Biometric in Motion: You Need This to Know Who You Are

Zexi Mao
zexim@andrew.cmu.edu

Siqi Tan
siqitan@andrew.cmu.edu

Yuchen Wu
yuchenw@cs.cmu.edu

Yimeng Zhang
yimengzh@cmu.edu

October 10, 2013

1 Introduction

Ubiquitous computing had never been so close to us as our phones become more powerful and smarter today. Sensors equipped in our cellphones like GPS, gyroscope, accelerometer and even barometer collect data from the surrounding environment and from ourselves. This information collected from our cellphones and other mobile devices can boost many more applications than just adjusting the brightness and rotating the screen. Google uses the GPS data from our phones to provide real-time traffic conditions. Apps can track how well you sleep by just putting your phone beside you on bed. Indoor localization takes advantage of wireless fingerprint, sound from microphone, footsteps from accelerometer and even colors from the camera to tell you where you are when GPS signal is weak inside buildings. More fancy applications far beyond the original purposes of these sensors are on the way. How to retrieve more information from the raw sensor data is one of the hot and promising topics today.

Accelerometer is one of the most interesting sensors in our cellphone which records the acceleration data in 3D. Posture of the phone, gestures and footsteps of the user, and even the user's sleeping condition can be measured by the accelerometer.

2 Targets

In this project, we are trying to develop a novel usage of the accelerometer: biometric identification. In other words, can we identify the user by only looking into how he/she moves? We believe that everyone has his/her own unique pattern of movement. If this assumption is true, we can identify the person when we match the current data from the accelerometer to the historical data we have learned.

We will learn a few things from this project besides a better understanding of machine learning itself if our machine learning algorithms finally identify users. First, the assumption about pattern of movement could be true. Second, applications such as anti-theft, health monitoring and emergency detection that adopt this novel identification technique will be available in the near future. Third, we will be able to know how much data from the accelerometer is sufficient to cause the leak of one's identity, which can trigger a serious privacy issue.

Success criteria:

100%: identify users as accurately as we can, try to win the competition.

120%: show how much data a user can provide before he/she compromises his/her own privacy.

3 Data

The data set can be easily acquired from Kaggle which consists of 3 parts: the training dataset (train.csv), the testing dataset (test.csv) and question set (questions.csv).

In the training data set, there are 30 million samples, each containing the time stamp, acceleration measurements in 3 dimensions, and the associated DeviceId.

In the testing data set, there are 30 million samples as before, without DeviceId, but demarcated into 90,000 sequences, each with a SequenceId.

In the question set, for each SequenceId, there's a proposed DeviceId.

Our ultimate task is to determine whether the accelerometer recordings in the test set belong to the proposed devices in the question set.

4 Methods

To transform raw accelerometer data to a form which is more appropriate for the classifiers, a low pass filter (LPF) may be first applied to filter out high frequency noise. Then we may split the 30 million samples in the training set into sequences of 300 samples following the practice of Kaggle. After that, we plan to extract three sets of features for classification, namely, time domain features, frequency domain features, and spatial domain features.

First, we plan to try some methods that make classification based on C binary classifiers, where C is the number of classes (DeviceId in this context). In terms of the binary classifier, we will initially try logistic regression. We choose it because its output can be interpreted as the confidence over classification, which is consistent with the evaluation criteria. If this doesn't work well, it may be due to that the data points are highly non-linear, and we may try some kernel versions of logistic regression. In addition, probabilistic versions of SVM are worth trying.

Second, we will try some generative methods, like naive Bayes classifier and Gaussian mixture model, to model the joint distribution of label (DeviceId) and features (timestamps and acceleration measures). Again, kernel tricks may be applied to these methods.

Third, nearest-neighbor methods will be used to classify the test cases according to our distance metric. Since we are required to give the confidence of classification, we may use weighted versions of k-NN methods to accurately express the confidence.