AMERICAN UNIVERSITY OF BEIRUT

SEMANTIC ACTIVITY RECOGNITION USING MOBILE PHONE SENSORS

by
CYNTHIA JOSEPH EL-HAYEK

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Computer Science
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
April 2015

AMERICAN UNIVERSITY OF BEIRUT

SEMANTIC ACTIVITY RECOGNITION USING MOBILE PHONE SENSORS

by
CYNTHIA JOSEPH EL-HAYEK

Approved by:

Dr. Wassim El-Hajj, Associate Professor — Advisor
Computer Science Department

on behalf of Dr. Safa, Wassim El-Hajj

Dr. Haidar Safa, Associate Professor — Member of Committee
Computer Science Department

Dr. Hazem Hajj, Associate Professor — Member of Committee
Electrical Engineering Department

Date of thesis defense: [April, 22, 2015]

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: El-Hayek Cynthia Joseph
Last First Middle

☒ Master's Thesis ◯ Master's Project ◯ Doctoral Dissertation

☐ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☒ I authorize the American University of Beirut, **three years after the date of submitting my thesis, dissertation, or project,** to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

Signature Date 08/05/2015

ACKNOWLEDGMENTS

AN ABSTRACT OF THE THESIS OF

Cynthia El-Hayek for Master of Science
Major: Computer Science

Title: Semantic Activity Recognition Using Mobile Phone Sensors

Recommender systems use contextual and non-contextual information about the user in order to make good and appropriate recommendations. User's emotion is one of the factors that play an important role in these recommendations and is affected by multiple elements such as the user's current activity. Consequently, knowing the user's current activity is essential for making appropriate recommendations. In this thesis, we address the problem of semantic activity recognition using data collected from the user's mobile phone. Our approach recognizes a large set of activities that are comprehensive enough to cover most activities users engage in. Moreover, multiple environments are supported, for instance, home, work, and outdoors. Our approach suggests a multi-level classification model that is accurate in terms of classification accuracy, comprehensive in terms of the large number of activities it covers, and applicable in the sense that it can be used in real settings. Hence, in literature, these three properties are not existent altogether in a single approach. Proposed approaches normally optimize their models for either one or at max two of the following properties: accuracy, comprehensiveness and applicability. When compared to the state-of-the-art in activity recognition from mobile phones, our approach outperforms the state-of-the-art on the fronts of activities types and quantity, environments and settings covered, comprehensiveness, and applicability. We were also able to achieve comparable results in terms of accuracy, while having a significantly higher number of activities.

ILLUSTRATIONS

TABLES

CHAPTER I

INTRODUCTION

Recommender systems are modules that use information about the user and his habits to help suggest items that might be of interest to the user. To suggest good and appropriate recommendations, a recommender system analyzes information that is either contextual, based on the user's context such as time, mood and emotion, or non-contextual such as interests, likes, and dislikes. User's emotion is one of the factors that play an important role in these recommendations and is affected by multiple elements such as the user's current activity (Figure 1). Consequently, knowing the user's current activity is essential for making appropriate recommendations.

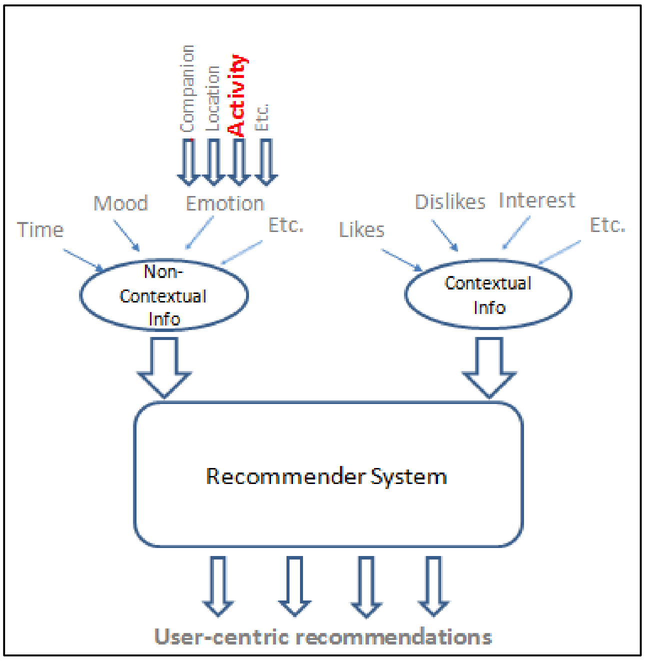

Figure 1 Recommender System Architecture

Consider, for instance, an individual in the following scenario: "It is a weekend evening, and the user is at home". A traditional recommender system would provide suggestions for a place to party depending on the user's interest (type of places the user likes) as well as the time. Suppose now that more information such as current emotion is available. A recommender system with such information can provide better suggestions such as dinner, if the user is in a relaxed mood or maybe a party with friends if the user is bored. Consequently, some insights on the user's contextual and non-contextual information become means for better assessment of the user's psychological and emotional condition leading to accurate and personalized recommendations by the recommender system.

According to [21], stress is the cause of almost a 100 million lost workdays. It is also related to nearly 50% to 75% of diseases which can affect an employee's performance, motivation towards goal achievement, and can result in low productivity. Other effects of stress can be physical, psychological and cardiovascular. One cause of stress for an employee on his way to work might involve traffic. By being able to detect the actual situation/activity (being stuck in traffic) , a recommender system can suggest some relaxing music that will eventually help distress the employee, and allow him to better start off his day.

Complex (semantic) activity recognition, along with emotion recognition constitute significant information that will aid in enhancing a recommender system's suggestions. Our ultimate aim is to incorporate activity recognition within recommender system to better personalize and enhance recommendations. We focus on activity recognition as it directly affects user's emotions which if known, it allows the recommender system to suggest actions that aim towards enhancing the user's current emotional state. Consequently, we focus on recognizing a set of sixteen activities consisting of : working regular, in a meeting, driving normally, stuck in traffic, in a vehicle, taking a break, eating, relaxing, watching TV, listening to music, playing games, exercising, biking, walking, running, spending time with family/friends.

In this thesis, we address the problem of semantic activity recognition using data collected from the user's mobile phone. Our approach recognizes a large set of activities that are comprehensive enough to cover most activities users engage in. Moreover, multiple environments are supported, for instance, home, work, and outdoors. Our approach suggests a multi-level classification model that is accurate in terms of classification accuracy, comprehensive in terms of the large number of activities it covers, and applicable in the sense that it can be used in real settings. Hence, in literature, these three properties are not existent altogether in a single approach. Proposed approaches normally optimize their models for either one or at max two of the following properties: accuracy, comprehensiveness, and applicability. When compared to the state-of-the-art in activity recognition from mobile phones, our approach outperforms the state-of-the-art on the fronts of activities types and quantity, environments and settings covered, comprehensiveness, and applicability. We were also able to achieve comparable results in terms of accuracy, while having a significantly higher number of activities.

CHAPTER II

RELATED WORK

This section describes the work done in literature to solve the "Activity Recognition" problem. The different approaches to the activity recognition problem will be presented alongside the limitations of each approach before stating our solution, its advantages and disadvantages.

The work presented in literature regarding the activity recognition problem is mainly divided into three categories: approaches recognizing pc/office related activities and states [2-5] and approaches recognizing everyday user's activity in different contexts that can be divided into two subcategories depending on the sensing mechanism used. One category of approaches uses wearable physical sensors [7, 10] for sensing while the other category uses mobile phone sensors [8-14] with some approaches supporting continuous sensing on mobile phones [11, 13, 14]. Our approach uses mobile phone sensors to recognize day-to-day activities.

