An activity based music application for android

Pieter Bamps
Promotor: Prof. dr. E. Duval
Department of Computer Science
Katholieke Universiteit Leuven, Belgium
pieter.bamps@student.kuleuven.be

Abstract

This paper will discuss the development of an activity based music application for the mobile platform android. The application is mainly focused on the activities of students. The application, described in this paper, will be able to detect 4 different activities by using an ad-hoc classification method and couple a genre-specific music-playlist to these activities to offer the user the right music at the right time.

1. Introduction

Nowadays there are a lot of mobile music applications that try to offer the user a way for organizing large music collections. These either make use of a giant text-based list which is pretty chaotic(ITunes)(see [1] for an Itunes study) or they try to organize the music in some kind of fancy visualization structure: figure 1. [2] The main problem with these methods is that music collections, in general, are pretty big and mobile devices have rather small screens. This problem can be solved in different ways. In this paper I choose to solve this problem by offering the user only the music he really wants at the moment of listening. To do this, different genre specific playlists are coupled to the user's activity. Over time the user can personalize these playlists by labeling individual songs to these activities. To detect the different activities we make use of classification methods. A brief summary of all the research already performed on this specific learning task can be found in section 2. A summary of the different classification methods is discussed in section 3. Section 4 will briefly discuss the implemented algorithm. Section 5 describes some test results performend on the htc desire-z. Section 6 describes the final application and at last, in section 7, some future plans are described.

Figure 1. aweditorium

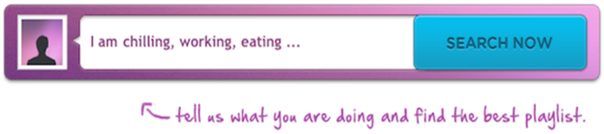
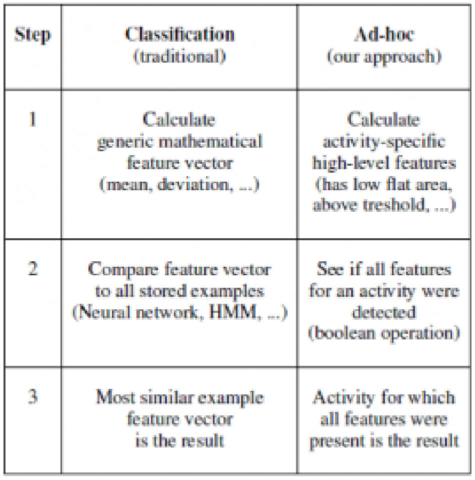

I am chilling, working, eating ...
SEARCH NOW
tell us what you are doing and find the best playlist.

Figure 2. Playlistnow

2. Previous work

There already exist some music applications that are focussed on the user's activity. ”Playlistnow”[3] is probably the most popular one.(see fig: 2) The difference with this application is that in ”playlistnow” the user has to type in his activity. The application isn't able to detect activities automatically. A second difference is that our application makes use of the Tunify webservers instead of the user's music library or youtube. There exist a lot of research in the field of activity detection. Most of this is based on accelerometers. In [4] they makes use of 3 motionbands(devices with a build-in accelerometer, magnetometer and gyroscope) that are attached to specific body parts to detect the user's activity. They were able to differentiate between 5 dif-



Step	Classification (traditional)	Ad-hoc (our approach)
1	Calculate generic mathematical feature vector (mean, deviation, ...)	Calculate activity-specific high-level features (has low flat area, above treshold, ...)
2	Compare feature vector to all stored examples (Neural network, HMM, ...)	See if all features for an activity were detected (boolean operation)
3	Most similar example feature vector is the result	Activity for which all features were present is the result

Figure 3. Classification vs Ad-hoc

ferent activities: resting, typing, gesticulating, walking, running and cycling with pretty high accuracy. In [6], they make use of three triaxial accelerometers which are worn on thigh, waist and wrist. The classification method used are based on decision trees. They were able to detect, with high precision, the folowing activities: sitting, standing, walking, lying, running and the folowing hand gestures: cutting, toot brushing, taking picture, shaking hands etc... . There is also research on other techniques to detect activities but these are out of scope because they make use of sensors that aren't always available for smartphones. In this paper, we make use of only 1 single smartphone with 1 accelerometer, 1 gps-sensor and 1 audiosensor so we can expect less precision than the results obtained in the previous mentioned work.

