数据挖掘第二次大作业报告

1. 任务一
2. 数据处理
3. 算法描述
4. 任务二
5. 数据处理
6. 算法描述
7. 任务三
8. 数据分析

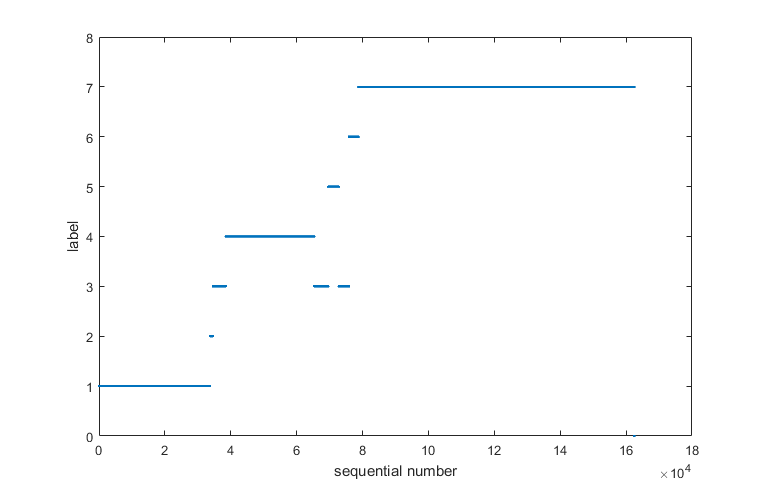
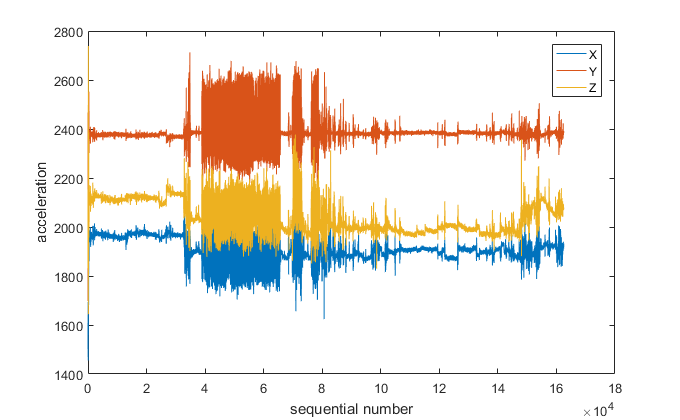


图3-1-1 1.csv文件中各方向加速度随时间的变化及其标签

根据图3-1-1和图3-1-2，我们可以发现，单纯从各方向加速度的变化图来看，该时序数据中存在明显的不同模式，但是仅凭观察，可以总结出近三种不同的模式，结合对应的label图，不难发现，许多类别序列的特征很难分辨，比如序列开始阶段的第1类状态和序列结束阶段的第7类数据，仅有微弱区别。

1. 单一活动类别识别

假设已知一段序列属于一个活动类别，可以凭借序列的特征对各序列进行分类。这是典型的时序数据分类的问题。目前常用的方法有：1）基于距离的序列匹配方法，如DTW；2）基于特征的机器学习分类方法，即对提取出的特征应用决策树，SVM，神经网络等分类方法进行分类。关于时序数据的特征，又主要分为三类：1）基于统计的特征，如一段序列的均值，方差，标准差，众数，中位数和相关函数，滑动平均等等；2）基于时频变换方法的特征，对时序数据进行傅里叶，拉普拉斯变换，将其对应频域的特征进行提取，用于数据分类；3）基于回归模型的特征，主要针对于随机过程，尤其是平稳性易于证明的过程，通过拟合回归对应的模型，将模型的参数作为数据的特征，如ARMA，ARIMA等模型。

