Project 2: Detecting Daily-Living Activities with Wearable Sensors

CS 7626/4403 Introduction to Behavioral Imaging March 25, 2018

Spring 2018

Instructor: Prof. James M. Rehg

Lead Teaching Assistant: Grady Williams

Background

Activity detection is a basic task in behavioral imaging, because the measurement of behavior is a key capability in understanding the link between behaviors and health outcomes and in devising interventions for effective behavior change. In Project 1, we investigated the detection of children's social behaviors (specifically, pointing gestures) using video. In this project, we are going to explore the detection of activities of daily living by adult participants using wearable sensors. This is an increasingly important application due to the widespread adoption of wearable devices, such as smart watches, incorporating multiple sensing technologies.

For this project, we will use the publicly-available OPPORTUNITY Activity Recognition Dataset [1]. The dataset contains signals from various on-body(wearable) sensors, ambient sensors and object sensors. We will work with a subset of the on-body sensor signals. From the Opportunity Dataset, two class of activities will be considered. The first class is activities related to locomotion. The second class of activities is related to gestures. The complete list of defined activities defined is given in Table 1 below. The subset of sensors from the OPPORTUNITY dataset that we will utilize in this project is the sensors circled in Figure 1 below.

Table 1. Activity Definitions for the Assignment.

Activity Number/Activity Class	Locomotion	Gesture
0	Null Activity	Null Activity
1	Stand	Open Door
2	Sit	Close Door
3	Walk	Open Dishwasher
4	Lie	Close Dishwasher
5		Open Drawer
6		Close Drawer
7		Clean Table
8		Drink Coffee
9		Toggle Switch

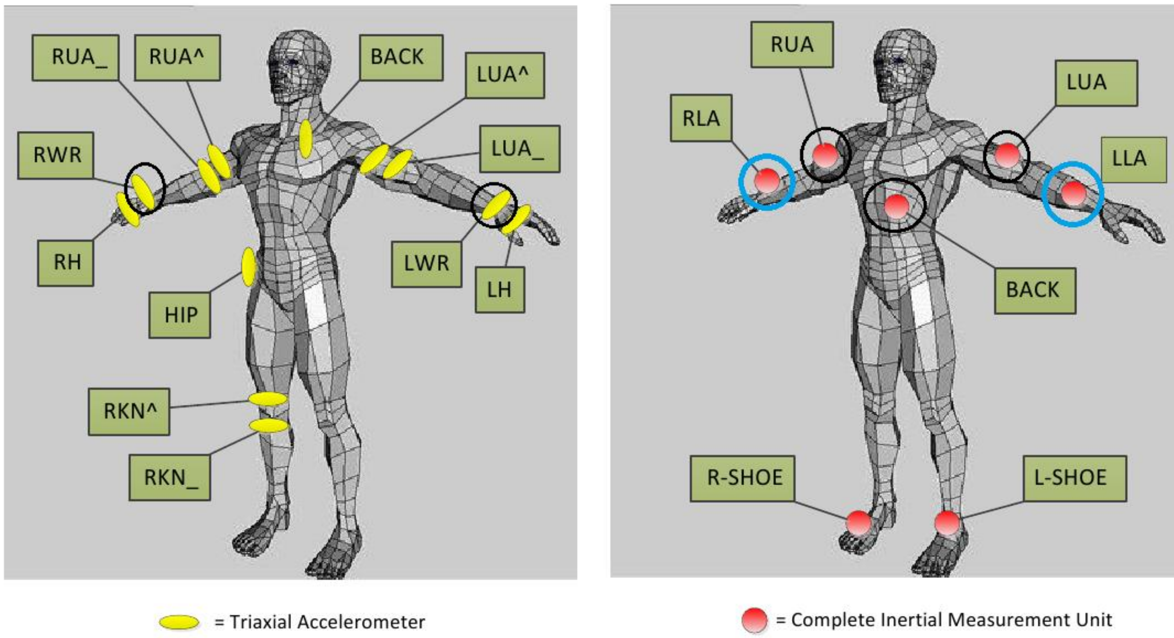

Figure 1. Sensor Placement.

Three sets of sensor data are going to be considered for this project. The first one is composed of only triaxial accelerometers located on the wrist. These are circled in left panel of the Figure 1. This set is called “3D Acceleration Wrist Sensors” hereafter. The second set adds IMU sensors from the forearm to the first set. The IMU sensors to be added are circled in blue in the right panel of Figure 1. This second set is referred to as “IMU Wrist Sensors.” The last set of sensor data incorporates in addition the black circled IMUs in the right panel of Figure 1. This set is referred to as “Full-body Sensor Data.”

To get more familiar with the dataset, you can read the documentation page in the docs folder of the .zip file. Note that this is the original documentation from [1].

The dataset consists of recordings from 4 subjects. For each subject, there are 6 different .mat files containing the data. Five of them are from the activities of daily living runs, and the last one is from the activity drill run. In each .mat file, there is a matrix named “data” that stores the data. Each row of data is a sample. Each column is either time from the start of the recording or sensor sample or a label for the sample. The file column_names.txt gives information about each of the columns.

