Mobile Activity Detection

Anonymous Author(s)
Affiliation
Address
email

1 Abstract

We aim to be able to categorize human movements using the sensors typically found on a phone. We first worked with a dataset collected by another project. Using these sets, we recreated the functions they used to generate feature vectors, and tested various learning mechanisms. From this, we decided to use a multi-svm to classify our data. We have created a small application to collect mobile sensor data from our phones, and plan to use that to create our own dataset. We are still experimenting with improving our feature space, collecting data, and improving our learning algorithms.

2 Introduction

Our project is about detecting user activity using mobile sensor data. Most mobile devices come equipped with an accelerometer and a gyroscope. When a person does their daily activities, their movements depend on what actions they take. The movements made while walking are subtly different from the ones made while walking up stairs, which are also quite different from the movements made while sitting. We aim to categorize these various types of motion, using only accelerometer and gyroscope readings.

2.1 Benefit to Society

We think that this project can create value in Context Aware Computing and for measuring human performance. For instance, if the system is aware of a user's current activity, it can tailor itself to match the user's needs better. The system can understand that a user is sitting at home or at the office. Based on such information, the system can make necessary adjustments such as changing the light settings in the room or bringing up a task list. For instance, a cell phone can start playing a user's favorite playlist and start a running app once the user starts to run. A cell phone may want to switch between vibrate and a ringtone depending on whether their user is sitting or walking.

Our system can also be used on athletes and physio patients to measure their performance. If we are able to successfully record the movements of a healthy runner and an injured runner, we may be able to detect injuries before they worsen. Similarly, the system can also be applied to assistive systems for the disabled. We can notify caretakers if an elderly person is having trouble climbing stairs.

3 Background

There have been various attempts at Human Activity recognition in research. Approaches have ranged from using ambient light and infra-red sensors to computer vision[3]. Some of these sensors are body worn, while some are external[4] to the user. There has also been some work [5][6] which proves that body pose and orientation can be estimated using acceleration measurements on the rigid parts of the body.

Since smartphones have become ubiquitous in our daily life, there has been significant research in utilizing data from its sensors.[2] provides a more comprehensive overview of the current state of the field. It also compares various classification algorithms, and stresses more focus on Hidden Markov Models. There has also been some work in figuring out the best approach for feature selection.[7] Approaches involve breaking the acceleration signal into high-frequency AC components for dynamic motion and low frequency DC components for static posture estimation[8]. Both, statistical measures such as variance, signal entropy etc. and frequency domain features such as Short Time Frequency Transform (STFT) and Direct Wavelet Transform (DFT)[9−10] have been used. [1] talks about a novel way to use SVM with fixed-point arithmetic for classification so as to reduce the computational cost on resource limited cell phones. It also hints at some methods of feature extraction.

3.1 Our Approach

Our approach was to first test our learning algorithms over some clean prepared data. We found several datasets online (UCI Machine Learning Repo and Kaggle) which have data from sensors annotated with activity. After some deliberation, we chose to focus on the UCI dataset. This dataset should serve as a good starting point, as it will test if our code works on well tested data. We didnt want bugs in our collection to give us trouble at the first stages of our project.

Later on, we will acquire an actual device to generate data ourselves. We will record ourselves doing various activities and then label the data. This data may require a significant amount of cleanup. The trial datasets describe the process they used to clean their data - we have already incorporated some of it in cleaning up our data and will choose the final setup based on experimental evaluation. We can then apply the same algorithms we used on the trial dataset to detect activity from cut sets from real world data.

Eventually, we plan to expand our algorithm to run in real time. Our algorithm will detect changes in activity and use our pre-learned classifier to output our activity in real time. So far we have we have modeled time series information by taking fixed-width duration readings and running classifiers like SVM on it. We would want to expand it to HMM based modeling to map different user states.

3.2 Project Scope

Our project is well within the scope of the term. We were able to able to get decent results running classification over the featurized datasets we found online. We were also able to recreate a good amount of the feature functionality they used, and get a good classification running over their raw data. We were hoping to get a specialized sensor for the experimental part of our project. This sensor has a long battery life and is light weight. Even if we do not get the sensor in time, getting readings is not difficult. As weve described, nearly all android phones come equipped with accelerometers and gyroscopes.