通过上述分析，我们你采用基于特征学习的方法对序列进行分类。

1. 特征提取

1）通过计算该序列的平稳性，其原序列和一阶差分序列均不平稳，所以不采用回归模型参数估计的方法进行特征提取。

2）我们对文件1.csv中的x方向上的加速度数据进行按label分类，绘制其频谱图，如下图3-1-2，3-1-3，3-1-4，3-1-5所示：

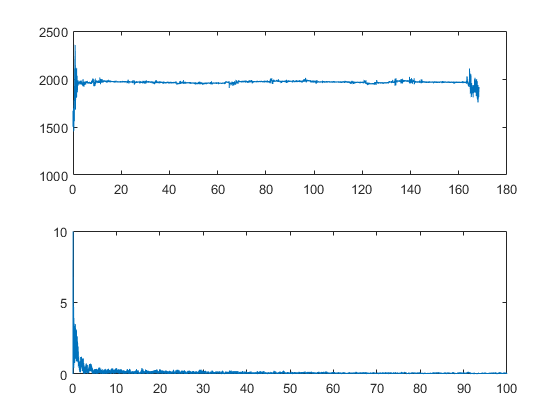


图3-1-2 label为1的序列及其频谱图（上面的图的横纵坐标分别是序列编号和加速度值；下面的图的横纵坐标分别是频率和幅值）

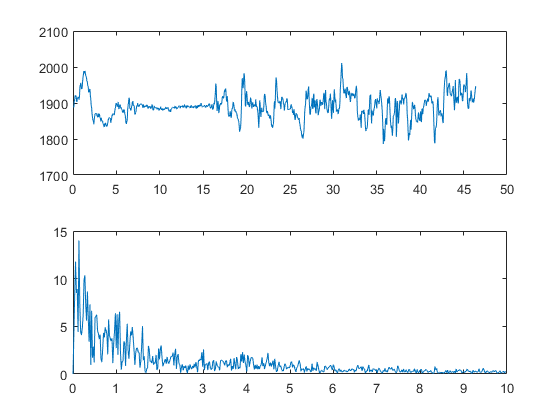


图3-1-3 label为2的序列及其频谱图（上面的图的横纵坐标分别是序列编号和加速度值；下面的图的横纵坐标分别是频率和幅值）

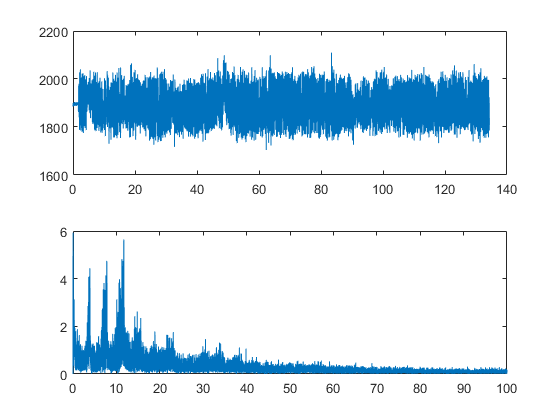


图3-1-4 label为4的序列及其频谱图（上面的图的横纵坐标分别是序列编号和加速度值；下面的图的横纵坐标分别是频率和幅值）

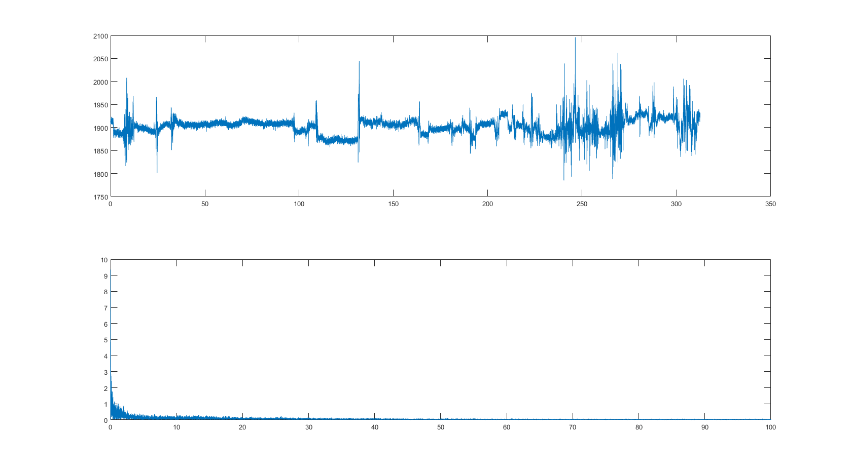


图3-1-5 label为7的序列及其频谱图（上面的图的横纵坐标分别是序列编号和加速度值；下面的图的横纵坐标分别是频率和幅值）

根据频谱图我们发现，不同类别的序列中，仅label为4的序列的频谱在大于10的频段有明显的幅值，其他序列的频谱衰减十分迅速，仅在频率范围为0-5之间有明显可见的值，且十分相似。用决策树算法尝试后仅能得到28%的分类准确率。

