Android app, which leverages Support Vector Machines to predict your activity

Overview

The objective of this project is to build an application which recognize human activities using machine learning technique. To recognize the activities, gathering datasets related to human actions in daily life would be required. Fortunately, we could solve this gathering issue using the smartphone. Nowadays, most people have their smartphone and the smartphones have many sensors such as accelerometer, gyroscope, orientation, GPS, proximity, etc. By using these sensors, we can obtain the datasets required for identifying human movements.

Part A

In this part, I generate the training database file to classify three different activities: walking, running, and eating. Each activity duration should be 5 seconds. The data sampling frequency should be over 10 Hz for the proper accuracy. Also, the number of each activity in the training dataset should be over 20. According to recent research papers, accelerometer sensor datasets (X, Y, and Z axes) are very useful for human activity recognition, so we suggest to use these accelerometer sensors as input dataset among many sensors in the smartphone. The below table is an example of database scheme.

ID	Accel X 1st	Accel Y 1st	Accel Z 1st	Accel X 2nd	……….	Accel X 50th	Accel Y 50th	Accel Z 50th	Activity Label

Part B

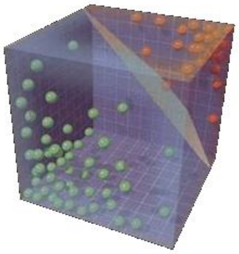
Based on the database which I generated (Part A), my application should classify the activities using Support Vector Machine. You can download free java SVM library from (http://www.csie.ntu.edu.tw/~cjlin/libsvm). You can download the android SVM application using the library (https://github.com/cnbuff410/Libsvm-androidjni). Also, it is fine to use other third party library or implement it yourself if you want. In your app, the SVM parameters and test accuracy should be displayed. The accuracy should be over 60%. For the validated test accuracy, we suggest to use 'K-fold cross-validation technique' which is supported by the library. The 'K' should be between three and five.

Future Work:

(a) Data visualization

Draw all dataset in 3D plane and display the decision boundaries in app.

i.e.) Green dot (running data), red dot (walking data), and 3d plane decision boundary.



(b) Performance

Draw graphs which compare “Part B” execution time and power consumption when working on server and when working on mobile in the app. When you get the execution time and power consumption of server, you should consider uploading the database and downloading the result. Therefore, you should draw two graphs (mobile and server) whose X axis is the execution time and Y axis is the power consumption in the same android activity.