4 Our Method

For our first tests, we worked with the UCI Human Activity Recognition Using Smartphone Data data set. This dataset was collected in a series of experiments over 30 volunteers. These volunteers wore waist mounted smartphones which had accelerometers and gyroscopes. They were instructed to perform several activities of daily living (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING). The accelerometers and gyroscopes captured linear acceleration and angular velocity for all three dimensions at a constant rate of 50 HZ.

The data was then cut into 2.56 second portions with 50% overlap. Using a Butterworth low pass filter, they then split up the accelerometer data into its gravity component and a body motion component. They assume that the gravity will have a low frequency component, so a filter with .3 Hz cutoff would suffice. They then compute a large number of features over these timeslots, creating a feature vector with 561 elements. 70% of the data was then put into our training data set, and the remaining 30% was put into our test data set. For all of our following tests, we trained over the data they labeled as training data, and tested over the data they classified as testing data.

Our first approach to classifying our data was to simply run a support vector machine. We used the LIBSVM library, mainly because it had many features built in that were not in the matlab svm implementation. One major feature it had was that it had built in support for multi-svms. The comparison is done using 1-against-1 classification. For this mechanism, we first solve the pairwise classification problem. For every pair of classes i and j, we solve the optimization:

$$Min_{w^{ij},b^{ij},\xi^{ij}} \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \xi_t^{ij}$$

$$(w^{ij})^T\phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij} \; if \; y_t = i$$
$$(w^{ij})^T\phi(x_t) + b^{ij} < 1 - \xi_t^{ij} \; if \; y_t = j$$
$$\xi_t^{ij} \geq 0$$

Where $\phi(x_t)$ is a function that generates a set of feature vectors. Once we have computed each of these pairwise classifications, we use the voting strategy to pick a class for each point. In this strategy, the class selected is the one that is chosen by the most pairwise classifications.

$$Max_i \sum_{j \neq i} sign((w^{ij})^T\phi(x_t) + b^{ij})$$

Using the support vector machines implemented in the library, we were able to get a pretty good classification on the UCI data. We were able to correctly classify 94.0278% of the data. For the sake of comparison, we tested various learning systems.

System	Parameters	Test Err
K-NN	(euclidean, n=10)	90.5
K-NN	(correlation, n = 15)	91.2
Naive Bayes		76.9
Multi-SVM	(C =1)	94.0
Multi-SVM	(C =100)	96.2
Multi-SVM	(C =1000)	96.6
Multi-SVM	(C =10000)	74.7

Based upon our results, we decided that we would get our best results working with a multi-SVM with C = 1000. We also tested various kernels, but found that none of them improved our classification.

Our next step was to create a feature generation function ϕ that could give us similar classification. The UCI Dataset described all 561 features they generated in the some of the files they included with their dataset. As described earlier, the raw vectors they worked with were the gyroscope readings in 3 dimensions (tGyro-XYZ), and the accelerometer readings broken up into the parts caused by gravity (tGravityAcc-XYZ) and the parts caused by human motion (tBodyAcc-XYZ). They took the total magnitude of tGyro and tBodyAcc using the euclidean norm, creating two more signals. They then derived each of these vectors in time to create jerk signals. Finally, they took fourier transformations of some of these signals. In the end, they had generated a total 17 signals.

To convert these signals into vectors, they generated variables using the following functions (as taken from their dataset):

- mean(): Mean value
- std(): Standard deviation
- mad(): Median absolute deviation
- max(): Largest value in array
- min(): Smallest value in array
- sma(): Signal magnitude area
- energy(): Energy measure. Sum of the squares divided by the number of values.
- iqr(): Interquartile range

- entropy(): Signal entropy
- arCoeff(): Autoregression coefficients with Burg order equal to 4
- correlation(): Correlation coefficient between two signals
- maxInds(): index of the frequency component with largest magnitude
- meanFreq(): Weighted average of the frequency components to obtain a mean frequency
- skewness(): Skewness of the frequency domain signal
- kurtosis(): Kurtosis of the frequency domain signal
- bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
- angle(): Angle between two vectors.

Once each of these features were computed they were then normalized to be between -1 and 1. We created functions for every one of these features except for band energy. However, we were not able to fix all of the bugs in our implementation of these features. We found that some of the values we generated were different from the values in the UCI featurized vectors. However, we were still able to classify the data pretty well using a subset of these features.