因此，频域的数据特征不足以用来区分不同类别的序列。

3）由于数据的采样频率为52Hz，我们将52个数据点即1s内的数据划分为一个序列，形成维度（52，3）矩阵形式。计算该序列每个方向的平均值，方差，最大值和最小值，相关函数等统计量。用决策树，随机森林，逻辑回归等算法均能得到比依据原始数据分类更高的准确率。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | x | y | z | 空间方向 |
| 平均值 | ave\_x | ave\_y | ave\_z | sqrt（ave\_x^2+ave\_y^2+ave\_z^2） |
| 最小值 | min\_x | min\_y | min\_z | sqrt（min\_x^2+min\_y^2+min\_z^2） |
| 最大值 | max\_x | max\_y | max\_z | sqrt（max\_x^2+max\_y^2+max\_z^2） |
| 方差 | var\_x | var\_y | var\_z | sqrt（var\_x^2+var\_y^2+var\_z^2） |
| 相关函数 | ate\_x(10) | ate\_y(10) | ate(10) |  |

表3-1-1 统计特征表

1. 分类算法比较
2. 算法运行环境：处理器i5-3470 内存8GB
3. 算法测试精确度比较

图3-16 三种分类算法在不同统计特征下的精确度

可见在特征相同的情况下，逻辑回归方法优于随机森林优于决策树算法。同时统计特征中相关函数对逻辑回归方法的结果影响很小。逻辑回归的最高准确率为59.65%。相比基于原始数据的平均值这一特征的分类结果34.16%提高了25个百分点。

1. 活动转换点检测

活动转换点的检测主要有两种方法，一种是基于序列比较的滑动窗口，自底向上，自顶向下等方法，另一种是基于概率模型，即最常用的隐马尔科夫模型的方法（hmm）。

首先我们尝试通过无监督学习的方法即hmm模型检测转换点。但是由于元数据存在大量的震荡，高频数据幅值低，但是频域宽度大，因此用元数据作为hmm模型的输入时，输出的状态预测结果仍然会有许多高频成分。

1）对原始数据进行滤波

调用scipy.signal模块中iirdesign设计低通滤波器参数，最后选取的参数为通带频率0.001，阻带频率0.002，通带最大衰减1dB，阻带最小衰减40dB，应用切比雪夫滤波。

