APPROVAL SHEET

Title of Thesis: Context-Aware Middleware for Activity Recognition

Name of Candidate: Radhika Dharurkar
Master of Science, 2011

Thesis and Abstract Approved:

Dr. Tim Finin
Professor
Computer Science

Date Approved: ________________

Curriculum Vitae

Name: Radhika Dharurkar

Degree and date to be conferred: MS in Computer Science, 2011.

Secondary education: Pune University, Maharashtra, India, 2003.

Collegiate institutions attended:

University of Maryland at Baltimore County, M.S in Computer Science
(Aug 2009 – May 2011).

Cummins College of Engineering, Maharashtra, India,
(August 2003 - May 2007).

Major:Computer Science.

Professional positions held:

Software Developer Intern, Yahoo, Sunnyvale, CA, USA
(June 2011 – Aug 2011).

Software Developer Intern, Constellation Energy, Baltimore, MD, USA
(June 2010 – Aug 2010).

Technical Associate, Tech Mahindra, Pune, India
(July 2007 – July 2009).

ABSTRACT

Title of Document: CONTEXT-AWARE MIDDLEWARE FOR ACTIVITY RECOGNITION

Radhika D. Dharurkar, Master of Computer Science, 2011

Directed By: Professor Dr Tim Finin,
Department of Computer Science and
Electrical Engineering

Smart phones and other mobile devices have a simple notion of context largely restricted to temporal and spatial coordinates. Service providers and enterprise administrators can deploy systems incorporating activity and relations context to enhance the user experience, but this raises considerable collaboration, trust and privacy issues between different service providers. Our work is an initial step toward enabling devices themselves to represent, acquire and use a richer notion of context that includes functional and social aspects such as co-located social organizations, nearby devices and people, typical and inferred activities, and the roles people fill in them. We describe a system that learns to recognize richer contexts using sensor data from a person's Android phone along with annotations on her calendar and general background knowledge. Classifier models predict the individual users' context with respect to a mid-level detailed activity he is performing like 'Listening a Talk', 'Walking', 'Sleeping', etc. We report on an evaluation of the individual and generic models in the University setting for predicting context.

CONTEXT-AWARE MIDDLEWARE FOR ACTIVITY RECOGNITION

By

Radhika D. Dharurkar

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Master of Science in
Computer Science
2011

Dedicated to
My Family and Friends

Acknowledgements

It is pleasure to thank those who made this thesis possible. I owe my deepest gratitude to my advisor Dr. Tim Finin. This work would not have been possible without his direction, suggestions, advice and motivation. I want to thank Dr. Anupam Joshi for his guidance and direction for my work. Dr. Laura Zavala has been always there to help me in every small problem I faced. I would like to thank Dr. Yelena Yesha for agreeing to be on my thesis committee. I extend my thanks to my family members and my friends who have been supporting and motivating me all the time.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the advent of various wireless technologies, people want access to the information anywhere and anytime with the personal devices they carry with them. For such extremely mobile scenarios, instead of expecting help from the infrastructure, researchers focus on developing mobile applications which can work with existing infrastructure and explore the capability of smart devices. By using smart devices, we can take advantage of dynamic environmental characteristics and users' information to develop mobile-aware applications which will be more effective and adaptive to users' needs.

Now a days' smart phones and other devices focus to deliver high-quality user experience according to the user's context. Researchers view capability of the smart phones to develop rich variety of useful, enjoyable applications embedded within them to make them intelligent. Context can benefit applications at different levels, namely, in a device (e.g., controlling a ringer depending on place and time), for a user's personal productivity (e.g., intelligently adjusting presence), in an enterprise (e.g., locating a cardiac surgeon who is nearby and not currently in surgery), within the network (e.g., prioritizing data streams that serve a user). Such a notion requires mobiles to be able to capture context information from its surrounding and also by collaboration with devices in vicinity. Checking feasibility for developing such applications with the use of smart phone and context of a user, was motivation of our work.

