STAT 542 Final Report

Wenzhao Xu (Environmental Engineering)
Haoyan Cai (Statistics)

December 17, 2013

1 Introduction

The data are about accelerometer data from mobile devices. The train data contain X,Y,Z axis acceleration values from 387 devices and testing data set consists of 90024 testing sequences. Each test sequence comes with a proposed device Id. The goal is to judge whether the claimed device is the true device that produces the test sequence. The main difficulties in this project are: (1) Data has outliers and gaps (i.e. the sampling time interval is not constant so that two sampling points might have very large time interval); (2) Feature extractions; (3) Treat it as a multi-label classification problem or as a 0-1 classification problem by some label creating methods.

2 Method

The basic assumption is that each user will have similar behaviors if he/she is doing the same activity like walking and running and he/she would probabily do the same activity at the same time of day.

First we treat it as a multi-class problems, that is trying to predict which device generates the test sequence. Given the training data of a certain device, we first divided the whole training data into pieces, each has around 400 points (400 is a tuning parameters). For most devices, 400 sampling points will last for about 1 minute and we assume the user is doing the same activities in this 1 minute. Each piece represents an activity, no matter what the activity is. So for train data of a device, we have several pieces and all of them have the same label as that device ID. Features are extracted from these pieces and also from the test sequences. In this way, we have training sequences with labels and test sequences. Certain classifier is trained and predicted the labels of test sequences.

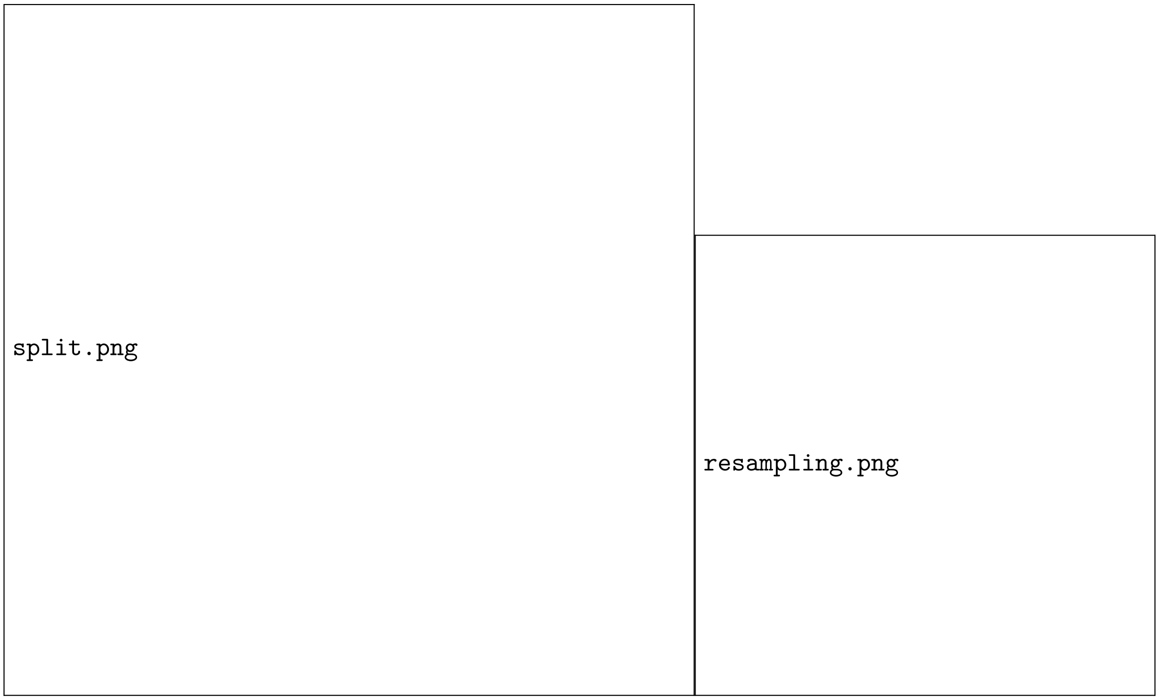
We are also trying to change the problem as a 0-1 classification problem. A possible approach is to use self-boosting. The general idea is that we randomly choose sequences from training data, this way we know what devices they belong to so that we could label them as 1. To label negative examples, we choose sequences from one device and claim it to be from another device. Currently, we are still working on it.

2.1 Data Preparation

The data is imported into R by "ff" packages (SQL database technique is also available). Then the data is preprocessed through splitting,resampling and smoothing to get the final training data and test data. Figure 1 shows an brief of how data is pre-processed.