A. PC/Office Related Activity Recognition

In what follows, the different approaches addressing the detection of pc/offices related activities will be presented along with their main highlights. The authors in [1] proposed a personalized recommendation model that suggests web pages to the user according to the recognized user state. They extracted the contents of browsed web pages and used software to detect 3 user-states: work, study and entertainment. Their approach used a Naive Bayesian model to deduce the user state. They achieved an overall accuracy of 77.6 %. In [2], the authors described Coordinate, a forecasting service that provides predictions about the user's presence and availability. The system logs periods of presence and absence representing a user's time at and away from the computer and provides forecasts about the time until the user returns to his office. This is possible by matching a set of cases from an event database and then building one or more Bayesian networks that are used to compute the cumulative distribution over the time until the user will return. Then, they extended Coordinate to support predictions about attendance and interruptability of a person using a decision tree and a Bayesian network. They achieved an accuracy if 92% for attendance and 81% for interruptability. In [3], the authors proposed a hybrid learning system that assists TaskTracer, a system that helps users to easily organize and manage their resources, to predict the user tasks from desktop activities and email messages. TaskPredictor.WDS makes predictions using desktop activities and TaskPredictor.email makes predictions using email messages. A Naive Bayesian method was used to decide when to make predictions and a linear SVM was applied to make the predictions. TaskPredictor.WDS achieved a precision of 80% with coverage between 10 and 20% and TaskPredictor.email achieved a precision of 92% with coverage of 66%. A Naive Bayesian model was used in [4] to detect user's activity (PC, Desk and discussing) and availability (for a quick question, for a discussion, soon and not at all) in an office environment. User's PC and phone usage, PDA location and ambient sounds were used by the Bayesian model to determine the activity and predict user's availability. The accuracy achieved was 80.3% for activity detection and 85.25% for predicting availability. In [5], a user-dependent model for activity recognition was suggested to overcome the problem of using single user's data to train an activity recognition model. The authors clustered the users based on their behavior defined in terms of activity, time and access points observed. Users from the same cluster contribute together to train the classifier. The model was trained using the Expectation-Maximization algorithm EM and the classifier chooses the activity having the highest likelihood. The accuracy achieved was 11% over the merged baseline where all users' training data are pooled together and an activity recognition model is trained for all users, and 13% over the single user's data baseline where the model uses each single user's data to train a personalized activity recognition model. Although the pre-mentioned approaches discuss activity recognition, however, they address a slightly different problem in terms of the data being used.

B. Everyday User's Activity Recognition in Different Contexts

Another category of approaches recognizes everyday user's activity in different contexts and can be divided into two main subcategories based on the sensing mechanism.

1. Activity Recognition Using Wearable Sensors

The different approaches to activity recognition using wearable sensors will be described in what follows. In [6], an acoustic wearable sensor around the neck was developed to detect a set of throat-related activities : coughing, sighing, laughing, whistling, whispering, speaking, drinking with and without a sip, eating cookie and eating bread, deep breath and seated. The features used were zero cross-rate, total spectrum power, sub-band powers, brightness, spectral roll-off, spectral flux and MFCCs. Two different protocols were used for training and testing: leave-one-participant-out cross validation and leave-one-sample-per-participant-out. SVM, Naive Bayes and 5-NN were used for classification. SVM outperformed the two other techniques. For laboratory evaluation, the leave-one-sample-per-participant-out improved the accuracy by 30% over the leave-one-participant-out protocol. The F-measure accuracy achieved was 79.5% for the lab evaluation. A small real life experiment was conducted using only eating, drinking, speaking and laughing and an F-measure accuracy of 71.5% was achieved. In [9], the authors proposed an activity recognition system that uses mobile phone sensors as well as wearable physical sensors residing in multiple tinyOS motes. The set of sensors consisted of accelerometers, microphones, GPS, WiFi, light sensors and temperature sensors. Three main classification categories were targeted: Environment (Indoors, outdoors), posture (Cycling, Lying down, Sitting, Standing, and Walking) and activity (cleaning, Cycling, Driving, Eating, Meeting, Reading, Walking, Watching TV, Working). An improved version of the Adaboost.M2 [16] classifier was used to predict activities. Adaboost trains a set weak classifiers for each sensor and combines them in a robust classifier. It also presents a sensor selection module that selects the most accurate and powerful sensors. The authors suggested an additional sensor selection module based on the Pearson Correlation Coefficient between sensors. This additional module generates a set of sensors that have uncorrelated classification decisions. A retraining detection module was proposed to detect when retraining is needed (i.e. runtime data distribution is different from the training data collected). The retraining detection model is based on the KL divergence between the runtime data distribution and the training dataset. Without the additional sensor selection and retraining detection modules, an average accuracy of 85.3% was achieved. The sensor selection module showed an improvement of 10% over the previous results and the retraining detection module showed a significant reduction in the number of training instances compared to a periodic retraining approach. In [16], a set of seven activities (standing, walking, and running, climbing up stairs, climbing down stairs, sit-ups, vacuuming and brushing teeth) was recognized using a wearable tri-axial accelerometer. Mean, standard deviation, energy and correlation are used as features. Base-level and meta-level classifiers are tested in four different training settings. Plurality voting achieved the highest accuracy 99.57%. In [17], a set of 20 activities is recognized using 5 wearable biaxial accelerometers. Mean, energy, frequency domain entropy and correlation features are fed into multiple classifiers and two training techniques (user specific and leave-one-participant-out) are used. The best accuracy (84.26%) was achieved with the decision tree using the leave-one-participant-out training. In [18], 7 activities (lie, row, exBike, sit/stand, run, nordic walk and walk) is recognized using a large set of sensors including acceleration, audio, temperature and light sensors and many others. The features were selected based on distribution bar graphs. Three models were used for testing using a 12-fold leave-one-participant-out cross validation. The automatically generated decision tree outperformed the custom decision tree and ANN with an accuracy of 86%. Although these approaches achieved high accuracy in recognizing activities, they remain unpractical in everyday people's life since they require the use of wearable physical sensors.

