Biometric in Motion: Identification Using Acceleration Data

Zexi Mao
zexim@andrew.cmu.edu

Siqi Tan
siqitan@andrew.cmu.edu

Yuchen Wu
yuchenw@cs.cmu.edu

Yimeng Zhang
yimengzh@cmu.edu

Abstract

As cellphones and other smart devices are closer to our daily life than ever, we are curious to know how the data gathered by accelerometers on them impact our lives, particularly in terms of identifying users of devices via such data. In this project, we try to answer this question with machine learning techniques. We will first study how to implement such biometric techniques by retrieving useful identification information from raw accelerometer data and building classifiers. Then we will show how such techniques impact our lives.

1 Introduction

1.1 Motivation

Why is it important Ubiquitous computing had never been so close to us as our phones become smarter and more powerful today. Sensors equipped in our cellphones like GPS, gyroscope, accelerometer and even barometer collect data from the surrounding environment and ourselves. This information collected from our cellphones and other mobile devices can boost many more applications than just adjusting the brightness of the screen and rotating it automatically. 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. Fancier 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 hottest and promising topics today.

What is the problem 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.

In this project, we tried to explore 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 with the historical data we have learned.

Brief outline of solution In this project, we chose the already available accelerometer dataset gathered by Kaggle. Due to the poor structure of the raw data set, we first did some preprocessing, mainly splitting and re-sampling. Then we extracted discriminative features from preprocessed dataset. The features we used included FFT coefficients, the distribution of timestamp intervals, and the set of possible acceleration readings. Finally, we applied several classifiers to the feature vectors. Now we

have got a score of almost 0.90 in terms of area under the ROC curve, which is satisfactory, although further improvement is still needed.

1.2 Related Work

A human being's walking gait can reflect the walker's physical characteristics and psychological state, and therefore the features of gait can be employed for individual recognition. The use of accelerometer data for biometric identification is relatively new but has been increasingly explored in recent years. Existing methods for gait recognition have shown good performance.

Gafurov et al. [1] attach multiple sensors to a subject at different body parts. Xu et al. [2] developed an Android App to collect gait acceleration data. With a reasonably sized dataset, by matching gait patterns across different paces, they show preliminary results indicating that not only can smart phones be used to identify a person based on their normal gait but also that there is potential to match gait patterns across different speeds.

Tao et al.[5] focus on the representation and pre-processing of appearance-based models for human gait sequences. Two major novel representation models are presented, namely, Gabor gait and tensor gait. Experiments show that the new algorithms achieve better recognition rates than previous algorithms.

Pan et al. [4] proposed algorithm based on signature points, instead of the whole gait signal. They consider acceleration-based gait recognition insensitive to changes of lighting conditions and viewpoint. Their algorithm firstly extracts signature points from gait acceleration signals, and then identifies the gait pattern using a signature point-based voting scheme. The experimental results shows the accelerometer-based gait biometrics is promising.

Kwapisz et al.[3] collect some data and also perform identification experiments . Based on the 600 raw accelerometer readings, they generated 43 features, which are variations of 6 basic features including average acceleration value, standard deviation, time between peaks and so on. They applied two classification techniques decision trees neural networks to classify and the identification performance turned out to be fairly good.

2 Information on the Kaggle competition

We got the dataset from the Kaggle competition "Accelerometer Biometric Competition". The dataset consisted of 3 parts: a training set, a testing set and a question set. Each single sample point in training and testing sets contained a time stamp in milliseconds, acceleration measurements in 3 dimensions, and an associated DeviceId (for the training set) or a SequenceId (for the testing set).

In the training set, there were 30 million samples, which were collected from 387 different devices as labeled. These samples were demarcated into 387 segments, each containing samples for a single device. In the testing data set, there were also about 30 million samples without label. These samples were demarcated into 90,024 sequences of 300 points, each with a SequenceId. In the question set, for each SequenceId, there was a proposed DeviceId.

The task on Kaggle was to tell whether each sequence's proposed DeviceId was the sequence's real DeviceId. For each sequence, we were required to give a belief, in real number, about the credibility of the proposed DeviceId. This competition was designed to investigate the feasibility of using accelerometer data as a biometric for identifying users of mobile devices. For more detail, please refer to the competition website as listed above.

3 Methods

In this section, we describe the methods we have used for preprocessing, feature extraction, and classification. This is not necessarily the final version of our approach.