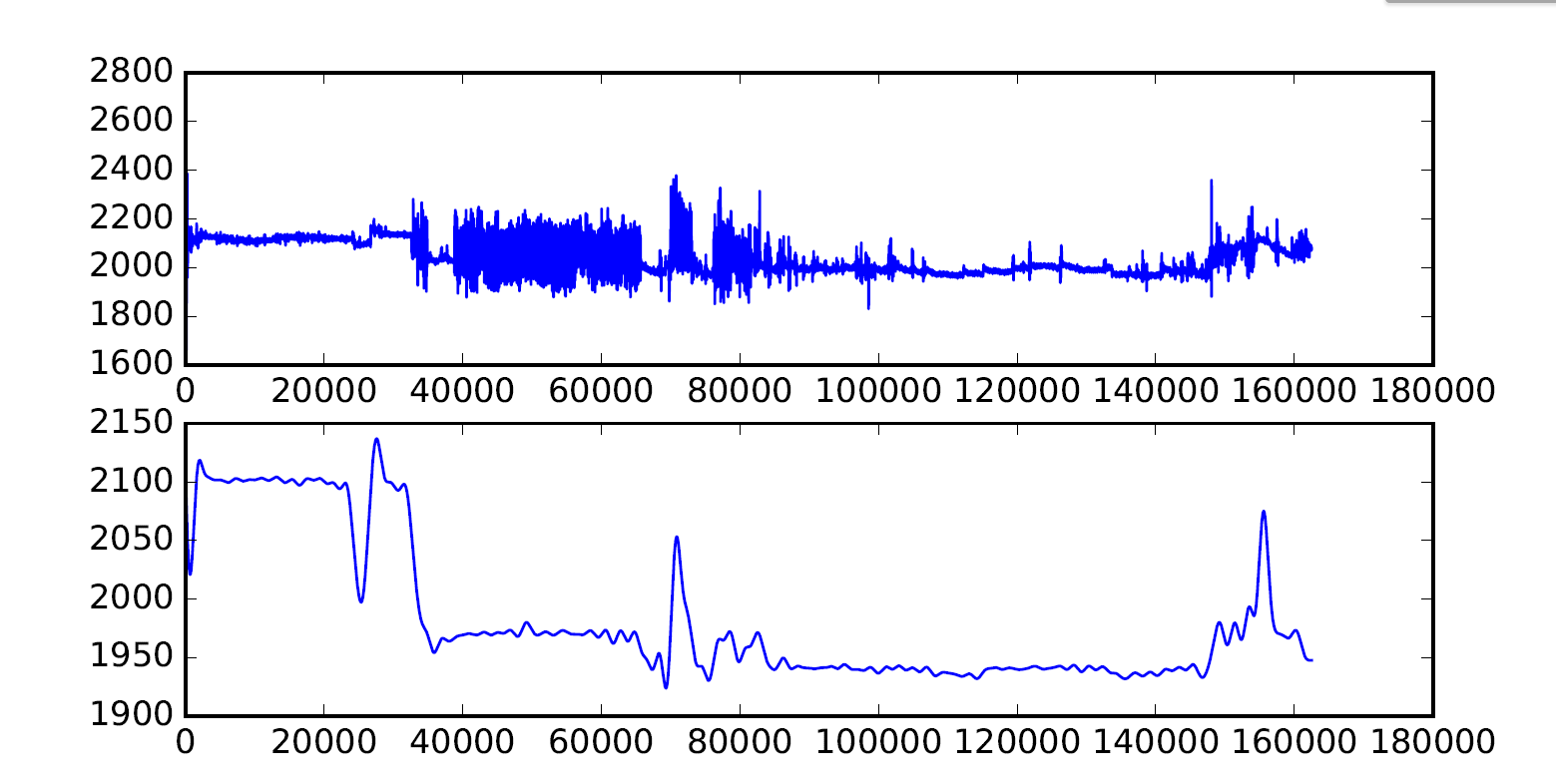