2. Activity Recognition Using Mobile Phone Sensors

Mobile phones with their increasing computational and sensing capabilities, storage and variety of applications have become an essential need in everyday user's life and events. The use of mobile phones to detect user's activity contributes in the development of healthcare monitoring systems, life logging applications and recommender systems. The recent approaches to the activity recognition problem focused on the use of mobile phone sensors to detect the user's daily activity. In [7], an energy efficient rule based approach was used to recognize a set of 8 user states (working, meeting, office_loud, resting, home_talking, home_entertaining, place_quiet, place speech, place_loud, walking and vehicle) using mobile-phone sensors that include GPS, WiFi, microphone and accelerometer. GPS was mainly used to detect the user's mode of travel by calculating the user's velocity, and to identify if the user entered a closed place. WiFi was used to identify the user's current location using a set of prerecorded access points. Accelerometer was used to detect other sensors whenever motion is detected and was also used as a motion classification tool, using the standard deviation values, only when the GPS was unavailable. Microphone was used for background sound classification (silence, speech, and noise/music). Each user state was defined by a combination of sensor values and an XML descriptor consisting of a set of state names, sensors to be monitored and conditions for state transitions was used to detect transitions from one state to another. The accuracy achieved by the suggested system reached a value of 92%. Although this rule-based approach presents a remarkable contribution in terms of energy efficiency, it is not scalable to a larger set of activities since the same combination of states can define multiple activities as the number of activities increases and the model needs to be reformulated upon the addition of new activities. Also this approach does not recognize actual activities, it recognizes states instead. In [8], the authors proposed a crowdsourcing framework that combines scene, event and phone context to recognize audio scenes (car, hall, indoor, restaurant, street) and events (keyboard, music, radio, speech, tv, walk, none) and phone context (in-pocket, out-pocket). The framework gathers everyday sounds from people and shares the audio models through a central cloud server. MFCC features were extracted from each audio clip and used to build a GMM. Each audio clip was represented by a histogram using the Gaussian Mixture Model (GMM). K nearest neighbor algorithm was used as a classification tool to recognize a new audio clip and label it by a scene, an event and a phone context (in-pocket/out-pocket). The local device programmed with common sounds tries to recognize a new audio clip and in case it failed to recognize it, it asks the cloud server that tries to recognize using the models collected from many users. Kullback-Leibler divergence and Euclidean distance were used as two distance measure and the experiments showed that KL outperformed the Euclidean Distance. The accuracy achieved for the three categories was between 77.6 % and 88.9%. Although this approach achieved good accuracy, it recognizes scenes/states rather than actual semantic activities. In [10], the authors proposed a continuous sensing engine for mobile phones that addresses the challenges of long term sensing and inference, and processing data from multiple sensors. The proposed engine can be used by any phone application that requires activity recognition as input. The authors suggested a set of three pipelines: accelerometer, microphone and GPS to recognize user's activities and location. Each of the accelerometer and microphone was used separately to recognize a different set of activities and the GPS was used to detect the user's location. The accelerometer pipeline detects walking, cycling, running, vehicle and stationary activities and addresses the challenges of the phone body position errors and temporary states errors by using orientation-independent features and recognizing the transition states and the periods of interaction with the phone. The microphone pipeline detects brushing teeth, showering, typing, vacuuming, washing hands, crowd noise and street noise states and addresses the challenges of the resource efficiency by regulating the amount of data that enters the pipeline by filtering audio frames that contains speech and minimizing the redundant classification operations. The GPS pipeline mainly detects the location of the user and reduces the amount of energy consumed due to the GPS sampling by introducing an optimal sampling schedule adaptive to the mobility mode of the user and the battery budget. After comparing multiple classifiers the decision tree was chosen for the accelerometer pipeline which achieved an accuracy of 94.52%. A GMM classifier was used for the audio-based classification and achieved an accuracy ranging from 0.6 to 0.98% depending on the activity and the decision tree classifier for recognizing voice frames achieved a recall of 85.35%. The GPS pipeline achieved an average error of 48.1m and power consumption of 0.112W for the weekend traces and an average error of 41.7m and power consumption of 0.076W for the weekday traces with the suggested MDP learned duty cycle. This resource efficient and body position independent approach does not do any classification when voice is detected in the audio samples. A hierarchical activity classification model for inferring semantic activities from accelerometer data only was proposed in [11]. The model was supported by the GPS sensor for location tagging. The semantic activities where referred to as Macro Activities and described as a sequence of smaller activities called Micro Activities. A lower layer detects micro-activities from the collected accelerometer data and an upper layer infers the semantic activities from the sequence of micro-activities. Statistical 2D and 3D features were extracted from the accelerometer data to detect micro activities and two feature extraction techniques, at the micro-activity level, were suggested and investigated: a duration preserving feature extraction finds the union of all such qualifying sub-sequences across all semantic activities in the training data and a transition preserving feature extraction that preserves the transition between distinct and adjacent micro-activities. A correlation-based selection of features was applied after the feature extraction step. The list of micro-activities detected consists of: sit, sit-active, walk, loiter, bursty move, stand and using stairs. The semantic (macro) activities consist of office activities {O_work, O_break, O_coffee, O_toilet, O_meet, O_lunch} and home activities {H_work, H_relax, H_break, H_cook, H_eat, H_baby}. The lower layer was tested by a 10 fold cross validation approach using a set of classifiers (Decision tree, Naive Bayes, Bayesian Network, LibSVM and Adaboost) and an accuracy greater than 88% was achieved for all users. The upper layer was tested by the same classifiers using an 8 fold cross validation technique. An average accuracy of 77.14% was achieved with individual accuracies per user ranging from 48.89% to 97.14%. Compared to a one-level classification approach, an improvement ranging from 7% to 20% was observed. While this approach detects semantic, non-atomic activities, it is limited to only two locations: office and home and is not scalable in terms of activities covered since only movement based activities are being recognized using accelerometer data and sound based activities are ignored. A hierarchical activity classification model was also introduced in [22] to detect a set of three activities: shopping, taking bus and moving (by walk). The model consists of a 2-Level HMM classifier where the first level detects a set of four actions (stand, walk, run, stair up/down) and the second level detects the actual activity where each activity consists of a sequence of actions. To recognize low-level actions, 3 different HMMs were trained for x, y and z accelerations respectively. For activity classification, an HMM was trained to detect the actual activity using the sequence of actions detected in level 1. The hierarchical HMM model was compared to 1 level HMM model and ANN Model and outperformed both in terms of average precision for the three activities. However, this method recognizes a very limited set of three activities. In [12], the authors developed SoundSense, a framework for modeling sound events on resource limited mobile phones. The sensing system consists of supervised and unsupervised learning techniques to classify general sound types as well as discover individual specific sound events using a multi-level classification approach. A coarse-level classification module used generic models to classify music, voice and ambient sounds and an intra-level classification module used unsupervised learning techniques are used to categorize the ambient sound. Processed audio data is never stored on the phone to safeguard the privacy of the users and a frame admission control manages energy, cpu and memory usage by using spectral entropy and energy measurements to filter silent and hard to classify frames. The coarse category classification consists of a decision tree and collection of markov models to recognize voice, music and ambient sounds. An unsupervised ambient sound learning was proposed as an intra-level classification. MFCC features along with a multivariate Gaussian classifier and an HMM model are used to classify different ambient sound types. A ranking strategy is used to keep track of the interesting sound events. An accuracy of 90% was achieved for recognizing ambient sound and an 80%. Music and speech were recognized with an 80% accuracy. Four sound events (walking, driving cars, riding elevators and riding a bus) were recognized with an accuracy ranging from 25% to 100% per activity. Although this approach is important for its unsupervised learning of user-specific sound events, it is limited in the number of activities detected. In [13], a people-centric sensing application, CenceMe, was developed to sense information about where a user is and what is he doing. A split-level classification between a phone software and a backend software and power aware duty cycles that overcome the limitations of mobiles phones in supporting continuous sensing and the limitations of the programmability of mobile phones were adopted. The phone software consists of accelerometer, audio and event sensors along with a Bluetooth daemon, random photos, GPS, accelerometer and audio clients, and a sensor controller. The phone software outputs 2 primitives: voice/no voice and activity (sitting, standing, walking, running). Voice classification is done based on a discriminant analysis learning algorithm applied to the mean and standard deviation of a 460 bin Discrete Fourier Transform, and a 22% misclassification rate is obtained. A J48 decision tree was used for activity classification using the mean, standard deviation, number of peaks and the tree axis values. A 78.89% accuracy was achieved. The backend classifier consists of: conversation classification, social context classification, mobility detection and a location classification. The classifier takes as input the activity and voice primitives. The conversation classifier uses a rolling window of 5 audio primitives and outputs a conversation/no conversation state using a rule-based approach. The social context classifier outputs the social context (restaurant, meeting, alone, partying dancing, etc.) based on the neighborhood condition (CenceMe buddies detected by the Bluetooth daemon), the conversation/no conversation state, the activity classification and an audio volume threshold. The mobility detector uses a JRIP learning algorithm using multiple distance/time measurements to detect whether the user is in a vehicle. Finally, the location classification is based on bindings between a physical location, a textual description and a generic class (restaurant, library, etc.). An accuracy of 73% and 82.4% was achieved for the conversation and mobility classifications respectively. This approach considers an acceptable number of semantic activities with a good detection accuracy, however, its main focus is sensing on mobile phones rather than actual activity detection.

Although these approaches achieved good accuracy in detecting activities, no approach recognizes a large set of semantic activities while combining accuracy, applicability and comprehensiveness.

CHAPTER III

OUR APPROACH

In this section, we will present our approach to semantic activity recognition using mobile sensors describing the different stages of this process from data collection, to data cleaning and processing, feature extraction and experimental setup.

A. Data Collection

To describe our data collection approach for collecting sensor data and user annotation in real life settings, we start by describing the data collection mobile application and the sensors used for collecting data. We then elaborate on the set of activities we will be recognizing highlighting the differences between similar activities such as working regular and in a meeting, driving normally and in a vehicle. Finally we present the data storage methods and data cleaning process applied for generating clean and useful data.

To collect labeled data for training and testing, we designed an Android data collection application and implemented it on a Samsung GT-I9001 Android phone. The application collects accelerometer, audio, GPS and WiFi readings as well as activity ground truth labels (discussed later). Additional details such as location, number of companions and activity duration are also collected and constitute additional information about the user's. The application prompts the user about his current activity once every 10 minutes. This time interval allows us to collect enough data without annoying the concerned user since we want to collect as much data as possible while keeping the user engaged in the data collection process. For every minute within the 10 minute interval:

- The accelerometer collects x, y and z acceleration values for 8 seconds and goes inactive for the coming 12 seconds. According to [7] a duty cycle of 6 seconds sampling and 10 seconds sleeping constitutes an acceptable trade-off between energy efficiency and robustness of state recognition. In our study we set the duty cycle to 8 seconds of sampling and 12 seconds of sleeping in order to obtain exactly 3 cycles of sampling for each one-minute interval of data collection.
- One WiFi scan is issued. For energy efficiency reasons, we limited the data collection to one WiFi scan every minute assuming that the location is subject to a minor change within a one minute interval in a location where WiFi is available.
- 10 seconds of audio is collected. For energy efficiency reasons we set the length of the audio recording to 10 sec. This parameter was tweaked later on, after running and analyzing a set of experiments as will be shown at later stages.
- GPS data is continuously collected.