4.1 Data Collection

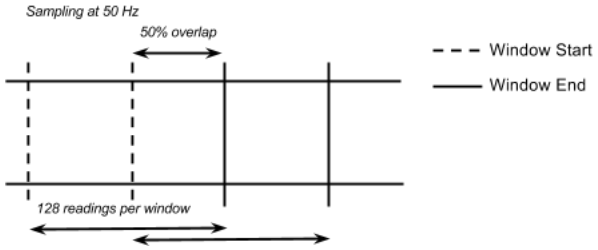
While we have had success generating and testing various models on data from repositories such as UCI Machine Learning, we have to remember that these datasets are partly pre-processed and may be specific to the condition to collection. We want to ensure that the models generated by us work on real world data collected from our phones. To that end, we developed a simple Android application which allows us to extract raw data from the phones sensors.

The android application collects 4 types of values

1. Total acceleration along 3 axes.
2. Gravitational acceleration along 3 axes.
3. Linear body acceleration along 3 axes.
4. Body Gyroscopic rotation along 3 axes.

We have the ability to tune the data collection using three variables.

1. SAMPLING_RATE - Time interval between two recordings of reading (in microseconds). For 50 Hz, it comes to $1/50 * (10^6)$ = 20k microseconds
2. WINDOW_SIZE - The number of readings we capture for the computation of 1 set of observations. Chosen to be 128, to capture 2.56 seconds worth of readings.
3. OVERLAP - This metric defines the extent to which we slide our window of feature extraction, or what percentage of the previous observations do we use in our current observation. By default it is set at 50



5 Results

As described earlier, we found that we were able to best classify the UCI data using a multi-svm. We were able to classify the data with 96.6% accuracy. When we attempted to learn over our own feature

vector we got decent classification, though we performed worse than the data sets classification. We found that adding many of our features would actually increase the test error. In the end, the best classification we got did not use any of the features generated from the frequency domain and did not use the angle feature. This gave us 91.0% accuracy.

We also briefly experimented with frequency domain features, but noticed that they were leading to overfitting of the data. We attempted to smooth it out by using moving averages and Gaussian kernels (with varying window sizes), which helped improve our test accuracy, but did not perform well enough. We plan to invest more time and effort into using frequency domain components effectively.

6 Updated Timeline

Date	Milestone
Nov 25	Application of HMM to data
Nov 27	Data Collection
Dec 1	Poster Submission
Dec 3	Simulations on HMM/SVM. Selection of Method.
Dec 5	More feature engineering and experimentation with features.
Dec 7	Modeling in prior information in decision making
Dec 9	Performance optimization
Dec 10	Final Report preparation

References

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012

[2] Mannini, A.; Sabatini, A.M. Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. Sensors 2010, 10, 1154-1175

[3] Poppe, R. Vision-based human motion analysis: an overview. Comput. Vis. Image Underst. 2007, 108, 418

[4] Su, M.C.; Chen, Y.Y.; Wang, K.H.; Tew, C.Y.; Huang, H. 3D arm movement recognition using syntactic pattern recognition. Artif. Intell. Eng. 2000, 14, 113118

[5] Padgaonkar, A.J.; Krieger, K.W.; King, A.I. Measurement of angular acceleration of a rigid body using linear accelerometers. ASME J. Appl. Mech. 1975, 42, 552556

[6] Cappa, P.; Masia, L.; Patane, F. Numerical validation of linear accelerometer systems for the measurement of head kinematics. J. Biomech. Eng. 2005, 127, 919928

[7] Sabatini, A.M. Inertial sensing in biomechanics: a survey of computational techniques bridging motion analysis and personal navigation. In Computational Intelligence for Movement Sciences: Neural Networks and Other Emerging Techniques; Begg, R.; Palaniswami, M., Eds.; Idea Group Pubilishing: Hershey, PA, USA, 2006; pp. 70100

[8] Veltink,P.H.;Bussmann,H.J.;deVries,W.;Martens,W.J.;VanLummel,R.C.Detection of static and dynamic activities using uniaxial accelerometers. IEEE Trans. Rehab. Eng. 1996, 4, 375385.

[9] Bao, L.; Intille, S.S. Activity recognition from user-annotated acceleration data. In Pervasive Computing; Springer Berlin/Heidelberg: Berlin, Germany, 2004; pp. 117

[10] Mantyjarvi, J.; Himberg, J.; Seppanen, T. Recognizing human motion with multiple acceleration sensors. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ, USA, October 710, 2001; pp. 747752.