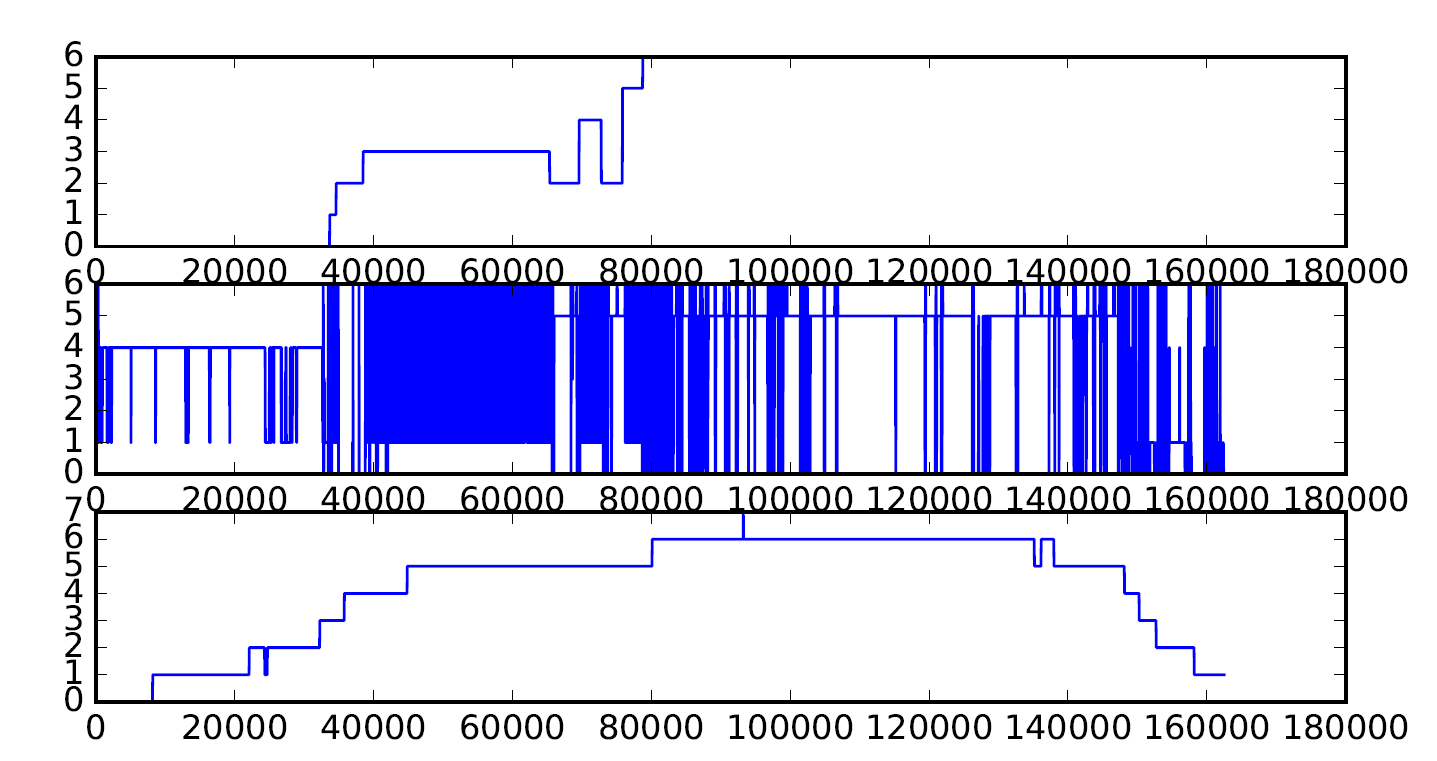


