Group 5: Gahyun Kim, Thai Nguyen, Colin Stern
CS390MB

A1: Step Counter Writeup

1) How well did the built-in Android step detector work? Did it work for some phone positions and not for others, e.g. in your pocket or in your hand?
    - Overall, the built-in Android step detector worked well. However, its performance varied depending on the phone position when walking. After testing the step detector in various phone positions, we found that the built-in Android step detection worked best when it was in our pants pocket and was less accurate when it was in our hands and our backpacks. We believe that the step detection works best when it was in our pants pocket because it is closest position to our leg and receives a more accurate accelerometer data. It does not work well when the phone in our hand because our upper body stabilizes the change in direction and speed, causing the received accelerometer data to be less accurate.

2) Which signal filter did you choose and why? What parameters did you select?
    - We chose Butterworth instead of smoothing filter because we want to filter by frequency, given the fact that we know the maximum frequency of the steps is two hertz.

3) Describe in words the details of your step detection algorithm.
    - Given the raw data from the accelerometer we use a Butterworth filter to remove all signal that is above two hertz. This filters out noise that looks like very rapid steps. We then convert the values into floats for easier computation. Next we square each data point and add them to produce a 3D vector value so as to reduce the impact of orientation on the algorithm. Then we append the current timestamp to the array and check if the size of the array is greater than the rate at which the algorithm runs (the sample rate). If so, we compute the average of the values in the array by taking the max and the min value and dividing by two. Next, we check for how often the accelerometer data makes an upward crossing across the average - we consider this a step. We do this by checking if pairs of adjacent values are increasing, and if so, we check if the first value is below the threshold and the second value is above the threshold. If all these conditions hold true, we register a step by calling onStepDetected and pass in the timestamp of the step and the accelerometer data that produced it.

4) Does your step detection algorithm work for some phone positions and not for others? It's perfectly OK if that's the case. Talk briefly about why you think that might be the case.
    - It works equally well for all orientations because we square each coordinate and add them to get the magnitude of the total acceleration of the phone. While not perfect, this enables us to ignore orientation entirely. Possible bugs include shaking the phone back and forth while texting or holding it and other body movements which do not constitute a step.

5) Did you have any trouble with Python? Any server connection issues? Any recommendations or concerns regarding the project?
    - We had server connection issues Thursday and posted in Piazza.

6) **Extra Credit**: **Step Count to Calories**: We computed the number of calories burned from the number of steps taken (using the number of detected steps from the built-in Android step detector). The number of calories burned is displayed on the top of the My Exercise page of the Application, next to the step count display.