3. Classification Methods

3.1. Machine learning approach

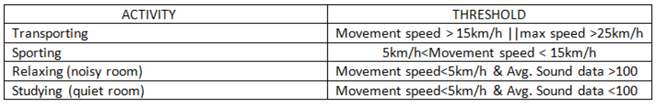
There exist 2 well known techniques to predict a user's activity.(see figure:3) There is the machine learning approach and the ad-hoc approach. The machine learning method [4] [5] [6] works by first monitoring all the incoming sensor data. It then calculates some mathematical features from it that can serve as input for the learning method. Neural networks, Simple vector machines and decision trees are the most popular. The test data gets clustered into different groups. The learner then classifies every new example to the most similar group to predict the correct activity. The pro's for this technique are that it works very accurate. There are a lot of different machine learning techniques that you can use to classify your example and there are a lot of papers about comparing these techniques against each other. The con's are that you need a lot of test data to make it work. The machine learning techniques are also pretty computational. So you need to send your incoming data to a server, process it, and then send the result back to the device. A lot of these machine learning techniques are able to differentiate between 20 and more activities with high precision because they use very accurate measurement devices that exceed the accuracy of a mobile phone.

3.2. Ad-hoc approach

This approach [7] starts by searching for specific thresholdsvalues. These values are used to differentiate between the different activities. Every new example is compared to these thresholdvalues and is then classified to the activity for which all features were present. The pro's are that you won't need as much training data as with the machine learning approach. The technique is also less computational-heavy so there is no need to work with an external server to do all the calculations. The ad-hoc approach is also much easier to implement. The con's are that this method isn't as general as the machine learning technique. It's also much harder to add new activities to the model because it's possible that you have to change all your thresholds. In this work we will use the ad-hoc approach because of the previous discussed pro's.

4. Detection algorithm

The first detection algorithm that was tried was based on the user's location. Studying takes place in a study room, sporting in a sport center or outdoors. Relaxing probably happens in the living room. 2 problems came up with this approach. Firstly, it is hard to detect the position of an user inside a building precisely enough. Secondly, there is the problem of multi-functional rooms. A lot of people that were enquired used the same room for different activities, especially students who, in general, live in small dorm rooms. In this simple version of the application the detection algorithm is only able to differentiate between 4 student activities. Relaxing, studying, traveling (transporting) and sporting. Most of the smartphones nowadays have a number of different sensors at their disposal.There is the camera, Wifi, a gps sensor, an accelerometer and gyroscope and even lightsensors. These are ideal to determine the user's context and environment. The problem with these sensors is that they consume a lot of energy so it's crucial to minimize the access time for



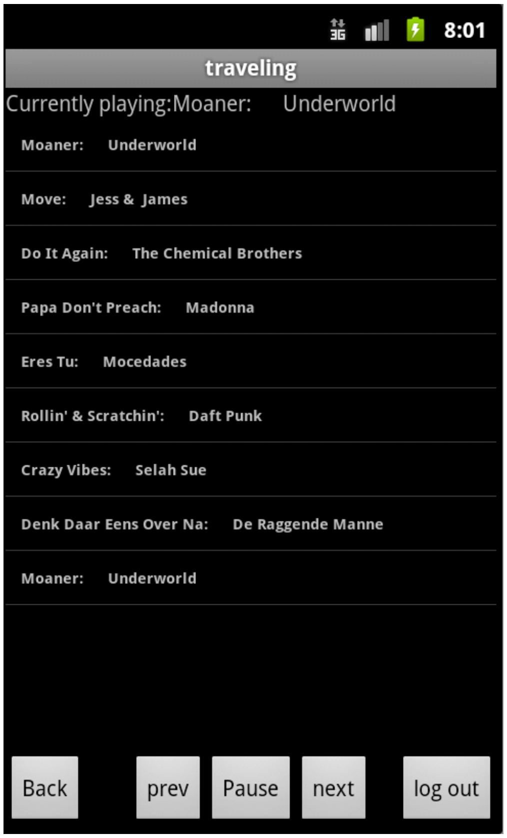
ACTIVITY	THRESHOLD
Transporting	Movement speed > 15km/h ||max speed >25km/h
Sporting	5km/h<Movement speed < 15km/h
Relaxing (noisy room)	Movement speed<5km/h & Avg. Sound data >100
Studying (quiet room)	Movement speed<5km/h & Avg. Sound data <100