Our aim is to obtain an activity annotation and sensor readings for every 1 minute interval. For each interval we need a WiFi scan to determine the set of available access points, GPS readings to determine location and/or infer the mode of travel of the user. Accelerometer readings and audio samples are necessary to capture the movement and the sound based activities respectively. The above sensor readings are then mapped to what the user inputs as his current activity, location, number of companions, and activity duration

1. Data Collection app

The developed application is a background data collection process that uses the sensors of the mobile phone to collect data that will be used in our activity recognition system. The sensors of the mobile phone used to collect the data are:

- Accelerometer - Collects information about X, Y, Z acceleration values
- GPS - Collects Data about longitude and latitude
- WiFi - Set of available Access Points alongside their Signal Strength and the Network ID
- Audio Recordings - Audio samples recorded at a certain period

a. <u>Functionality:</u>

The application starts by prompting the user (notification) to input data answering the below questions as shown in figure 1:

Where are you? (Figure 2)

What are you doing?

With how many people?

For how long? (Figure 3)

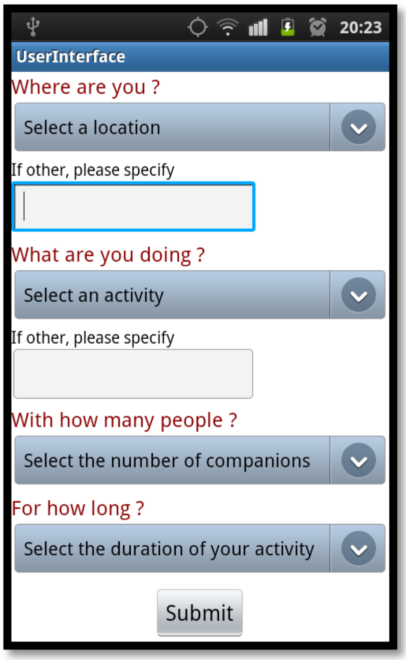

Figure 2 Activity Annotation Screen

The user has a one minute to open the notification before it disappears. This restriction along with the time allowed to annotate the activity (discussed later) will keep an interval of 8-12 minute between 2 activity logs.

Each of the above questions is associated with a drop-down list of suggestions from which the user can select an answer. Moreover, the first two questions also allow the user to input a typed-in answer not included in the drop-down list.

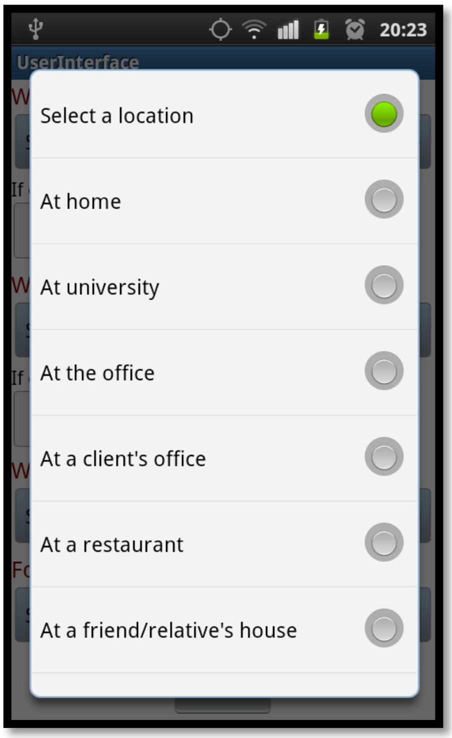

Figure 3 Activity Annotation

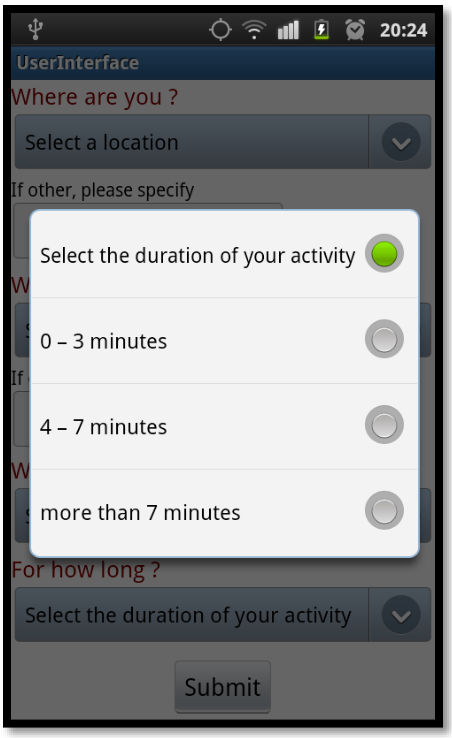

Figure 4 Activity Annotation Duration

The user has an interval of one minute to annotate his activity. If he does not submit his data one minute after he opens the notification, the annotation screen will disappear. The activity process is repeated every 10 minutes provided that the application is still running in the background. We have later changed the annotation frequency to 1 annotation every 5 minutess in order to collect more data.

The set of possible “location” annotations include: home, university, office, a client's office, restaurant, friend/relative's house, gym, street and other. The set of possible “number of companions” annotations include: 0, 1, 2, 3, 4, 5 and more than 5. The set of possible activity “duration” annotation include: 0 to 3 minutes, 4 to 7 minutes and more than 7 minutes.

The set of possible activity annotations include: working regular, in a meeting, driving normally, stuck in traffic, in a vehicle, taking a break, eating, relaxing, watching TV, listening to music, playing games, exercising, biking, walking, running and spending time with family/friends. Following are the elaborations on each activity separately highlighting the characteristics and the difficulties of detecting each activity. Here is a brief explanation of each:

Working regular: Corresponds to a state where the user is in active regular daily work activities, excluding meetings (if applicable).

In a meeting: Corresponds to a state where the user is active in a meeting with one or more people

Driving normally: Corresponds to a state where the user is driving with continuous movement.

Stuck in traffic: Corresponds to a state where the user is driving a car, or present in a car with non-continuous and minimal movement.

In a vehicle: Corresponds to a state where the user is in a car with continuous movement, but not driving.

Taking a break: Corresponds to a state where the user is not performing any of his work tasks at the work place.

Eating: Corresponds to a state where the user is having breakfast, lunch, or dinner.

Relaxing: Corresponds to a state where the user is not performing any activity provided he is not located at his workplace.

Watching TV: Corresponds to a state where the user is watching TV.

Listening to music: Corresponds to a state where the user's primary activity is listening to music.

Playing games: Corresponds to a state where the user is playing games.

Exercising: Corresponds to a state where the user is exercising, other than walking, running, or biking.

Biking: Corresponds to a state where the user is biking.

Walking: Corresponds to a state where the user is walking (Outdoor location).

Running: Corresponds to a state where the user is running (Outdoor Location)

Spending time with family/friends: Corresponds to a state where the user is with more than one person, chatting, and interacting with other people.

b. Data Storage

Every time the application starts, five csv files are created: four of them corresponding to the sensors data and one for activity. Each file's name is simply the application start date-time along with the sensor name or 'activity'.

After the user submits his input, data is saved in the structured csv file (will be discussed later) on the mobile phone storage. All of the sensor data and ground truth labels along with location, companion and duration are saved in the corresponding structured csv file. All the csv files have the date-time as the data of the first column in each row. Each audio file's name consists of "audio_" concatenated with the date-time at which the audio recording started. The audio file name is saved in the corresponding csv file.