Though there are different sources from where we can get information pretty easily these days like internet, social networking sites, etc. for capturing context of a person, we need to rely on system which can make use of some resource which is been carried by the user and which can try to capture dynamic changes happening in environment surrounding the user. Though social networking is booming, we cannot rely on them for very specific and time bound queries which are related to the activity of the user. Our system should make use of these different resources we mentioned to get more information about the person's activities, profile, likes, dislikes which is static information along with the dynamic information. This work is part of the bigger research work which involves use of smart phone to capture context of the user and to share it with other devices nearby in secure way. Such system can be used for communication between devices which are in the same context and can help each other to find more information. For example, an application which can help query nearby devices for knowing the parking condition or exchanging information in a meeting. Such things would be feasible if we have the framework setup for the proposed system.

The system described in this paper serves the purpose of capturing activity of the user which can help to develop interesting applications to help users' live a better life. We tried to capture context of the user with the capabilities of smart phone, he is carrying along. The system captured the data of the users' over time and tried to learn about the context of the user and hence the activity the user is involved in. We try to evaluate the performance of the system for activity recognition for individual users and also across the users to understand extent of generalization. Different interesting patterns can be seen with this evaluation and these can be used for applications like recommendation systems, planner, etc. We tried different approaches to collect data for our activity recognition system. This paper would be helpful for people who are researching in similar area and would like to know different problems we faced and how we overcome them.

Context is the set of environmental states and settings in which an application event occurs and is interesting to the user. Schilit [1] divides context into four major classes. The computing context relates with network connectivity, communication costs, and communication bandwidth, nearby resources. User context considers users' profile, location and people nearby. Physical context captures physical attributes such as lighting, temperature, noise, traffic conditions, etc. Last is the Time context which details time of a day, month, and season of the year, etc.

There are other classifications for contexts are location, identity, activity and time. Difference between this classification and the earlier one is the use of activity instead of environment. Environment is related to context and does not add more information to the context. But, Activity describes what is occurring in that situation. For example, given a person's identity, we can acquire related information such as phone numbers, addresses, email addresses, birth date, list of friends, relationships to other people in the environment, etc. With an entity's location, we can determine what other objects or people are near the entity and what activity is occurring near the entity.

Elements of context are gathered from multiple sources, namely, devices, user actions, user surroundings, and network properties. This poses a challenge not only for network architecture but also for the software architecture of applications. Therefore, we can say that effective use of context information is still a challenging problem for application programmers.

The goal of the paper is to survey most relevant literature in this area and develop a framework for context gathering and predicting activity of the user to develop interesting applications in this area. Section 2 describes background study and research work in the same area. Section 3 describes the approach of the research work and next section highlights the implementation aspects of our work. Section 5 gives final summary and conclusion of this research. Ongoing work is also mentioned in section.

Chapter 2

Related Work

Context-Aware have been studies for years in the research community. The Active Badge Location System [2] focuses on predicting location of the user with the use of infrared technology to forward calls to nearby phones. [3] Survey paper mentions research on context-aware systems that support collecting and disseminating context and applications that adapt to the changing context. It gives summary of different applications like Teleporting, Shopping Assistant, Cyber guide, etc. which uses context information. But these applications use small pieces of context information and were specifically developed to suit a particular model. For example, Cyber guide project focuses only on identity and location content types and presentation context-aware feature. Schilit [1] classifies context-aware applications into following categories:

1. Proximate selection: A user-interface technique where the objects located nearby are emphasized or otherwise made easier to choose.

2. Automatic contextual reconfiguration: A process of adding new components, removing existing components, or altering the connections between components due to context changes.

3. Contextual information and commands: This can produce different results according to the context in which they are issued.

4. Context-triggered actions: simple IF-THEN rules used to specify how context-aware systems should adapt.

Pascoe includes features like contextual sensing, contextual adaptation, contextual resource discovery and contextual augmentation. Dey specifies general categories of context-aware features that context-aware applications may support: presentation of information and services to a user, automatic execution of a service, and tagging of context to information for later retrieval [4].