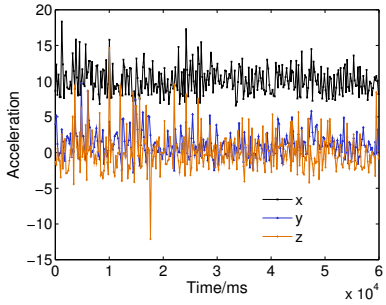

Figure 1: A good sequence from raw data. Samples points were close in time. We regarded these points as representing a single activity.

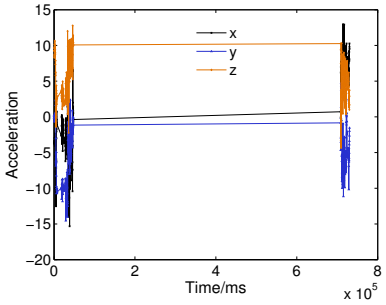

Figure 2: A bad sequence from raw data. This piece essentially contains information about 2 separate activities that were far apart in time. The line segments in the middle of the figure was due to the large time intervals between adjacent sample points.

3.1 Preprocessing

There were two problems with the raw dataset for further processing: 1) accelerometer data for each device was not properly segmented; 2) accelerometer readings were not uniformly sampled. For the first problem, we performed segmentation on 387 training sequences and 90024 testing sequences. For the second problem, we performed re-sampling based on cubic spline interpolation.

3.1.1 Segmentation

First, the raw data was not properly segmented. For each sequence, the time intervals between two adjacent sample points were less than 500 ms most of the time. However, some intervals could be as huge as 100,000 ms. This problem was illustrated in Figures 1 and 2. They contain two test sequences of 300 points. In the first sequence (Figure 1), all the points were close in time, and we could regard these points as representing a single activity. In the second sequence (Figure 2), there were two points that were far apart in time (so all the other points were pushed to two sides of the figure), and we can consider this sequence containing information about two separate activities.

Clearly, we couldn't consider the second sequence as representing information for a single activities, since so much information was unknown between the two far apart sample points. Such huge time interval was not uncommon in the dataset.

To overcome this problem, we performed segmentation on the raw data to split each sequence into subsequences. Our segmentation had two phases. In the first phase, we split the sequence at points where the time interval was larger than T_{split}. After this, for each original sequence, we got several shorter sequences that had sample points close in time. In the second phase, we continually split the longest subsequence among all the subsequences until 1) we got $k_{\#\text{seq}}$ subsequences, or 2) the longest subsequence was shorter than $2l_{\text{maxL}}$. Also, sequences shorter than T_{short} were thrown.

We did the second-phase segmentation because after the first-phase segmentation, some resultant sequences were too short (less than 1000 ms) so that not much reliable features could be extracted, and some were too long (longer than 100 seconds) so that it might contain information about multiple activities of the user.

In this manner, on the one hand, we make more sequences from one device, hopefully keeping more information about the device's diverse patterns. On the other hand, we keep the sequences long enough, so that they contain reliable information about patterns of the device.

3.1.2 Re-sampling

After segmentation, for each device, we got several subsequences. However, these sequences were not uniformly sampled, due to various reasons. To make feature extraction simpler, we interpolated

each segment using cubic spline and uniformly re-sampled each sequence with τ_{resample} being the re-sampling time interval.

3.2 Feature extraction

After preprocessing, for each raw sequence (from training set or testing set), we got several shorter re-sampled subsequences. In this section, we describe how to extract a feature vector from each subsequence.

In the end, for each subsequence, we got a feature vector in $\mathbb{R}^{1290}$. This vector can be decomposed into 3 parts: frequency domain features, time domain features, and accelerometer reading features.

3.2.1 Frequency domain features

Basically, we thought the patterns of the device might be better revealed in the frequency domain. To this end, for each subsequence, we first extracted sliding windows of size w_{fft} and 50% overlap (determined empirically). Then we did FFT on all windows. Finally, we calculated the magnitude of all first $(w_{\text{fft}}/2+1)$ coefficients and use the average values over all windows as the frequency domain features for the subsequence. Since there were readings from 3 directions, we have $(w_{\text{fft}}/2+1) \times 3$ values for frequency domain features per subsequence.

3.2.2 Time domain features

As the mobile devices people use are increasingly diverse, the characteristics of different devices offer us a set of very distinguishing features for recognizing users. One of these characteristics was the sampling intervals of the devices.

As we observed from the raw data set, sampling intervals of devices has a range of approximately 5-250 ms, and for different devices, the distributions of the sampling intervals were quite different. Thus, for each subsequence, we used the histogram (0-600 ms divided into bins of 20 ms, in total 30 values) of the sampling intervals of the original sequence it came from as the time domain features.