Figure 4. detection thresholds

them. In this algorithm we tried to preserve as much energy as possible to save battery lifetime. The implemented algorithm that is used works in 2 steps. First, it activates the gps-sensor and waits till it has extracted 10 movement speed values. It then calculates the average of these values and differentiates between someone who is traveling/sporting and someone who is relaxing/studying and thus not moving. If the person is moving the algorithm looks at the average speed and decides with some thresholds if this person is either transporting (car/train/plane) or sporting.(see fig:4) In the second step the microphone is activated and starts recording audio data. Again this information is averaged and compared against the thresholds from the table. Note that this step is omitted when the person was either transporting or sporting because sound isn't useful then and we can save battery lifetime. When the activity is detected the genre-specific playlist the user selected starts playing.

5. Results

In this section we will summarize some results obtained from the htc desire-z. The gps sensor of the HTC desire-Z works pretty accurate. The gps is also tested inside buildings. The standard deviation from 5 minute long position queries was 0.0000571174 and the variance was 0.00000000326239 so there were no real problems with this. The max error rate was around 20 meters which is decent enough for the functionality this application has to offer. The only problem with the gps-sensor is that it sometimes takes a while before a signal is detected inside a building. This can be cumbersome and has to be fixed. The Android api offers a 'getSpeed()'-method that returns the moving speed of the device. I tested this while running, riding a bike and being in a car. The test results show that this method works pretty accurate. Section 7 briefly discusses how to improve activity detection by gps-sensor. The second sensor that is tested in this work is the microphone. This sensor will be used to differentiate between quiet and noisy rooms. The student will most probably study in a quiet room while relaxing takes place in a more noisier environment. The microphone of the HTC desire-Z was able to differentiate the difference between these rooms by averaging out 5 seconds of sound-data. In conclusion, the htc desire-z was able to detect around 90% of the test-cases (no real life situations are tested yet)



Currently playing:Moaner: Underworld

Moaner: Underworld

Move: Jess & James

Do It Again: The Chemical Brothers

Papa Don't Preach: Madonna

Eres Tu: Mocedades

Rollin' & Scratchin': Daft Punk

Crazy Vibes: Selah Sue

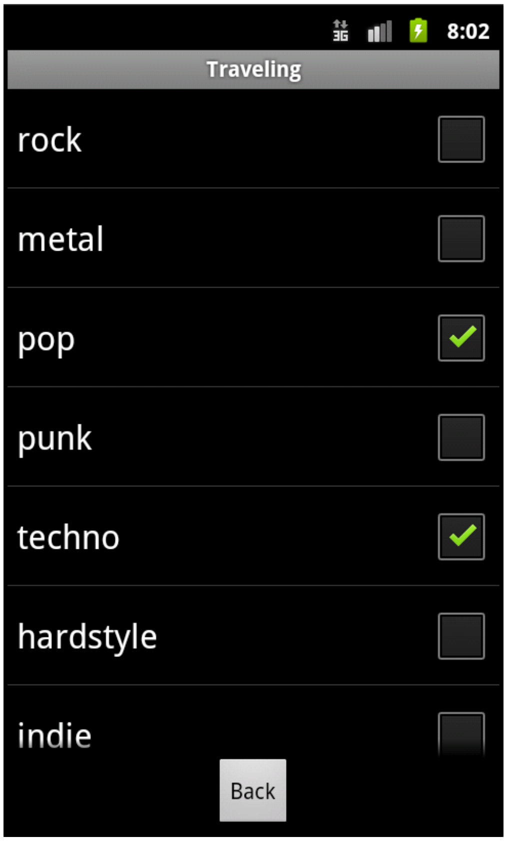
Denk Daar Eens Over Na: De Raggende Manne