Mobile applications make use of the context in mainly two ways: Active context and Passive context. In former one, application adapts to discovered context, by changing its behavior. In later context, application presents new or updated context to an interested user or makes the context persistent for the user to retrieve later.

Research by Kotz [3] describes some of the mechanisms to sense and deliver the current context to applications. “Retrieval of context-aware applications on mobile devices” [5] paper presents retrieval of context-aware applications on mobile devices tested within their framework (MoBe). There is large scale implementation of Tourist Guide in museums project for context-aware services in public places [6]. [13] Paper presents Cyber guide Project which built a prototype of a mobile context-aware tour guide that provide information to a tourist based on knowledge of position and orientation. Some applications use graph abstraction for collecting, aggregating, and disseminating context information and a variety of critical design issues to support context-aware applications [7].

Since applications need to handle information from different data formats from various sensing technologies, it becomes difficult to abstract data in standard format. Wang paper helps developer to design an implementation framework, specify context rules and create development environment to develop context-aware application [10].

[9] Uses middleware approach to develop context-aware service platform which helps to build and deploy context-aware services. It provides abstraction layer for application developers. The context gathering framework shown in Figure 1 has been designed in a way to facilitate the operational requirements of the other components in the platform. Data structure designed to encapsulate the sensory data will cater to the data modeling requirements from the ontology component. The sensor abstraction written by the developers will handle the collection of low-level sensing from the physical sensors. The framework can be generalized to any implementation since framework and sensor abstraction part is loosely coupled.

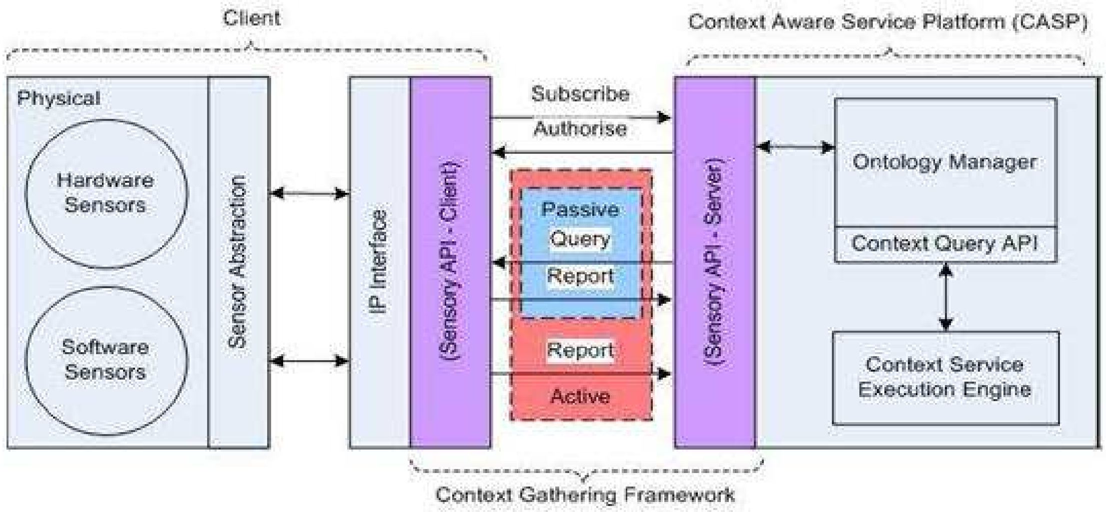


Figure 2.1: Context Gathering Framework

There has been an interesting research done at MIT [11], to infer the friendship network structure of an individual by collecting information from mobile phones over an extended period. They took relationship information from students and compared it with the results of behavioral social network. The data captured by the mobile was able to find out the distinctive behavioral signatures between the friend circles. Phones equipped with the software recorded the data about the call logs, applications used on the phone, phone status, Bluetooth devices in the vicinity, cell tower information, etc. which is been used to infer some information about the social network of a person.

