Sensors can't tell whether you are walking, running, or napping. But in this assignment, you will teach them to classify the raw sensor data into meaningful activities. This can be done by having a smartphone in your pocket that can collect different types of motion data as you are doing different activities. From this motion data, you will then build a classifier to infer back what actions you were doing.

1 Instructions

Come to Jeff or Nedi's office hours for help with this assignment if you need it!

1.1 Setup

Download a motion sensor logging app on your phone. Two suggested options are PowerSense on iOS and AndroSensor for Android. Set it to capture accelerometer data at 0.1 second intervals (100 Hz). You may optionally capture gyroscope, compass, and noise level data as well, but they may be less useful. Test it out so you can collect motion sensor data into a CSV and move it to your computer somehow.

1.2 Collection

Do each of these activities for 5 minutes each with the phone in your pocket or attached to you:

1. **walking**
2. **running** or jogging
3. climbing **stairs**
4. browsing the web while **sitting**
5. being in a **vehicle**

It is recommended to start doing each on ten-minute marks so it's easy to keep track of, like 2:10pm or 11:45am. If you can't do any of these activities for 5 minutes, ask a friend or train your pet (you can attach the phone to the pet). For each activity, the first 4 minutes will be *training data*, and the second 1 minute will be *test data*.

Look at just the first 4 minutes of each activity. You can plot them as line graphs in Excel. You should have one line for each combination of sensor and axis.

It would be a good idea to finish up to here before spring break. The rest of the assignment can be challenging. Even setting up sklearn and Jupyter Notebook (see below) can take a day.

1.3 Make Features

Now the creative part—coming up with features that differentiate between activities. Each 10 seconds of data is considered a data point, so you will need to eventually infer the activity for each 10-second period. Therefore, each activity will have 24 data points of training data, and 6 data points of test data. There are 5 activities so you will have 30 10-second periods of test data.

You will be writing code that takes as input 10 seconds of raw sensor data, and outputs a handful of features which are numbers. Some example basic features are: mean, standard deviation, skewness, number of times the sensor reading exceeds a particular threshold, values at different quartiles. Also consider putting the data through a discrete fast fourier transform function, which changes time-series data into the frequency domain; this will give you another set of features to use. Be selective with your features! You probably don't want too many features, because you are only training on 24 data points. Pick ones that make sense after looking at the sensor reading plots. You can use features for any of the x, y, or z axes, and for any of the above sensors mentioned (accelerometer, gyroscope, compass, noise).

1.4 Classification

Use the python machine learning library called sklearn to do the classification of features from your data into activity categories. Try logistic regression, and two of (support vector machines, k-nearest neighbors, or decision trees). So you will have tried 3 supervised learning algorithms in total, and can compare which ones are most effective.

Here are a couple of tutorials for doing this on a different dataset (the Iris flower dataset): decision tree (just the classification section) and Logistic Regression, k-Nearest Neighbors, and Support Vector Machines. Run through the tutorials to get familiar with sklearn.

Jupyter Notebook (formerly caleld IPython Notebook) is a nice way to explore data in python. At the end, evaluate how accurate your classifier is by running each classifier on all the test data. Which activities does it most correctly classify? Use that to do your work—it will make things a lot easier, and then share through Github the link to @jeff on Slack. You can just present or show & tell your Jupyter Notebook page.

At the end of this, think a bit about which features were useful, and which learning algorithms were effective. Why do you think so?

2 Bonus challenge

Lay down and place your phone on your chest. Capture 10 minutes of normal breathing and heart beats. Can you write a function that takes this data to figure out your respiratory rate (breaths per second) or heart rate?

This assignment is worth 10 points, with a possible 2 extra points for the bonus challenge.