Classification of Mobile Device Accelerometer Data for Unique Activity Identification

Kai Kuspa, Tony Pratkanis

December 13, 2013

Abstract

Accelerometer datasets from 36 smartphones were analyzed in order to distinguish between different user activities. Training data from jogging, walking, standing, stair climbing (ascending and descending) and sitting were collected for each device at a sampling rate of 20Hz. The spectrogram for each activity was transformed using PCA to reduce the dimensionality of the dataset, and the eigenvalues of three dimensions (X, Y, and Z coordinates of the accelerometer) were used as inputs for Gaussian Discriminant analysis to label the recorded activity. Kwapisz et. al. explored the possibility of using accelerometer data to classify user activity, and achieved a 90% success rate in their classification. Using our methodology, we achieved a global success rate over 92%, however, our results are generated using a direction-agnostic approach which increases the practicality of such a system for real-world use.

1 Introduction

The increasing prevalence and diversity of sensors in smart mobile phones provides exciting new opportunities to collect and analyze continuous sensor data. Data-mining information from cameras, microphones, accelerometers and other sophisticated sensors could provide advantageous feedback informing the mobile device about the user's daily activities. Providing context aware information to the user could further enrich the experience and utility of owning a smart device. In this paper we describe a new application of machine learning algorithms that aim to classify physical user activities such as running, walking, and standing based on accelerometer data. An accurate system such as this could be used to create context-sensitive mobile applications that provide content and direct device alerts based on the users' actions (e.g. sending calls to voicemail if the user is jogging). Further, this information could be used to provide data for "quantified self" applications, in which people utilize data analytics in an effort to track individual health habits such as measuring the user's daily amount of exercise.

2 Previous Work

There has been a diverse history of previous work in the field of activity recognition. Many methods involve specialized devices that are not general purpose mobile devices. These devices (such as FitBit) are expensive and represent an extra burden to the user who is often already overwhelmed with information. Therefore, work has been directed to accomplishing similar results with general purpose devices such as phones and music players. Kwapisz et al. and Saponas et al. are examples of schemes for activity recognition. However, these schemes are deficient because they are sensitive to the way in which the phone is affixed to the user, making them impractical for real-world scenarios. Despite this, commercial systems are available for both iOS (Moves) and Android (Provided by Google). These commercial systems also offer reusable APIs for development of applications on top of these classifiers. However, they suffer from a limited range of activities recognized and they are cloud services, leading to privacy concerns. These deficiencies motivated us to create an improved mobile device-based activity recognition system.

3 Methods

Our general methodology was to take samples of accelerometer data from user's phones and attempt to predict the activity. The general expectation of the workflow of our project is that the user will first use a graphical interface to train their phone by specifying what activity they are undertaking. Once this training process is completed, the system will be able to automatically detect the current activity. We decided, in agreement with Kwapisz, et. al., to employ 10-second windows of data in order to gather enough data to successfully classify activity without greatly inconveniencing users. We also believe that this window contains enough time for activities with periodic repetitions (e.g., footsteps) to become evident. We then took these windows and fed them through various algorithms to generate a feature space and then classified based on this space.

3.1 WISDM Actitracker Dataset

The Actitracker dataset is a high-quality dataset produced by Kwapisz, et. al., at the University of Fordham. The Actitracker dataset consists of over 1 million data points from 36 different users. The data is further subdivided into six labels: walking, jogging, sitting, standing, upstairs, and downstairs. The data consists of raw time series data with time, X, Y, and Z for each data point. The X-axis represents side-to-side (across the hips), the Y-axis represents the vertical axis and the Z-axis represents front-to-back motion. The phones were placed in the pockets of the participants' pants, but carefully oriented such that the axis labels always held. Further, the data was sampled at a constant frequency of 20 Hz, so the resulting 10 second windows had 200 measurements.