Dartmouth College recently worked on sensing applications on mobile phones with Jigsaw engine developed by them [8]. They mainly focused on sound samples from microphone, accelerometer data, GPS reading and random photos. Their system is not tied to some application but uses sensor-specific pipelines and mentions specific problems observed while using them. Also, we came across paper [12] giving information about research in field of mobile context-awareness. They mention about different sensors which can be used for context-awareness. Also, they mention some more sensors which are not present in smart phones but which can help to greater extent to capture context of the user.

Chapter 3

System Architecture

3.1 Context Modeling and Reasoning Approach

The main motivation of the context-aware system we developed is on activity recognition or location recognition. The aim is to recognize the activity (e.g., "sleeping", "walking", "in meeting", etc.) or the conceptual location (e.g., "at work", "at home", etc.) of agent from a chain of observations on the actions and the environmental conditions of the users' [15]. These observations are usually captured through readings taken from range of sensors present in smart phone carried by user and other sources which provide information about users' context, such as user's profile, location, people nearby and time context.

Machine learning algorithms are used to recognize user's location and activity, both general (at home) and specific (taking notes in class). The base framework for collaborative data gathering [14] was developed to capture user, computing and time context which focuses more on environment than activity. We enhanced the framework to incorporate location, identity, activity context. For example, along with temporal, spatial information, we are capturing information from sources like the user's calendar for recognizing current activity of the user. We use tagging method to obtain a training data which can be used for supervised learning.

To represent more inclusive and higher level notion of context in context-aware systems, we need models which can represent and reason over it. Our model captures user's location, surroundings which include devices and people, and activity he is performing. We have developed platys ontology which is light-weight, high level ontology. It models place according the activities taking place by users. We use OWL (Web Ontology Language) and inference mechanisms to model it. We are in progress of using these ontologies as priors to machine learning inputs.

The base framework was developed to cater to three different use cases. First was to have a request response field survey. Second, we can have context-based reminders for users. Finally, a Business service which will be providing special offers to specific set of people. Our modified framework can be used in all such use cases but especially we focus on modeling users' activities. Therefore, we can have interesting applications which helps users' to keep track of their activities over time, make changes in their schedules according to the recommendations for specific programs (gym workouts, study schedules, program meetings, update calendar, etc). Also, applications which can locate specific service or people in nearby, adjust presence of person, etc.

Figure 2 shows high level system architecture for our system. First module is data collection module which captures different information through smart phone and also tagging from the user. We will cover details about it in the next section. The raw input will be processed and cleaned. We extract important information from it and give it to the parse which will work on the data and generate a feature vector. Context ontology will be used as prior knowledge before classification. Feature vector is obtained from the parser and also saved in relational database. Classifier works on the feature vector and outputs the prediction for activity.