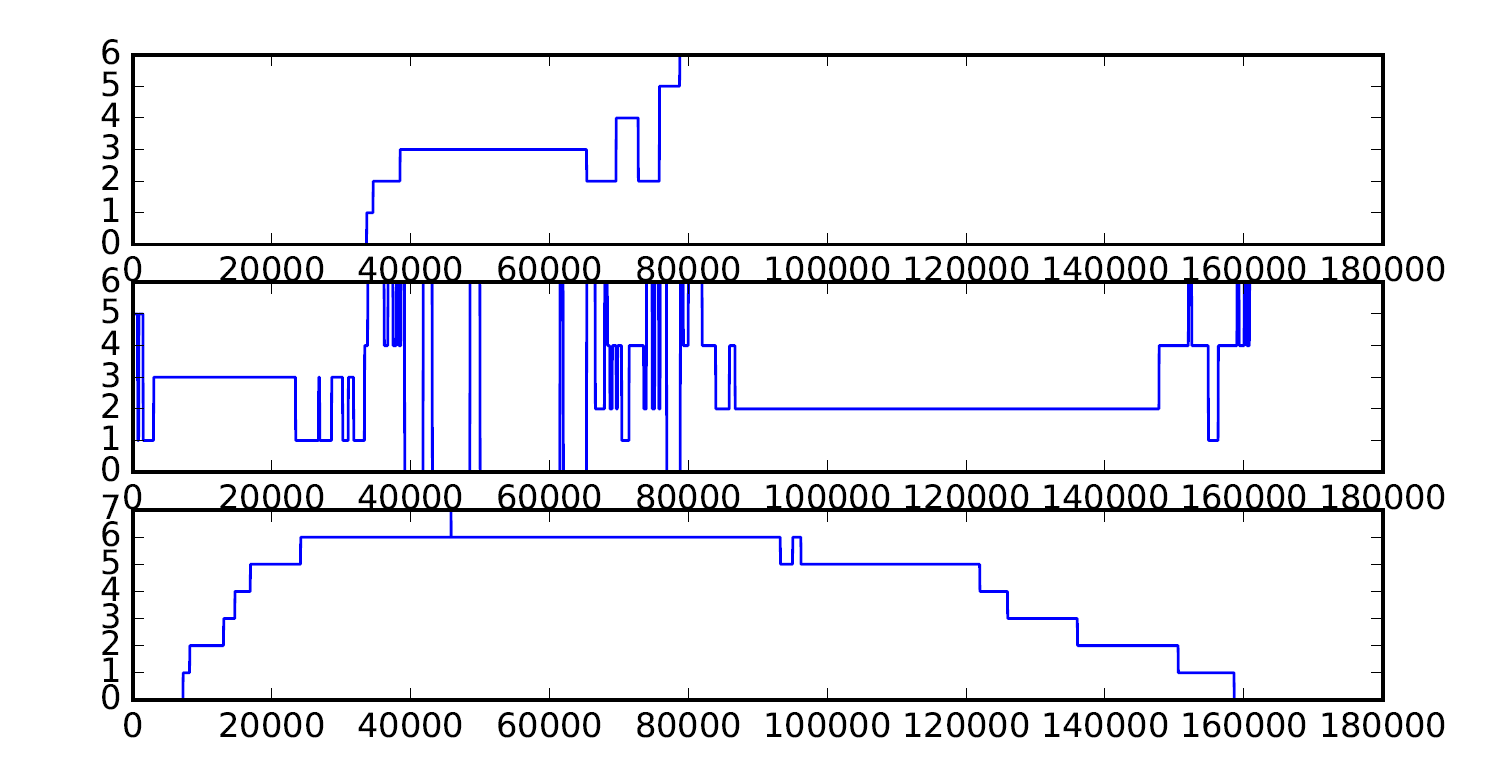
图3-2-1 1.csv中的z方向的数据及其滤波后的情况

2）在未滤波的数据上进行状态转移点检测，除去label为0的个别点，我们将数据分为7个类别。下面三幅图中最上面的图是文件中的数据标签，我们可以认为一条竖直线就是一个转移点。中间图是用HMM检测出的数据状态，可见能够识别出所有的状态转移点，但是也会产生很多并非真实转移点的情况，造成极大的干扰。下面的图是对HMM的输出进行滤波后将结果进行7等分得到的状态图，相比之下丢失了大量真实的转移点。这里对HMM输出状态序列滤波的参数是，通带0.005，阻带0.001，通带最大衰减1dB, 阻带最小衰减40dB。



3）在滤波后的数据上进行HMM状态分类的结果：

下面三幅图中最上面的图依然是文件中的数据标签，我们可以认为一条竖直线就是一个转移点。中间图是用HMM检测出的数据状态，可见能够识别出所有的状态转移点，但是也会产生很多并非真实转移点的情况，造成极大的干扰。下面的图是对HMM的输出进行滤波后将结果进行7等分得到的状态图，相比之下丢失了大量真实的转移点。这里对HMM输出状态序列滤波的参数与2）中一致 。