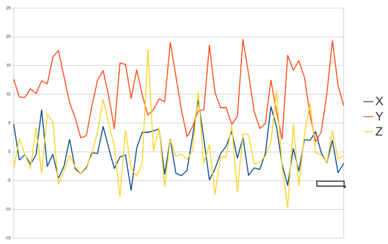

Figure 1: 3 seconds of raw accelerometer data for a user

In addition to the advantages of being highly controlled and labeled, this dataset has the advantage of having a performance benchmark in that Kwapisz, et. al. used this dataset to achieve 91.7 correct behavior prediction rate. The approach used was a multilayer perceptron algorithm using an ad-hoc feature set from each of the windows, including average acceleration, standard deviation, peak-to-peak time, and a histogram of the data.

We had to modify the Actitracker Dataset slightly for our purposes. First, we found some issues with duplicate data, as well as obviously irrelevant data (data with 0 timestamp). This data was removed. Second, we split the dataset into separate data series, by user and then by activity. Finally, we split each of these separate datasets into a train section (the first half of the file) and a test section (the second half of the file).

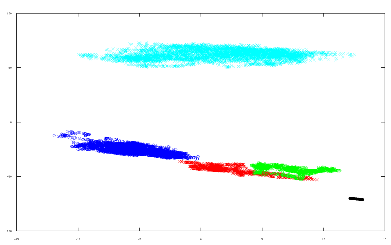

Figure 2: Dimensionality reduced plot of the 7 direction-agnostic features (PCA eigenvalues and length statistics). Black represents standing, the blue walking, the red upstairs walking, the green downstairs walking, and the cyan jogging.

3.2 PCA-Based System

As stated previously, our algorithm took raw time-series data and converted it to feature space. We initially did this using Principal Component Analysis, or PCA, on the raw values of the multitude of 200-sample windows. We used the eigenvalues and eigenvectors from this process as features for the classifiers. In an attempt to improve classification, we attempted to add other features, such as the products and sums of the eigenvalues, however we concluded that this was ineffective in improving classification, yet greatly slowed the classification process. Therefore we deemed this ineffective.

3.3 Ad-Hoc Features and Spectrograms

In attempt to add more features, we added four ad-hoc features to our classifier. First, we considered the lengths zero-meaned accelerometer vectors and created a feature for the mean and standard deviation of this value. Second, we created similar features from the raw lengths of the vectors. These four features were direction-agnostic and easily calculated.

We also extracted features from a fast-Fourier transform (spectrogram) of each of the individual accelerometer data series. For each of these axes, we found average of the most prevalent frequency at every point in time. The calculation of these three features involves a great deal of additional computation.

3.4 Direction Agnosticism and Training Methodology

In selection of features, we considered two variables in addition to efficacy: computational complexity and direction-agnosticism. Obviously, it is important that the features not be too complex to evaluate otherwise that they would tax the device and thus shorten battery life. Direction agnosticism is important because the user may place the phone in their pocket in any orientation. It is important to note that these direction agnostic features are not invariant to all rotations of the phone - during the 10 second window rotations may still impact the feature values. However, if the phone is rotated for the whole window then the feature values will not be changed. The direction agnostic features consist of the PCA eigenvalues and the length values. The PCA eigenvalues are direction agnostic because PCA eigenvalues are invariant under rotations of the initial dataset, while the length values are direction agnostic because of basic geometry.

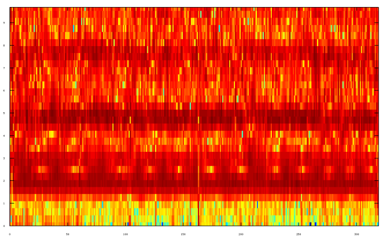

Figure 3: Spectrogram of one axis of accelerometer data

Given these criteria, we split our data into 3 sets of features: the simple direction agnostic features (PCA eigenvalues and the lengths), the simple direction dependent features (PCA eigenvectors), and the FFT features. We ran the classifiers on each of these categories but included features from the less burdensome categories as well. Additionally, we trained the classifiers in two approaches - in one case, we trained a classifier for each user and in the other case we trained one classifier for all users. Thus, we had a total of 6 trials for each classifier for a total of 12 trials between the SVM classifier and GDA classifiers.