2. Data Cleaning

After collecting the data and before the feature extracting process, cleaning the data is a crucial process. Extracting only useful data as well as structuring the data in a simple format facilitates the feature extraction process. Data was extracted from the mobile phone to an HP Pavilion DV6 and two java scripts were implemented in order to clean the data as follows:

- assembling the logs from different sensors together with the activity annotations in one file to create a time-based sequence of sensor readings and activity annotations
- deleting sensor logs with no activity annotation, sensor logs falling outside the activity duration interval and sensor logs corresponding to the user's interaction period with the phone to annotate his current activity

a. <u>Sensor Readings and Ground Truth Labels Grouping</u>

As mentioned earlier, different files are used to store the sensor readings and activity ground truth labels. This is done, to make it possible for every sensor to immediately write its logs on its own csv file without the need to wait other sensor writing on the same file to finish. The sensor readings are being annotated immediately after they are sampled. We grouped all the data from the different files in a single file to be further processed by other data cleaning scripts to obtain a single continuous time-based sequence of the sensor readings and activity annotations. Below are examples (figures 3-6) of the different sensor readings files and one file combining the readings of the different sensors with the activity annotations after running the sensor readings and ground truth labels grouping script.

5599	6/2/2013 21:10:01						-0.15323	-0.61291	9.500074
5600	6/2/2013 21:10:01						-0.15323	-0.61291	9.653301
5601	6/2/2013 21:10:02						-0.15323	-0.61291	9.500074
5602	6/2/2013 21:10:02						-0.15323	-0.61291	9.653301
5603	6/2/2013 21:10:03						-0.15323	-0.61291	9.500074
5604	6/2/2013 21:10:15						-0.15323	-0.61291	9.653301
5605	6/2/2013 21:10:36						-0.15323	-0.76614	9.653301
5606	6/2/2013 21:10:36						-0.15323	-0.61291	9.653301

Figure 5 Accelerometer Readings File

58	6/2/2013 21:09:12	/audio_2013-06-02_21-08-41-188.m4a
59	6/2/2013 21:10:12	/audio_2013-06-02_21-09-41-776.m4a
60	6/2/2013 21:11:13	/audio_2013-06-02_21-10-42-349.m4a

Figure 6 Audio Readings File

	A	B	C	D	E	F
1	Time	Activity	Location	Companic	Duration	User Interaction Time
2	6/2/2013 20:32:30	other_shopping	other_mall	0	4 â€“ 7 minutes	user intercation period at: 2013-06-02 20:32:30.423: 17
3	6/2/2013 21:12:33	Eating	At a restaurant	1	0 â€“ 3 minutes	user intercation period at: 2013-06-02 21:12:32.725: 13

Figure 7 Activity Annotations File

	A	B	C	D	E	F	G	H	I
12342	6/2/2013 21:10:01						-0.15323	-0.61291	9.653301
12343	6/2/2013 21:10:02						-0.15323	-0.61291	9.500074
12344	6/2/2013 21:10:02						-0.15323	-0.61291	9.653301
12345	6/2/2013 21:10:03						-0.15323	-0.61291	9.500074
12346	6/2/2013 21:10:12	/audio_2013-06-02_21-09-41-776.m4a							
12347	6/2/2013 21:10:15						-0.15323	-0.61291	9.653301
12348	6/2/2013 21:12:33	Eating	At a restaurant	1	0 - 3 minutes	user intercation period at: 2013-06-02 21:12:32.725: 13			
12349	6/2/2013 21:10:36						-0.15323	-0.76614	9.653301
12350	6/2/2013 21:10:36						-0.15323	-0.61291	9.653301

Figure 8 File combining the readings of different sensors with the activity annotations in a time-based sequence

b. Non-Useful Data Deletion

After aggregating all the data in one file, non-useful data are deleted:

- Sensor data with no activity annotations are removed as they do not contribute to the results in any way since they do not provide any information about the user's activity.
- It is possible for the user to do more than one activity in a 10 minutes interval that's why the duration of the activity input by the user helps maintaining only data corresponding to the actual annotation: Data sampled during the 10 minutes interval of an activity annotation but that do not correspond to the indicated duration of the activity are also removed.
- Sensor data corresponding to the user's interaction period with the phone while annotating his activity are also removed because the current activity might have been interrupted by the activity annotation process.

Below is an example (figures 7-8) of a log file before and after applying the non-useful data deletion script. This example shows that readings collected between 17:28:04 on the 31st of May and 17:44:46 of the same day were deleted after being identified as non-useful data by the script.

	A	B	C	D	E	F	G	H	I
2556	5/31/2013 17:27:51						-0.15323	-0.61291	9.653301
2557	5/31/2013 17:27:55	Working regular	At the office	3	more than 7 minutes	user intercation period at: 2013-05-31 17:27:55.131: 6			
2558	5/31/2013 17:28:04						-0.15323	-0.61291	9.500074
2559	5/31/2013 17:28:04						-0.15323	-0.61291	9.653301
2560	5/31/2013 17:28:05						-0.15323	-0.61291	9.500074
2561	5/31/2013 17:28:05						-0.15323	-0.61291	9.653301

Figure 9 Training data file before deleting non-useful data

	A	B	C	D	E	F	G	H	I
1842	5/31/2013 17:27:49						-0.30645	-0.61291	9.653301
1843	5/31/2013 17:27:55	Working regular	At the office	3	more than 7 minutes	user intercation period at: 2013-05-31 17:27:55.131: 6			
1844	5/31/2013 17:44:46						1.838724	0.153227	4.903264
1845	5/31/2013 17:44:46						3.06454	-3.06454	12.56461
1846	5/31/2013 17:44:46						0.766135	-1.6855	10.57266
1847	5/31/2013 17:44:46						2.758086	0	6.435534

Figure 10 Training data file after deleting non-useful data

B. Data Processing

In this section we describe two data processing techniques on the data set obtained after the cleaning phase. These techniques will result in a data set that is clear, well-structured, organized and ready for the feature extraction phase.

Further processing of the data is needed before the feature extraction step to allow an easier manipulation of the huge amount of data obtained. Java scripts were used for the following purposes:

- mapping the activity annotation individually with every minute of the corresponding 10 minutes interval (1)
- replacing every one-minute interval logs with a single record (2)

1. Generation of One-minute Interval Annotations

During the data collection phase, the user is triggered every 10 minutes about his current activity although sensors are being sampled on a one minute basis. After removing non-useful data as described in the previous section, a mapping between the activity annotation and each of the 10 (or less) corresponding one-minute logs is executed resulting in 10 (or less) one-minute logs for each activity annotation. The resulting file is a sequence of one-minute sensor logs followed by their corresponding activity annotation. This sequence is repeated for every minute of sensor data collected.

2. One-minute Format Records Generation

For more simplicity and easy manipulation of the data, all the logs corresponding to a one minute interval are replaced by a single record. The single record consists of a timestamp, activity label, location, number of companions, duration of the activity, set of WiFi APs with their signals strength and the name three data files. Each of these files hold the logs of a different sensor (accelerometer, audio and GPS).

C. Feature Extraction

In this section we describe the feature extraction process we applied to the final data set in order to extract audio and accelerometer features, as well as build the feature matrix that will be fed into our activity recognition model. Two main phases exist:

- Audio and accelerometer features extraction
- Building the feature matrix

The result of the feature extraction process is a huge matrix, the single elements of which are the feature vectors corresponding to each minute of data collection.

We are considering a feature vector consisting of audio and accelerometer features. For this purpose, we performed two feature extraction procedures and then combined the resulting feature sets in a single feature vector.

1. Audio Features Extraction

Audio features used in our approach consist of MFCC features, Spectral Rolloff, Spectral Flux and ZCR. to extract these features, we used a Unix based tool called Yaafe that takes the audio files and the list of features to be extracted as input, and outputs a "csv" format file for each feature of an audio file. Each file contains the values of the different components of a single feature. These audio features together constitute the audio feature matrix.

2. Accelerometer Features Extraction

Accelerometer features used in our approach consist of : Means, Variances, Mean-Magnitude, Magnitude-Mean, Single Magnitude Area (SMA) and Standard Deviation of Magnitude as detailed in the below table.

Table 1 Accelerometer features used in the first experiment

Mean	AVG($\sum x_i$)
Variance	VAR ($\sum x_i$)
Mean-Magnitude	AVG($\sqrt{x_i^2 + y_i^2 + z_i^2}$)
Magnitude-Mean	$\sqrt{\bar{x}^2 + \bar{y}^2 + \bar{z}^2}$
Single Magnitude Area	$\frac{1}{n}\sum_{i=1}^{n}(\lvert x_i\rvert + \lvert y_i\rvert + \lvert z_i\rvert)$
Standard Deviation of Magnitude	STDEV($\sqrt{x_i^2 + y_i^2 + z_i^2}$)

To extract these features, a script scans the list of accelerometer log files and executes the necessary calculation for each feature.