2.1.1 Splitting

It is found that the training data of a given device don't have a constant sampling frequency, that is, the time interval of sampling poitns are not the same. This might be due to users' manually



(a) Splitting

(b) Resampling

Figure 1: Splitting and Resampling Steps

switching off device or the way Android OS deal with accelerometer sensors. So the whole training data might have large gaps in time. In order to deal with this, we first calcuate the sampling time intervals. If the time interval is larger that 2 minutes, we split the training data into two data pieces at this point. After the data is divided into several pieces, we further divide each pieces into smaller sequences, each sequence has 400 raw data points. At the end of this step, the training data from 387 devices are splited into almost 70000 sequences. This step is shown in Figure 1(a). In addition, for test data, similar splitting method is used. If the time interval in one test sequence is larege than 3 minutes, we divide the test sequence into 2 parts and discard the part with less data points.

2.1.2 Resampling

In splitting step, we split the train data into sequences. However, in each sequence, the sampling frequencies are not always the same. Some might be 200 ms other others might be around 100ms. Such inconsistance makes it very hard for further analysis, especially for frequency analysis. So we do a resample step. A constant sampling time is set based on the initial sampling time and the median value of original sampling intervals. Here median value is used to eliminate the influence of sampling interval outliers. Then, on the new sampling time, the resampled acceralation value is calculated by linear interpolation based on two nearest sampling points. In Figure 1(b), the brown points are the new data points based on interpolation.

2.1.3 Smoothing

Train data have noises such as large spikes. So a 5-point moving average algorithm is then used to smooth the data. 5-point moving average is also used in other literatures. Figure 2 shows the data piece before and after resampling and smoothing steps.

Rplot.pdf

Figure 2: Comparison before and after resampling and smoothing

Table 1: Feature Selection

Kind	Description	Physical Meanings
1.Mean and Variance (8 features)	Mean and variance of acceleration values in each axis as well as the total accerlation A	The habbit of how user put their cellphones and the strength of their activities
2.Correlation (cor) (3 features)	the correlation coefficients between x and y, x and z, and y and z axis.	The users features when doing activities
3.Frequency Features (8 features)	The mean value of the first 5 dominate frequencies (frequencies with highest amplitude) and mean value of energy in these frequencies	Users walking features.
4.Time (1 feature)	mean time of day	User's habbit when doing such activity. 5.Sampling Frequency (1 feature)
The sampling frequencies of this device	Device feature	

Table 2: Case Setting

Case No	Splitting Size	Feature Selection	Negative Size/ Positive Size
0	300	Feature 1,2,3,4	1
1	300	Feature 1,2	1
2	300	Feature 1,2,3,4,5	1
3	300	Feature 1,2,3,4	5
4	400	Feature 1,2,3,4	1

2.2 Feature Extraction

First, the total acceralation value is calculated by $A = \sqrt{(a_x^2 + a_y^2 + a_z^2)}$ and added as a new time series. Based on literature research, 5 kind of features are extracted from the raw data. The first is mean and variance. The second is the correlation. The third is the frequency pattern and the last is the time. Totally, the number of features are 21, which is shown in Table 1.

2.3 Classification

There are two ways to analyze this problem. The first is to treat the problems are a multilabel classification problem, i.e. determine which device the test sequencies are belong to

3 Result

We have set different cases to study the effect of different parameters. The training error are calculated by cross-validation (in KNN method), out of bag error (in random forest) or error on training data (in other methods). And the prediction results are uploaded to kaggle website to see the prediction results.

3.1 subsection name

Case 0 is the baseline scenario. Case 1 reduces the features to the basic means and variances. Case 2 add sampling frequencies to the feature space. Case 3 increase the negative sampling size and Case 4 increase the average size of each data pieces.

3.2 Prediction Result

- All methods except multilabel knn have good results in training data
- Compare with Case 0 and Case 1, it seems FFT frequency features are not very important
- Compare with Case 0 and Case 2, it seems sampling frequencies are not important too, although random forest do have a better results.
- Compare with Case 0 and Case 3, it
- Compare with Case 0 and Case 4, it seems with small splitting size have only marginal improvement.

4 Conclusion

We will work on feature extraction and also tune the some parameters such as size of training sequences. Moreover, other methods such as random forest and ANN will be applied to see the improvement. Finally,feature selections will also be performed. If we sucessfully convert the problem into a 0-1 classification problem, we will also compare the performance of these two frameworks.