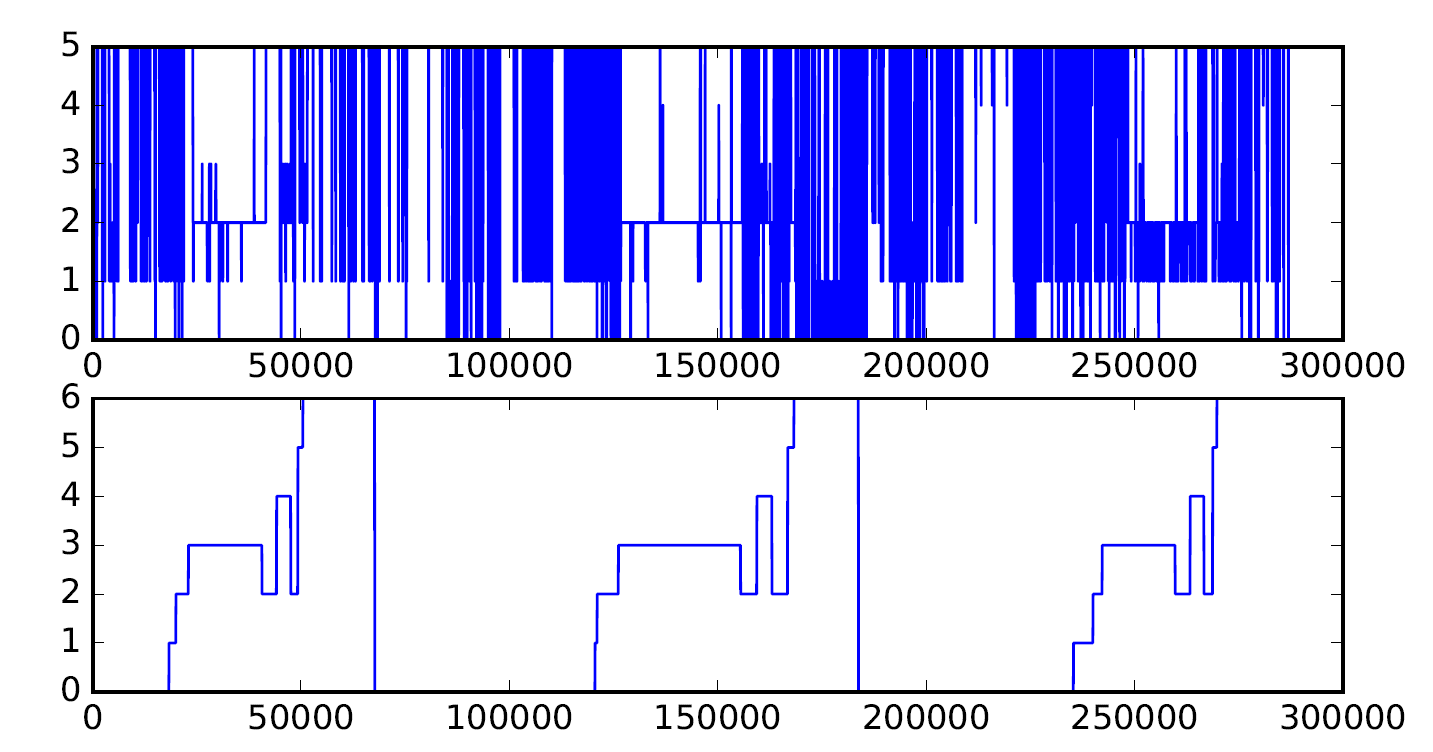


4）应用HMM得到的状态转移点

可见基于HMM得到的结果震荡依然严重，虽然包括了所有的状态转化点，但是不进行滤波的话会有大量假转换点产生，滤波之后又会失去一些真实转换点，很难达到折中的平衡结果。在这种情况下，我们选择使用滤波后的数据输入到HMM模型中，获取状态分类结果，不进行进一步滤波，直接输出其所有转移状态，即遵循不漏掉任何真实转移点的原则。

同时，我们又做了另外一个假设：我们默认每一秒内的状态是不会发生改变的，利用序列分类的方式，将到来的序列按照52个数据点为一个序列进行切割，对每一个序列进行分类，这样即使原本不是同一状态的序列被切割到一类序列中，仅产生1s的误差也是可以接受的。

用该方法获得的序列状态如图所示：测试集的所有数据组成的。



1. 多活动类别识别

对于一个序列的多活动类别识别有两种方法，一是利用上面的结论，首先对数据进行状态转移点的检测，再对检测出的每一个单一类别进行单一活动分类。我们用HMM的结果和逻辑回归分类方法进行了尝试，整体单个时间点的分类准确率只有15.0080256822 %。但是某些段内的准确率很高。列举一些较高准确率的类别如下：

|  |  |
| --- | --- |
| 段标签 | 预测准确率 |
| 0 | 100% |
| 0 | 23.0765% |
| 2 | 62.162% |
| 4 | 100% |
| 5 | 20% |

另一种方法是对每一个时间点的状态进行判断，即不依赖于序列的特征，我们设计了一个4层前向神经网络模型，神经网络参数为：层数4，对应的每层节点数分别为：100，50，20，10；学习率0.01，最大迭代次数4000。对原始数据进行训练，每一个时间点输出一个活动类型，结果43.27%。