3. Building the Features Matrix

After extracting both audio and accelerometer features and building the corresponding matrices, we combine the two feature matrices into a single matrix to be fed into the classification tool, the rows of a matrix being the combined feature vectors. The matrix is built by reading from the both matrices and mapping each audio feature vector to the corresponding accelerometer feature vector.

D. Classification Model and Experiments

Two users were engaged in our research, both carrying the application on a Samsung GT-I9001 Android phone over a period of six weeks. The users carried the phones during their daily regular activities from home to the office then back home and during any other activity and in any location. After the process of gathering, cleaning, and processing the data, the below highlights the statistics pertaining to the remaining data:

Activity	# of Instances
working regular	71
in a meeting	92
driving normally	20
stuck in traffic	83
in a vehicle	23
taking a break	17
eating	93
relaxing	13
watching tv	69
listening to music	76
playing games	0
exercising	14
biking	29
walking	100
running	18
spending time with family or friends	54
All Activities	**772**

Figure 11 User 1 - Data Statistics

Activity	# of Instances
working regular	39
in a meeting	38
driving normally	28
stuck in traffic	5
in a vehicle	17
taking a break	8
eating	126
relaxing	28
watching tv	68
listening to music	9
playing games	15
exercising	14
biking	34
walking	244
running	0
spending time with family or friends	43
All Activities	**716**

Figure 12 User 2 - Data Statistics

It is worth noting that, both users had recorded instances for all activities, except for user 1 who recorded zero activities labeled "Playing Games" and user 2 who recorded zero activities labeled "Running".

We then performed feature extraction on the above data set, for audio and accelerometer feature and built the feature matrix as mentioned in the previous sections.

Our approach; a 2 - Level Classifier - is an improved version of a 1 - Level Classifier , whereby it's goal is to reduce the number of misclassifications between semantically similar activities, the first step being reducing the number of misclassifications between non-similar activities. In what follows, we will go in depth through the process of the 1-Level Classifier, followed by the observations that resulted from it. We then move on to discuss the motivation of our 2nd level classifier in details, along with the improvements it had introduced on the previous results.

1. 1-Level Classification Model

The 1-Level Classification model used in our approach consists of a machine learning model - whose core is the SVM classifier which can be found in WEKA as *classifiers.functions.SMO*. SVM was used following a set of preliminary experiments using Naïve Bayes, Decision Tree and SVM. In all cases, SVM has outperformed the previously mentioned classifier - which constitutes the reason behind using it in our classification model.

The classification model takes the feature matrix that resulted from the audio and accelerometer features as input, and outputs a label relative to one out of the sixteen activities we are detecting in our work.

The model was trained using a 5-Folds Cross Validation technique; a technique which segments the data set into 5 equal sets, and at each of the 5 iterations, takes 1 set as a test-set and the remaining 4 as training data.

We have applied the classification model for both users under study, and the results are illustrated as follows:

Activity	F-Measure
working regular	0.717
in a meeting	0.654
driving normally	0.000
stuck in traffic	0.926
in a vehicle	0.936
taking a break	0.522
eating	0.747
relaxing	0.870
watching tv	0.831
listening to music	0.849
playing games	-
exercising	0.867
biking	0.881
walking	0.794
running	0.593
spending time with family or friends	0.578
All Activities	**0.718**

Figure 14 F-Measure - User 1

Activity	F-Measure
working regular	0.623
in a meeting	0.667
driving normally	0.918
stuck in traffic	0.000
in a vehicle	0.417
taking a break	0.000
eating	0.696
relaxing	0.814
watching tv	0.970
listening to music	0.941
playing games	0.815
exercising	0.692
biking	0.795
walking	0.966
running	-
spending time with family or friends	0.512
All Activities	**0.655**

Figure 13 F-Measure - User 2

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	fp	fn
a: working regular	**57**	12	0	0	0	0	1	0	0	0	0	0	0	0	0	1	31	14
b: in a meeting	20	**66**	0	0	0	0	3	0	1	0	0	0	0	1	0	1	44	26
c: driving normally	0	0	**13**	5	1	0	0	0	0	0	0	0	1	0	0	0	4	7
d: stuck in traffic	0	0	1	**81**	0	0	1	0	0	0	0	0	0	0	0	0	11	2
e: in a vehicle	0	0	1	0	**22**	0	0	0	0	0	0	0	0	0	0	0	2	1
f: taking a break	2	4	2	0	0	**6**	0	0	0	0	0	2	0	0	0	1	0	11
g:eating	0	3	0	0	0	0	**68**	0	1	0	0	0	0	18	0	3	21	25
h: relaxing	0	3	0	0	0	0	0	**10**	0	0	0	0	0	0	0	0	0	3
i: watching tv	0	3	0	0	0	0	1	0	**59**	4	0	0	0	1	0	1	14	10
j: listening to music	3	3	0	0	0	0	1	0	9	**59**	0	0	0	0	0	1	4	17
k: playing games	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0
l: exercising	0	0	0	0	0	0	0	0	0	0	0	**13**	1	0	0	0	3	1
m: biking	0	0	0	1	1	0	0	0	0	0	0	0	**26**	0	1	0	4	3
n: walking	2	5	0	0	0	0	8	0	0	0	0	0	0	**83**	0	2	26	17
o: unning	0	0	0	5	0	0	1	0	0	0	0	1	2	1	**8**	0	1	10
p: spending time with family or friends	4	11	0	0	0	0	5	0	3	0	0	0	0	5	0	**26**	10	28

Figure 15 Confusion Matrix - User 1

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	fp	fn
a: working regular	**19**	1	0	0	0	0	15	0	1	0	3	0	0	0	0	0	31	20
b: in a meeting	4	**17**	0	0	0	0	13	0	0	0	0	0	0	0	0	4	20	21
c: driving normally	0	0	**26**	0	0	0	2	0	0	0	0	0	0	0	0	0	1	2
d: stuck in traffic	0	0	0	**4**	0	0	0	0	1	0	0	0	0	0	0	0	0	1
e: in a vehicle	2	0	0	0	**4**	0	3	1	3	0	0	0	4	0	0	0	5	13
f: taking a break	4	0	0	0	0	**0**	3	0	0	0	0	0	0	0	0	1	1	8
g:eating	12	17	0	0	4	1	**73**	1	3	0	1	0	0	2	0	12	52	53
h: relaxing	0	0	0	0	0	0	0	**21**	2	0	4	1	0	0	0	0	6	7
i: watching tv	3	1	0	0	0	0	3	2	**54**	0	0	0	0	5	0	0	12	14
j: listening to music	3	0	0	0	0	0	0	1	0	**4**	1	0	0	0	0	0	0	5
k: playing games	1	0	0	0	0	0	4	1	0	0	**9**	0	0	0	0	0	10	6
l: exercising	0	0	0	0	0	0	0	0	0	0	0	**14**	0	0	0	0	1	0
m: biking	0	0	0	0	1	0	1	0	0	0	0	0	**32**	0	0	0	4	2
n: walking	0	0	1	0	0	0	0	0	0	0	0	0	0	**243**	0	0	14	1
o: unning	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
p: spending time with family or friends	2	1	0	0	0	0	8	0	2	0	1	0	0	7	0	**22**	17	21

Figure 16 Confusion Matrix - User 2

The above results show that for both users, the minimum F-measure was zero. However, the maximum F-Measure for User 1 was 0.932 for the activity "in a vehicle", and for User 2 was 0.970 for the activity "watching TV". The resulting average F-Measure across both users was 0.68.

An in-depth look at the confusion matrix highlights that the activities generating the highest number of false positives/false negatives are:

- Working Regular
- In a Meeting
- Spending Time with Family or Friends

All the above activities, present all 3 aspects of a semantic activity: sound, motion, and social aspect - which makes these activities more complex and thus present more difficulty in accurately detecting each.

User 2 in specific, has exceptionally shown a high number of false negatives, showing the activity "eating" classified as "working regular" or "in a meeting", possibly due to the fact that "eating" in our case has been collected at both locations of "home" and "work", as well as the nature of the work of User 2, which might have resulted in both activities being similar.

