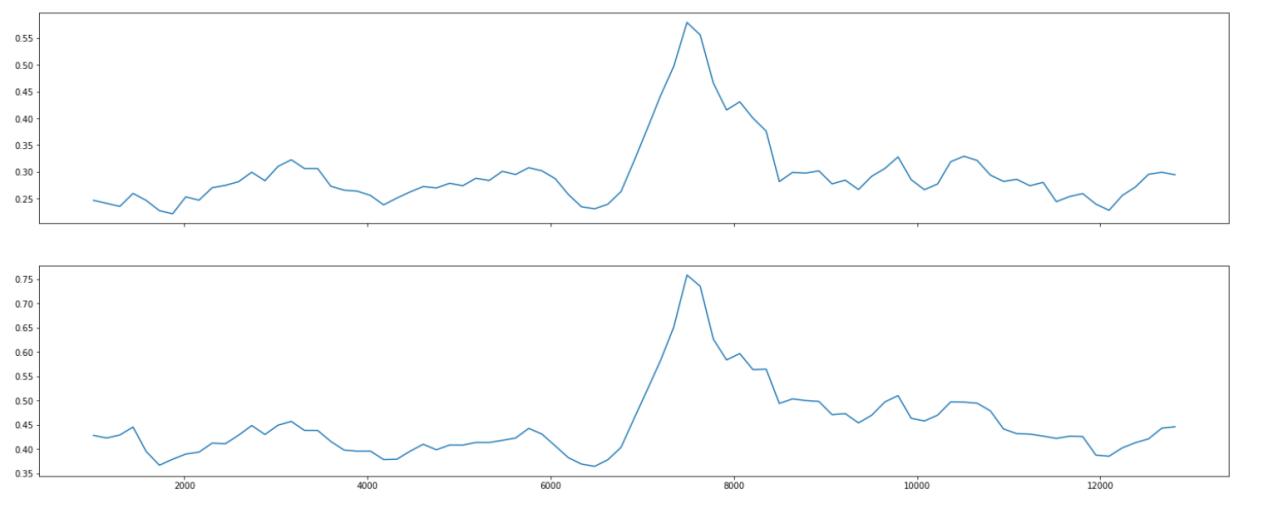


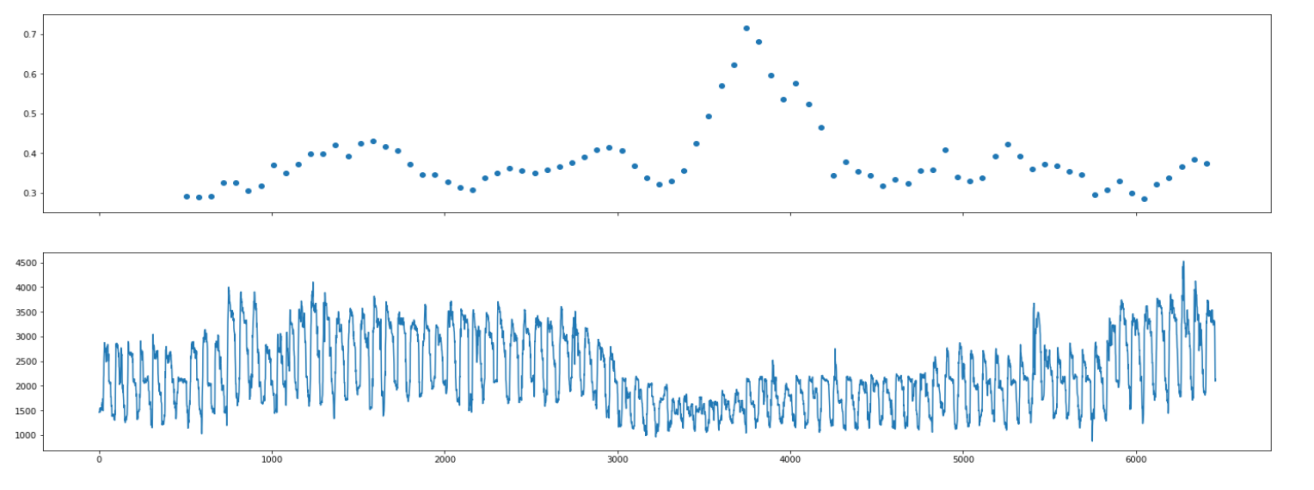
图2

1. 使用样本熵和近似熵进行了对比测试，结果基本一致，如图（3）中所示上图为样本熵计算结果，下图为近似熵计算结果。

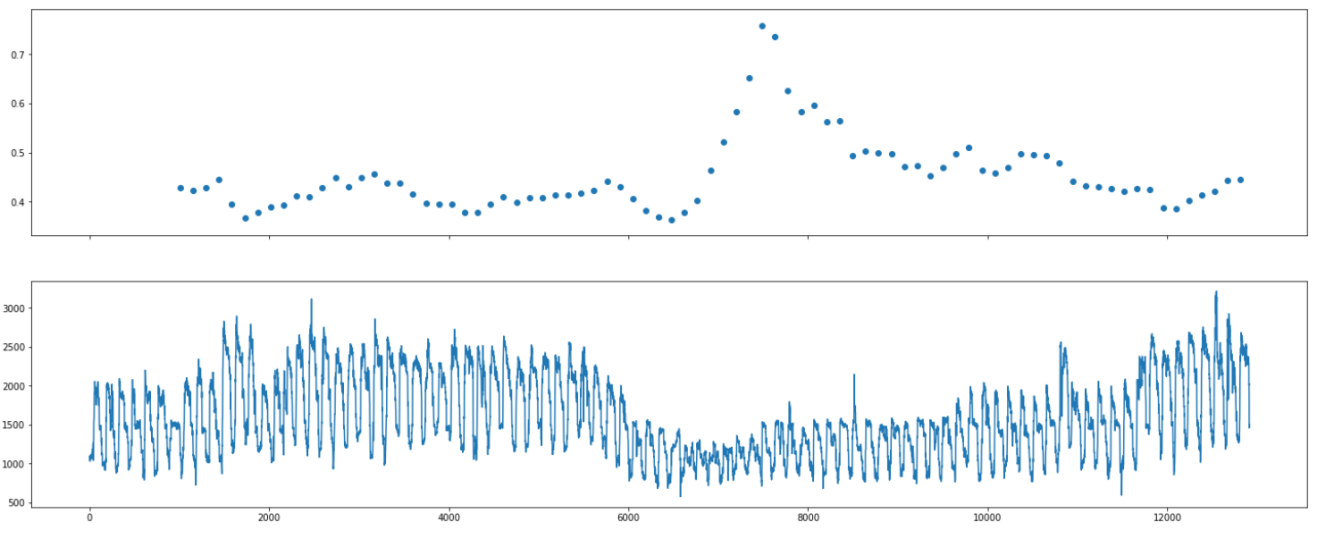


图（3）

（3）采用小波分解的思想对近似熵函数进行了测试，图4为采用一层小波分解后的近似分量进行近似熵计算，得出的结果与原结果基本一致，且由于小波提高了信噪比，使得新模式导致的熵增更加明显，且经过小波分解后，计算熵值得计算量只需原来的一半。



图（4）



图（5）