Procedures and Deliverables

The project has five parts, and each part has a deliverable associated with it. The deliverables are slides, and you must submit a single PDF file containing all the slides for your project on T-Square, along with your source code. See below for the submission instructions.

Preliminaries

For each of these parts, except part 1, you will be developing a classifier to predict the activity label for a window of data corresponding to an interval of time. The focus of the project, again, is developing a classifier but in this case, it is a multi-class classifier. You are free to use any modern classifier design for the classification task. Some valid choices are support vector machine (SVM), random forest, and boosted decision trees. Choices that are not allowed are simple, generally inferior methods like nearest-neighbor and logistic regression. You are free to use any software package or implementation of these methods. Scikit-learn is a good choice for Python, the Weka library is another option. If you have any doubts or questions about the classifier please post on Piazza.

For each of the five parts below, except part 1, you will perform leave one subject out (LOSO) cross-validation to assess the performance of your classifier. For each cross-validation split, you must compute the confusion matrix for the resulting classifier. Then sum confusion matrices from each of the splits. Then compute the F-score for your classifier, which will be the final performance measure.

Part 1: Setting-Up the Data

We will perform initial pre-processing of the dataset. There are two predominant problems that arise in working with wearable sensors: synchronization issues and data loss. Synchronization issues arise in the case where there is no central clock that determines the sample times. In that case, there is an ambiguity in the timestamps of the samples. In this project, we will assume data is perfectly synchronized. The second issue arises if the wearable sensors transmit data through some wireless channel. In that case, data loss is almost inevitable. You will address this problem in this project. In the data matrix, some sensor values have NaN for missing data. You need to impute data for those NaN values. However, for some of the runs, e.g. S2-ADL1, the final parts of the recording are all NaNs for the 3D Acceleration Wrist Sensors. If this is the case, you can just discard that part.

One way to impute missing data in the middle of a continuous sensor recording is through interpolation. For this assignment, you can use numpy.interp or the interp1 function of matlab. For this part, submit the imputed data for the S3-ADL3.mat

Part 2: Sliding Window Activity Detection from 3D Acceleration Wrist Sensors

Similar to part 3 in project 1, your classifiers will be designed to work over a sliding window. In the data folder, there are two files for each run. For the rest of the assignment, you will use the file ending with “proc.”

In this part, the 3D Acceleration Wrist Sensors will be used. For your classifier of choice, you will average the sample values in your window to extract the feature vectors. Your implementation should be parameterized, so the window can be of variable length and variable stride. Window length is the number of consecutive samples that are combined to perform classification. Window stride is the number of samples separating the start of each window position. So, with a length of 5 and a stride of 2, first window position would process samples (1, 2, 3, 4, 5) and the second window position would process samples (3, 4, 5, 6, 7). In order to compute the accuracy of your method, you will need to assign each window position a ground truth label of the activity. This will be done by computing the overlap between the sliding window position and the annotated activity labels. For the ground truth label of the window, you can choose it as the majority of the activity labels of the samples.

For this part, you will report two F-scores, one for the locomotion activity class and the other one for the gesture activity class.

Part 3: Sliding Window Activity Detection from IMU Wrist Sensors

This part is very similar to Part 2. In this part, you will use the IMU Wrist Sensors. This is the only difference.

Report the same results as in part 2.

Part 4: Sliding Window Activity Detection from Full-body Sensor Data

This part is very similar to Part 3. In this part, you will use the Full-body Sensor Data. This is the only difference.

Report the same results as in part 2.

Part 5: Feature Extraction for Activity Detection

This part is like Part 4. In this part, rather than only averaging sample values in a window, you will implement some standard features that are used in activity classification. For each window extract the following features:

Let $\vec{y}$ be the samples of from a window and n be the number of samples in a window. Extract the following features:

-RMS Value: $\sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i^2}$

-Standard Deviation: $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$ where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the average of the sample values.

-Mean Absolute Deviation: $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}|y_i - \bar{y}|}$

-Relative Power in a Frequency Band:

Let $\vec{x}$ be the Fast Fourier Transform (FFT) of $\vec{y}$. Divide the frequency range into M bands. For each band calculate the relative power with the following formula:

$$\mathrm{RP_i} = \frac{\sum_{\mathrm{j=FFT\ coefficients\ in\ the\ i'th\ band}} |x_j|^2}{\sum_{\mathrm{j=1}}^{\mathrm{L}} |x_j|^2} \ for\ i\ =\ 1, \ldots, M$$

For this part, you can try different FFT lengths and different numbers of frequency bands. After you extract these features, use them in your classifier.

Report the same results as in part 2.

Part 6: Final Detector

In this part, you need to try at least one additional idea, beyond what you have already done in parts 2-5, to improve the detector. This could be an additional feature. Provide a justification as to why that feature is helpful in classification.

References

[1] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Gerhard Tröster, Paul Lukowicz, Gerald Pirkl, David Bannach, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, and José del R. Millàn. "Collecting complex activity data sets in highly rich networked sensor environments" In Seventh International Conference on Networked Sensing Systems (INSS'10), Kassel, Germany, 2010.