The above confusion matrices show two types of misclassification: one between activities which are semantically similar (working regular-in a meeting for both users) - the other between activities which semantically different (eating-walking in the case of user 1, and eating-meeting in the case of user 2).

The process of reducing misclassifications can be divided into two main parts:

- Reducing the misclassifications between semantically similar activities
- Reduce the misclassifications between semantically non-similar activities

As a result, we have introduced a 2-Level Classifier, which will reduce misclassifications between semantically non-similar activities through grouping them into different groups of similar semantics:

- Work Activities

- Continuous Movement Activities
- Sound Based Activities
- Other Activities

This will enable us to, after reducing misclassifications between semantically non-similar activities, further enhance the detection accuracy of each of the activities by minimizing the misclassifications between activities which are non-semantically similar.

A detailed description of the 2-Level Classifier will follow in the coming section.

2. 2-Level Classification Model

Four groups of activities were introduced in this model as follows:

Group 1 - Work Activities:

- Working regular
- In a meeting
- Taking a Break

Group 2 - Continuous Movement Activities

- Driving normally
- Stuck in traffic
- In a vehicle
- Walking
- Running
- Biking

Group 3 - Sound Based Activities

- Watching TV

- Listening to Music
- Playing Games

Group 4 – Other Activities

- Eating
- Relaxing
- Gym Exercising
- Spending Time with Family\Friends

It is worth mentioning that Group 4 activities do not represent semantically similar activities, but rather these are activities that do not have semantic similarities with other activities and will thus be treated individually.

Although the activity grouping process can be formulated as an optimization problem, in this thesis we have grouped the activities based on their semantic similarities.

In what follows we will describe the two different levels of our 2-level classifier.

a. First Level Classification

The purpose of this classification is to assign each activity instance to 1 of the 3 activity groups or to a single activity from group 4. The first level classifier takes as input the feature matrix previously produced and labels each activity corresponding to one of Group1, Group2 or Group 3 with the corresponding group label.

i. Classification Model

The classification model consists of two main phases:

Phase 1: Feature selection

Phase 2: Classification

Given the large number of features we are dealing with, we used a built-in filter in WEKA called FilteredAttributeEval for the feature selection phase. This performs a ranking of the different features based on their importance by running an attribute evaluator on these features. After ranking the FilterdAttributeEval on the feature matrix, the matrix was filtered keeping only the first 5000 features as ranked by the filter. As a classifier, the SVM used in the 1 level classification model was also used in this approach. However, SVM was combined with a CfsSubsetEval feature selection algorithm that internally selects the top features and performs activity classification using the selected features.

ii. Training, Testing and Results

A 5 folds cross validation technique was also used in this experiment to perform training and testing. First the feature matrix was filtered based on the results on FilteredAttributeEval filter, after which the filtered matrix was fed into the combined CfsSubsetEval-SVM classifier resulting in the final classification model.

Training and testing were performed for both users and the First level classification results are illustrated below:

	a	b	c	d	e	f	g	fn	fp
a: work	**154**	4	5	1	9	0	7	26	75
b: continuous movement	5	**253**	9	1	2	0	3	20	13
c: eating	13	4	**72**	0	3	0	1	21	22
d: relaxing	3	0	0	**7**	0	0	3	6	3
e: sound	28	1	2	0	**110**	0	4	35	20
f: exercising	1	1	0	0	0	**12**	0	2	0
g: spending time with family or friends	25	3	6	1	6	0	**13**	41	18

Figure 17 Confusion Matrix - User 1 (Level 1)

	a	b	c	d	e	f	g	fn	fp
a: work	**45**	0	34	1	4	0	1	40	43
b: continuous movement	2	**315**	4	1	4	0	2	13	18
c: eating	30	6	**79**	0	4	0	7	47	52
d: relaxing	0	0	0	**22**	6	0	0	6	7
e: sound	8	5	6	5	**68**	0	0	24	21
f: exercising	0	0	0	0	0	**14**	0	0	0
g: spending time with family or friends	3	7	8	0	3	0	**22**	21	10

Figure 18 Confusion Matrix - User 2 (Level 1)

	fn		fp	
	one level classification	First Level classification	one level classification	First Level classification
work	51	26	75	75
movement	40	20	48	13
sound	27	35	18	20
other	57	70	34	43
	175	151	175	151

Figure 19 Inter-Group Misclassification Comparison - User 1

	fn		fp	
	one level classification	First Level classification	one level classification	First Level classification
work	54	40	57	43
movement	19	13	24	18
sound	25	24	22	21
other	81	74	76	69
	179	151	179	151

Figure 20 Inter-Group Misclassification Comparison - User 2

In summary, the total number of misclassifications across both users has been reduced significantly - ~ 14% reduction across User 1 and ~16% reduction across User 2 (as compared to the 1-Level Classifier) → which was the initial purpose of the 1st Level Classification of the 2-Level Classifier. The above results will help in improving the overall results of the model – as they can aid in isolating activities which are semantically similar with lower misclassification when compared to the 1-Level Classifier.

b. Second Level Classification

After performing the first level classification, a more granular classification is performed to assign each activity instance to one of the 16 activities we are detecting. Four different classifiers were built, one per group, each classifier recognizing activities within one group. Each classifier takes as input a feature matrix including the features of the instances that were classified as members of the corresponding group. Each of the four classifiers was built the same way as the first classifier. We start by performing a Feature ranking, after which the feature matrix is filtered using the top 5000 features and finally a combination of CfsSubsetEval and SVM is used to train the classifier. A 5 folds cross validation was also used for training and testing.

The below shows the final results of the 2 level classification model after performing both level 1 and level 2 classifications using the models built as described above.

Activity	F-Measure
working regular	0.577
in a meeting	0.466
driving normally	0.555
stuck in traffic	0.914
in a vehicle	0.843
taking a break	0.280
eating	0.761
relaxing	0.640
watching tv	0.583
listening to music	0.816
playing games	-
exercising	0.760
biking	0.718
walking	0.867
running	0.905
spending time with family or friends	0.387
All Activities	0.671

Figure 21 User 1 results

Activity	F-Measure
working regular	0.427
in a meeting	0.453
driving normally	0.946
stuck in traffic	0.889
in a vehicle	0.308
taking a break	0.000
eating	0.582
relaxing	0.764
watching tv	0.806
listening to music	0.615
playing games	0.529
exercising	0.966
biking	0.914
walking	0.970
running	-
spending time with family or friends	0.537
All Activities	0.647

Figure 22 User 2 results

i. Improvements on the 2-level Classification

In order to confirm whether group 4 activities, that do not represent a group of semantically similar activities, are taking advantage of the two level classification or a level classification can provide a high detection accuracy, we compared the results for these activities when using 1st Level Classification only versus using the 2-Level Classifier.

The below tables show the improvement in the results of both users:

User 1 F-Measure improved from 0.671 to 0.675

User 2 F-Measure improved 0.647 to 0.655

It is worth noting that, in all the above experiments, work activities in general, and working regular and in a meeting in specific, were some of the most semantically similar activities that showed a high number of misclassifications. As a result of this observation, we have merged these activities in an attempt to study the impact of this merging on the overall results.

The impact of this merging process is highlighted below:

User 1 F-Measure improved from 0.675 to 0.704

User 2 F-Measure improved 0.655 to 0.676

The resulting average F-Measure for the two users is now 0.69→ which shows a 0.01 (+1.5%) improvement when compared to the 1-Level Classifier.

CHAPTER IV

APPROACH EVALUATION

As mentioned earlier, in the literature, the three properties (accuracy, comprehensiveness and applicability) are not existent altogether in a single approach, and proposed approaches normally optimize their models for either one or at max two of these properties. In our work, we have developed a framework, which will be able to accurately detect a relatively large number of activities in an accurate, comprehensive, and applicable manner. The below table summarizes the added value of our approach when compared to state of the art methods:

Table 2 Accuracy, Scalability and Applicability Comparison

	Approach 1	Approach 2	Approach 3	Our Approach
Accuracy	+	+	+/-	+/-
Scalability (comprehensiveness)	-	+	-	+
Real life Applicability	-	-	+	+

As illustrated on Table 2 above, none of the “state-of-the-art” approaches has been able to combine all 3 aspects of accuracy, scalability (in terms of number of activities), and applicability in all real life.