Moaner: Underworld

Figure 5. playlist screen

6. The Application

The application consists of 2 important menus. There is the playlist screen(see fig: 5), which shows a list of songs of the automatically created playlist and there is the activity screen(see fig: 6), where the user can link the different activities to genres. The first time the user uses the application the generated playlist will be quite random. That's why there is an extra selection mechanism build-in. The user is able to label individual songs to an activity. This way he is able to personalize these random-generated playlists by including songs he prefers. The user interface is kept simple because the application focusses on a simple and user friendly design without relying much on user interaction. A screen transisition diagram of the application can be found in(figure: 7)



8:02

Traveling

rock □

metal □

pop ☑

punk □

techno ☑

hardstyle □

indie □

Back

Figure 6. activity screen

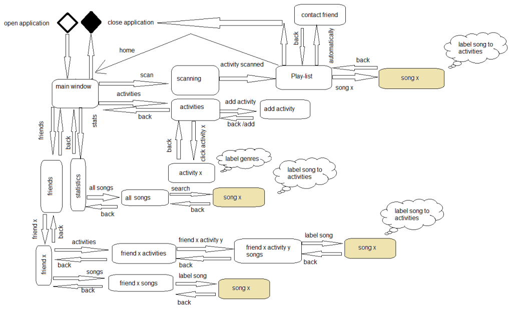


Figure 7. screen-transistion-diagram

7. Further plans

There are a couple of improvements that are planned for the near future. The first one is to also make use of WiFi to detect the user's location and movement speed. It sometimes takes a while for the gps to detect a signal and starts updating the location inside a building. To increase the detection speed of the application it can be smart to also search for a possible WiFi network to increase the detection speed. If this is not enough there is the option of just assuming the user is inside a building when no signal is found. Another improvement can be made in the transporting/sporting detection. The detection algorithms uses movement speed to differentiate between these. Actually this isn't very accurate. Someone can be training on a home trainer or in a gym and then there is no movement speed at all. To improve this it can be smart to also take the accelerometer into account. When someone is in a car or train his mobile phone will stay still in their pocket but when sporting there surely will be movement and this can probably be detected by this extra sensor. Another improvement is the detection of extra activities but these have to be chosen carefully because the application works with the ad-hoc method and it can be difficult to not influence the detection of the other activities with a new one. At last, there is the possibility to improve the user-interaction. At the moment the user always has to press a 'Scan-button' to start scanning the activity. This can be automated so no user interaction is needed. This implementation will rely on a detectionalgorithms that starts detecting at regular intervals(for example: at the end of every song) instead of just after the click on the scan-button. This will probably decrease battery lifetime but also user interaction. Usertests are planned in the near future to test the correctness of the detection algorithm and to test usability.

8. Conclusion

We have described a playlist application developed for the android platform, that couples specific music genres to a user's activity. The application is able to detect these activities automatically. The application automatically generates these playlists by using webrequests to the tunify-webserver. The user is able to label specific songs to an activity to personalise these playlists. At the moment, the application is only able to differentiate between 4 actvities but we plan to improve this in the future. At the moment of writing there weren't any user tests performed yet but these are planned in the near future. The results from a first alpha test were promising though.

References

[1] Tammy Greene, Jim Laing, Steven Laforest, Kunaal Pandya, Raj Vashi : Itunes evaluation and design, 1-34

[2] www.aweditorium.com

[3] www.playlistnow.fm

[4] Györbíró, N., Fábián, A., Hományi, G.: An activity recognition system for mobile phones. Mob. Netw. Appl. 14(1), 82-91 (2009)

[5] César Bernardini: Recognition of Actions That Imply Movement by Means of a Mobile Device with a Single Built-in Accelerometer.,1-9 (2010)

[6] Yu-Jin Hong, Ig-Jae Kim, Sang Chul Ahn and Hyoung-Gon Kim: Activity Recognition using Wearable Sensors for Elder Care, Second International Conference on Future Generation Communication and Networking, 1-4 (2008)

[7] Robin Marx: Ad-hoc accelerometer activity recognition in the iBal, 1-6 (2010)

[8] Uta Christoph, Janno von Stülpnagel, Karl-Heinz Krempels, Christoph Terwelp: Context Detection on Mobile Devices, 1-8