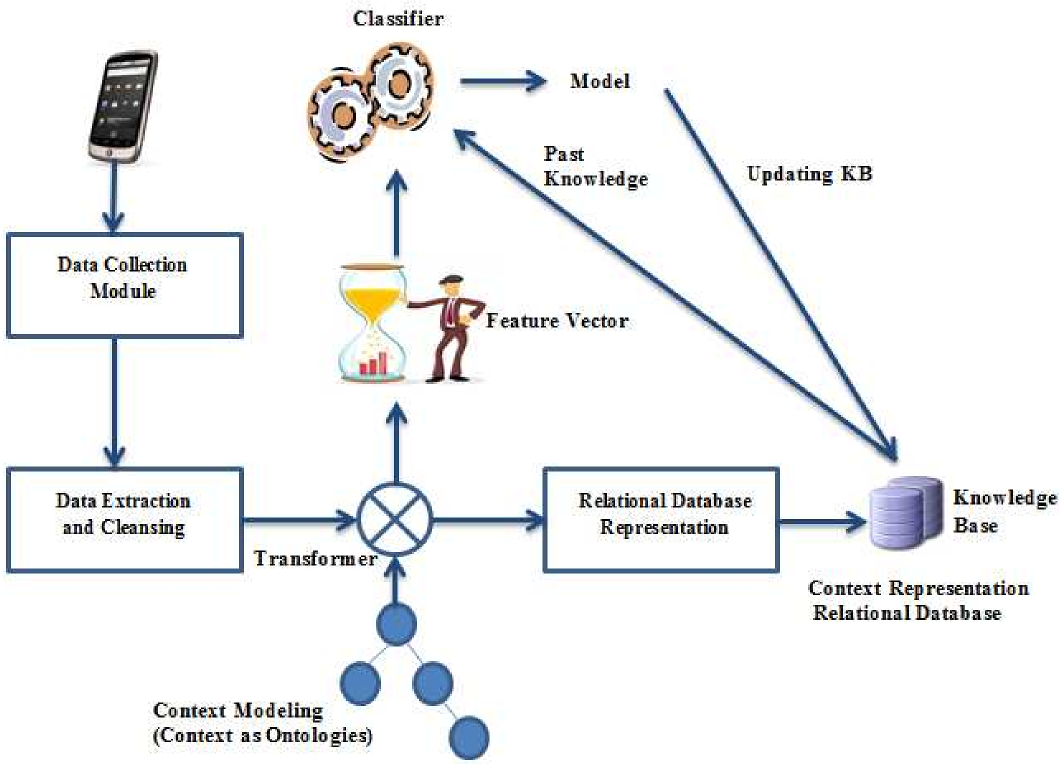


Figure 3.1: High Level System Architecture

3.2 Information Capture

We used the base framework [14] for data collection initially. That approach used the agent on phone which is configured to poll for sensor data and sent it to server over internet every fifteen minutes. The battery usage is huge and also connection to internet is required to transfer data. We used this framework for the toy experimentation. We added functionality where phone can capture the details of Google calendar of the user. Since most of the users keep calendar up to date, this can help us a lot while recognizing place and activity e.g. most of the meetings and holidays are marked by the users on calendar.

Further, we decided to remove the dependency of data transfer on Internet. Hence, it reduced battery usage of the phone while transferring data. We decided to save the data on phone memory and upload it periodically, mostly after a day or two. We had two options to do that. We can store data in SQLite database on Android phone or simple text files. We took approach of storing data on text files since we have large amount of data to be captured e.g. Megabytes of data for a single day. And this would be helpful to save data for days if the user forgets to upload data to server periodically.

Figure 3.2.1 shows snapshot of Google calendar highlighting the important information which can be captured from our system.