Approach 1 shows high accuracy but misses on the aspects of scalability (same combination of states can define multiple activities as the number of activities increases), so the model used is in this approach needs to be reformulated whenever there is a need to introduce a new activity. This approach also misses on the aspect of applicability (as it only detects states instead of actual activities; e.g: Office Loud, Home Talking, Place Quiet, Place Loud).

Approach 2 on the other hand, enjoys both aspects of accuracy and scalability, but is not applicable in real life, since the classifier gets to a point where it checks whether there is any voice and flags this as voice. In real life, almost all activities have a voice component attached to them, especially those with a social companion. In their case, almost all real-life activities will be flagged as voice and not the actual activity taking place.

Approach 3, which enjoys a relatively good level of accuracy, and is relatively applicable in real-life, is in fact not scalable since it cannot cover any sound-based activities, as the accelerometer is the only sensor used.

Through our approach, it is important to note that all three aspects of Accuracy, Scalability, and Real-Life applicability are present, and show improvement when compared to the other approaches.

In the following section, we will discuss and compare in detail, our approach versus approach 3, as it is the only applicable approach among the others.

	User 1	User 2
working regular	0.717	0.646
driving normally	0.565	0.946
stuck in traffic	0.912	0.889
in a vehicle	0.837	0.308
taking a break	0.191	0.000
eating	0.763	0.582
relaxing	0.696	0.764
watching tv	0.589	0.806
listening to music	0.822	0.615
playing games	-	0.529
exercising	0.880	0.966
biking	0.720	0.914
walking	0.867	0.970
running	0.909	-
spending time with family or friends	0.391	0.537
	0.704	0.676

Figure 23 F-Measure Results - Our Approach

	User 1	User 2	User 3	User 4	User 5
O_work	0.913	0.888	0.931	0.950	0.812
O_break	0.882	0.433	0.000	0.853	0.767
O_meet	0.760	0.246	-	0.639	0.000
O_lunch	0.875	0.708	0.869	-	0.097
O_coffee	-	-	0.675	-	-
O_toilet	-	-	0.778	-	-
H_work	0.847	0	-	0.721	0.000
H_cook	0.655		0.875	1.000	0.673
H_relax	0.317	0.757	0.991	0.322	0.710
H_break	0.373		-	-	-
H_eat	0.555	0.508	0.940	0.715	0.000
H_baby	0.687	0	-	-	-
H_clean	-	-	0.896	-	-
	0.687	**0.442**	**0.773**	**0.743**	**0.382**

Figure 24 F-Measure Results - Approach 3 (Calculated F-Measure since it is not given)

As illustrated in the above figures, the average F-Measure in our approach is 0.69, as opposed to Approach 3, where the average F-Measure is 0.60 (+15%). Moreover, if we take the average F-Measure for the most accurate 2 users of Approach 3, it would be 0.75 (8% higher than our approach) – all said, taking into consideration that we are detecting more than double the number of activities than Approach 3 (15 Activities in our approach vs 7 Activities in approach 3). In summary, our approach compared to approach 3, performs better in terms of accuracy, and covers more than double the number of activities that approach 3 targets – with applicability being present within both approaches. Despite the fact that our approach performs better in terms of accuracy and scalability than approach 3, the below are the comparison obstacles we have faced, which have led us to perform yet another type of comparison that will follow. The obstacles can be summarized by: the 2 approaches use different sets of activities, we do not have access to the raw data of approach 3, and we need to perform changes at the data collection level if we were to re-implement their method. Apart from the fact that we achieved a better average f-measure – and thus better accuracy – we will present a full-fledged comparison on approaches. The comparison can be found in Figure 25 below:

	Approach 3	Our Approach	
Number of activities	8	15	✓
Indoor Activities	+	+	
Outdoor Activities		+	✓
Movement-based Activities	+	+	
Sound Based Activities		+	✓
Real-Life Applicability	+	+	
Comprehensiveness		+	✓

Figure 25 Full Fledged Comparison - Our Approach vs Approach 3

From the above Figure, we can see the our approach, (using the 2 level classifier), performs better than approach 3, as it detects almost double the amount of activities that approach 3 is capable of, it has the ability to detect outdoor activities (which is a missing feature in approach 3), can detect sound based activities, which are really at the core of any activity recognition problem, as well as scale in terms of number of activities and additions of new activity sets.

In summary, we have presented a 2-Level classifier, which categorized activities in semantically similar groups in order to help reduce the misclassifications that had occurred in the previous 1-Level Classifier, and has succeeded in doing so, thus helping us in achieving better accuracy results than Approach 3. The backbone of our approach has also positioned our approach as a scalable (comprehensive) and applicable approach, which when compared to any state of the art method of activity recognition, performs better in terms of combining all 3 aspects of accuracy, scalability, and applicability.

References

[1] Ji, Teng, etal. "A personalized recommendation model based on the user-state awareness." *Web Society, 2009. SWS'09. 1st IEEE Symposium on*. IEEE, 2009.

[2] Horvitz, Eric, et al. "Coordinate: Probabilistic forecasting of presence and availability." *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002.

[3] Shen, Jianqiang, etal. "A hybrid learning system for recognizing user tasks from desktop activities and email messages." *Proceedings of the 11th international conference on intelligent user interfaces*. ACM, 2006.

[4] Muhlenbrock, M., et al. "Learning to detect user activity and availability from a variety of sensor data." *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE, 2004.

[5] Zheng, Vincent W., and Qiang Yang. "User-dependent aspect model for collaborative activity recognition." *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*. AAAI Press, 2011.

[6] Yatani, Koji, and Khai N. Truong. "BodyScope: A Wearable Acoustic Sensor for Activity Recognition." (2012).

[7] Wang, Yi, et al. "A framework of energy efficient mobile sensing for automatic user state recognition." *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009.

[8] Hwang, Kyuwoong, and Soo-Young Lee. "Environmental audio scene and activity recognition through mobile-based crowdsourcing." *Consumer Electronics, IEEE Transactions on* 58.2 (2012): 700-705.

[9] Keally, Matthew, et al. "Pbn: towards practical activity recognition using smartphone-based body sensor networks." Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems. ACM, 2011.

[10] Lu, Hong, et al. "The Jigsaw continuous sensing engine for mobile phone applications." Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems. ACM, 2010.

[11] Yan, Zhixian, et al. "Semantic Activity Classification Using Locomotive Signatures from Mobile Phones." (2012).

[12] Lu, Hong, et al. "SoundSense: scalable sound sensing for people-centric applications on mobile phones." Proceedings of the 7th international conference on Mobile systems, applications, and services. 2009.

[13] Miluzzo, Emiliano, et al. "Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application." Proceedings of the 6th ACM conference on Embedded network sensor systems. ACM, 2008.

[14] Berchtold, Martin, et al. "Actiserv: Activity recognition service for mobile phones." Wearable Computers (ISWC) (2010): 1-9.

[15] Siirtola, Pekka, and Juha Röning. "Recognizing human activities user-independently on smartphones based on accelerometer data." IJIMAI 1.5 (2012): 38-45.

[16] Ravi, Nishkam, et al. "Activity recognition from accelerometer data."Proceedings of the national conference on artificial intelligence. Vol. 20. No. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[17] Bao, Ling, and Stephen Intille. "Activity recognition from user-annotated acceleration data." Pervasive Computing (2004): 1-17.

[18] Parkka, Juha, et al. "Activity classification using realistic data from wearable sensors." Information Technology in Biomedicine, IEEE Transactions on 10.1 (2006): 119-128.

[19] Huynh, Tâm, and Bernt Schiele. "Analyzing features for activity recognition."Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies. ACM, 2005.

[20] Kwapisz, Jennifer R., Gary M. Weiss, and Samuel A. Moore. "Activity recognition using cell phone accelerometers." ACM SIGKDD Explorations Newsletter 12.2 (2011): 74-82.

[21] Imtiaz, Subha, and Shakil Ahmad. "Impact Of Stress On Employee Productivity, Performance And Turnover; An Important Managerial Issue."International Review of Business Research Papers 5.4 (2009): 468-477.

[22] Young-Seol Lee and Sung-Bae Cho. "Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer." HAIS, pp. 460–467, 2011