4 Results

We present our results in terms of tables for success of the various classifiers on our dataset, split up as 50% train and 50% test.

Name	Individual	Global	Individual Simplified	Global Simplified
GDA (non-directional)	89.582%	72.509%	92.011%	73.248%
GDA (directional)	73.838%	78.633%	75.078%	79.440%
GDA (spectrogram)	71.834%	77.875%	73.046%	79.239%
SVM (non-directional)	90.629%	X	92.222%	X
SVM (directional)	86.698%	X	88.844%	X
SVM (spectrogram)	88.952%	X	91.196%	X

Table 1: This table shows the percent of test cases correctly identified by each of the classifiers in each of the datasets. The "Individual" columns mean that the classifiers were individually trained for each user while the "Global" columns mean that the classifiers were trained for all users. We found that a common source of confusion was the upstairs and downstairs results, so the "Simplified" columns represent the case where such misclassifications between Downstairs and Upstairs are ignored. An X represents that classifier failed to converge in this case.

	Total	Downstairs	Jogging	Sitting	Standing	Upstairs	Walking
Downstairs	20.23%	0.00%	0.12%	0.00%	0.00%	9.03%	11.08%
Jogging	4.29%	0.00%	0.00%	0.00%	0.00%	0.20%	4.09%
Sitting	12.86%	0.00%	0.00%	0.00%	12.86%	0.00%	0.00%
Standing	6.78%	0.00%	0.00%	6.78%	0.00%	0.00%	0.00%
Upstairs	21.53%	9.66%	1.67%	0.00%	0.00%	0.00%	10.21%
Walking	7.26%	4.39%	0.04%	0.00%	0.00%	2.84%	0.00%

Table 2: Confusion matrix for the non-direction individually-trained SVM (92.222% overall success rate). The percentages represent the percent of misclassified events labeled as a given row misclassified as a give column (the total column is the sum of all columns).

5 Conclusion and Future Work

We successfully developed a high performance classifier for recognizing mobile device user activity. The classifier achieved good results except in the case of confusing upstairs and downstairs, and to a lesser extent stair-walking and level-walking. Despite being direction-agnostic, the classifier performed similarly and even better than direction-dependent classifiers. In addition, our classifier is efficient as the PCA and related features consist of very few operations. In addition, our SVMs generally had very few support vectors (on the order of 10-20), meaning that applying the SVM classification is efficient and maybe possible in real time.

To improve the system accuracy, we hope to develop a system to distinguish stair-walking and normal-walking. Based on previous experience of the author, we believe that the integration of accelerometer data fused with magentometer and gryo data (standard on modern mobile devices) can predict elevation change over the course of a window. This would improve our results. Unfortunately this is not possible with the current dataset because it does not contain gryo or magnetometer data.

In order to continue this project, we must actually develop an application for mobile devices (as all our testing was performed off-line) and actually attempt to recognize activities. We also hope to experiment with different phones, users and additional activities, such as biking, driving, and riding public transportation. Finally, we would like to distribute this codebase as a library for other developers to use in their applications.

References

[1] Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore, *Activity Recognition using Cell Phone Accelerometers.* SIGKDD Explor. Newsl. 12, 2 (March 2011), 74-82.

[2] T. Scott Saponas, Jonathan Lester, Jon Froehlich, James Fogarty, James Landay, *iLearn on the iPhone: Real-Time Human Activity Classification on Commodity Mobile Phones.* CSE Tech Report UW-CSE-08-04-02

[3] Moves. ProtoGeo. http://www.moves-app.com/.

[4] Recognizing the User's Current Activity. Google.
http://developer.android.com/training/location/activity-recognition.html.