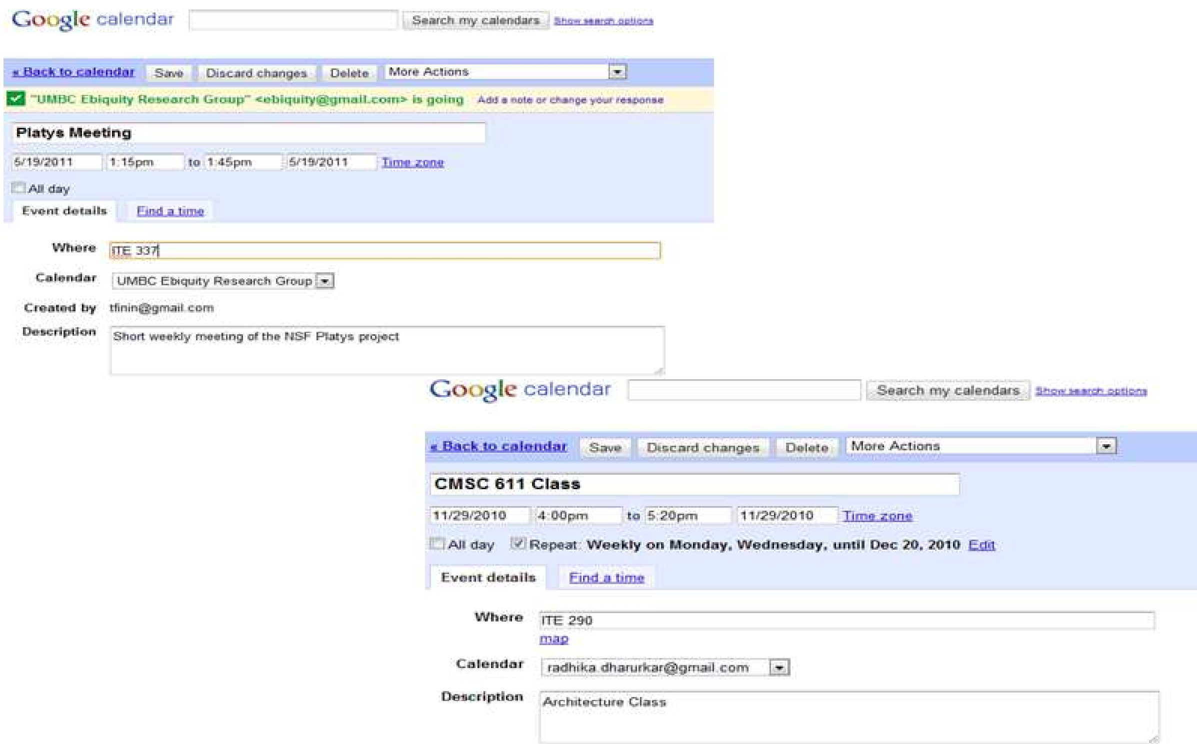


Figure 3.2.1: Snapshot of Google Calendar

Figure 3.2.2 shows our data collection application which runs on android smart phones. We use tagging method to collect current activity and the place where it is performed. We used Nexus one and Droid Incredible mobile phones for experimentation. Left panel allows user to select one of the many places user can visit. Right panel selects the current activity for the user. Both of these can be edited by user to add/remove locations/activities, keep frequent places/activities on top. For hierarchical places/activities, we allow users' to select multiple e.g. user can be having lunch while listening to a talk. We collect the data every two, five or twelve minutes according to the users' preferences. Some activities are observed for short period of time. Therefore, we allow users' to log those activities with interval of two minutes. Users' need not have to select place and location each time if they are doing the same activity at same place which was tagged before. The application would by default select the place and activity which was selected before but will put a flag saying it was not changed by the user.

Figure 3.2.2: Data Collection Program

3.3 Data Integration

Lot of information has been used to give input to the classifier for predicting the activity. We capture timestamp and use it to give input to machine learning algorithm as time of day, day of week, weekday or weekend. We collect lot of sensor information viz. orientation, magnetic, accelerometer, proximity, ambient light and noise. Also, we capture latitude, longitude, geographic location, call statistics (missed calls, answered calls, duration, etc.), Wi-Fi and Bluetooth ids (Paired, non-paired) in surrounding. Data from user's Google calendar has also been considered for classification.

Following figure shows snapshot of the captured data. It shows most of the attributes captured by our data capture program. We capture data for different sensors though we did not list all of them here.