3.2.3 Accelerometer reading features

Apart from the sampling intervals of devices, we found another set of features with potential ability of distinguishing devices, which was the accelerometer readings.

By observing the data, we found that each kind of device could only provide certain discrete values of accelerometer readings, and this might be due to the limited precision of accelerometers. Thus, the set of accelerometer readings a device can provide is also treated as a set of features. For each subsequence, we used the overlap between the the set of readings from the original sequence it came from and the set of readings from the ith device, $i = 1, \ldots, 387$. Since we had reading from 3 directions, $3 \times 387 = 1161$ values constituted the accelerometer reading features for each subsequence.

3.3 Classification

This is the ongoing part of our work. The basic idea is to leverage various kinds of classifiers and optimize their parameter for more accurate results.

After preprocessing and feature extraction, for each original sequence, we now had several feature vectors associated with it. These vectors were used for classification. For testing, the average of prediction results of all feature vectors associated with a test sequence was used for submission.

We start from some naive classifiers. The performance of K-Nearest-Neighbor depends on the number of K and the definition of distance function. This classifier gave us a baseline result. Then we move on to popular classifiers such as support vector machines and logistic regression.

4 Implementation and Evaluation

In this section we show how our approach performed. We will also try to justify the selection of parameters for preprocessing.

4.1 Parameter selection

In this subsection we discuss how to determine the parameters and thresholds we defined in the previous sections.

T_{split} was set to be 10 seconds to be consistent with the preprocessing setting done by Kaggle. T_{short} and l_{maxL} were both set to be 4.8 seconds for the training set and 3.2 seconds for the testing set empirically. τ_{resample} was set to be 200 milliseconds based on statistics of the distribution of the time intervals of the raw data, but any value near 200 ms should be a reasonable choice. The window size of FFT, w_{fft}, was 64, which meant a window of duration 12.8 seconds. For training, $k_{\#\text{seq}} = 50$; for testing, $k_{\#\text{seq}} = 5$.

4.2 Optimizing classifier

This is the To Be Done part.

4.3 Results

According to the rules of this competition, classification results were judged on area under the ROC curve. The three kinds of classifiers we used were k-nearest neighbors, SVM with Gaussian kernel (using LIBSVM) and L_2-regularized logistic regression (using LIBLINEAR). The results along with the competition baseline provided by Kaggle (kNN, with the mean of all accelerometer readings as the feature vector for each device in the training set, and the the mean of all readings as the feature vector for each sequence in the testing test), is shown in Table 1. For k-nearest neighbors (k-NN-FFT in the table), only the frequency domain features were used, with $k = 40$. For SVM with Gaussian kernel (SVM-RBF-FFT in the table), $C = 5, \gamma = 10$ was used in LIBSVM, and only the frequency domain features were used. For logistic regression (LR-3 in the table), $C = 2$ was used in LIBLINEAR, and all three kinds of features were used.

Table 1: Testing results of four methods

Methods	k-NN-Mean (baseline)	k-NN-FFT	SVM-RBF-FFT	LR-3
AUC	0.50277	0.58590	0.80077	**0.89729**

Appendix

Our updated timeline is shown in Table 2.

Table 2: Updated Timeline

Time	Steps To Desired Goals
	Make use of new set of features (e.g. , time of day in each sample)

Our plan for future goes here.

If possible, we will evaluate our technique not by testing it on the Kaggle dataset, we will also develop apps for our smart phones to collect real time data for learning and evaluation. We hope this approach will give us the clue of how to protect our id privacy.

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.

References

[1] D. Gafurov, E. Snekkenes, and P. Bours, "Gait authentication and identification using wearable accelerometer sensor," in *Proceedings of IEEE Workshop on Automatic Identification Advanced Technologies*, June 2007, pp. 220–225.

[2] F. Juefei-Xu, C. Bhagavatula, A. Jaech, U. Prasad, and M. Savvides, "Gait-id on the move: pace independent human identication using cell phone accelerometer dynamics," in *Proceedings of IEEE 5th International Conference on Biometrics*, 2012, pp. 8–15.

[3] J. R. Kwapisz, G. M. Weiss, , and S. A. Moore, "Cell phone-based biometric identification," in *Proceedings of Fourth IEEE International Conference on Biometrics: Theory Applications and Systems*, September 2010, pp. 1–7.

[4] G. Pan, Y. Zhang, and Z. Wu, "Accelerometer-based gait recognition via voting by signature points," in *Electronics Letters, 45(22)*, 2009, pp. 1116–1118.

[5] D. Tao, X. Li, X. Wu, and S. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence ,29(10)*, October 2007, pp. 1700–1715.