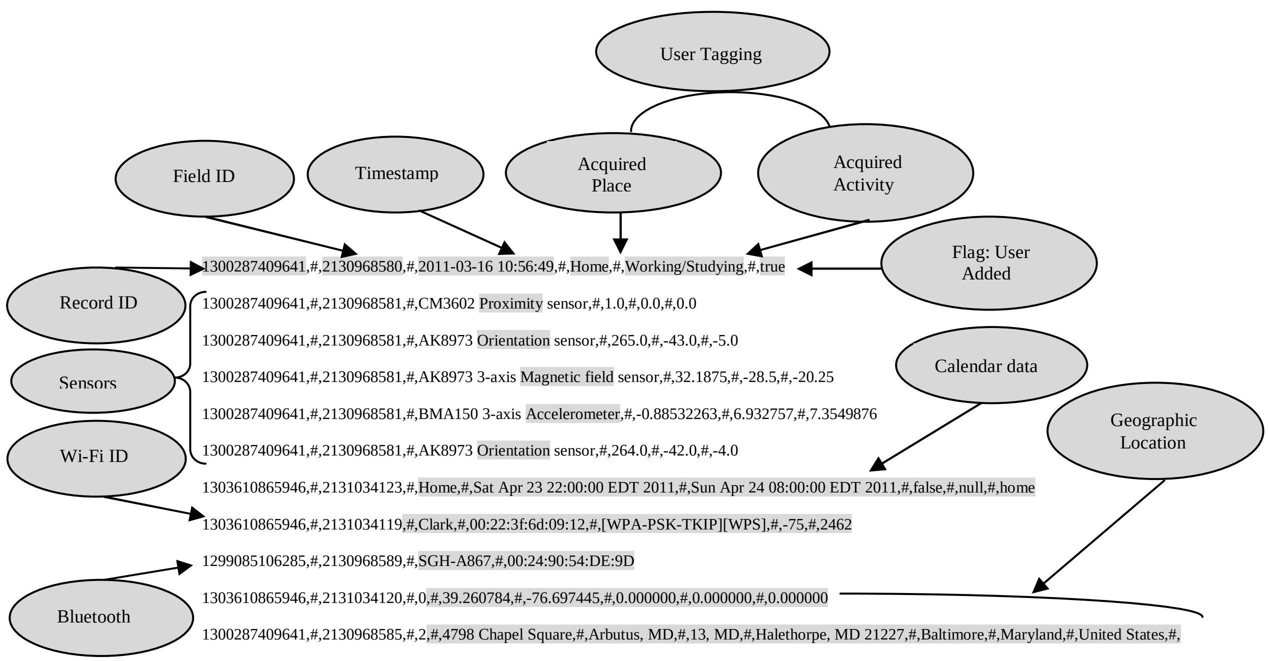


Figure 3.2.3: Snapshot of Captured Data

We had to face different problems while data integration since data was in the form of raw text with records collected for a longer duration. Each file will have data collected for a day or two with number of records captured within a day. Each entry of the data also contains multiple sensor values for each sensor. Object matching was tricky. Inconsistencies in some attributes also caused some redundancies in resulting data set.

3.4 Activity Recognition

Our data collection program was set to log several activities since we wanted to log finer details. Though we knew our model may not be able to differentiate between most of the activities, because of the limited set of sensor values present in smart phones, we tried to capture activities which can be put in a hierarchy of activities. E.g. in class- listening to lecture or taking notes can be a finer detail of being in class. Our model allows selecting finer details since we assume the model is intelligent enough to understand the hierarchy of activities. Same analogy is applied to place also, e.g. ITE227 is considered as part of School. Activity recognition using various sensors wore on body helps to understand finer details of the activity performed by the user. But our model just concentrated on temporal, spatial, sensor (available in smart phone) and profile information of the user.

Table 1, 2 shows different places and activities logged by our application. Next chapter will talk about accuracy for each of the activities. But it is seen that we get good accuracy only for few activities which are highlighted in the table for first few experiments.

Activities	
Working/Studying	**In Meeting**
Sleeping	Watching TV
Reading	Watching Movie
Driving/Transporting	**Talk Listening**
Chatting/Talking on Phone	Other/Idle
Coffee/Snacks	Dinner
Walking	**Shopping**
Cleaning	Playing
Cooking	Talk Presenting
Lunch	Listening Music
Class Taking Notes	**Outdoors**
Class Listening	**In Class**

Table 3.4.1: Activities Captured

Places	
Home	ITE227
Grad Lab	Sondheim Hall Corridor
Sondheim Hall 110	Friend’s Place
Restaurant	Commons
Library	Chick-Fill-a/Starbucks(School)
Lab	eBiquity Lab
School / Work	Admin Building
ITE Corridor	Sondheim Hall 208
Shopping Mall	Outdoors
ITE 346	Office
Coffee Shop	ITE 325
Catonsville Library	Miller Library
ITE 338/377	Grocery Store
Theatre	Elsewhere

Table 3.4.2: Place options for user

Chapter 4

Implementation and Evaluation

4.1 Data Capture Model

4.1.1 Experimental Setting:

Initial toy experimentation was improvement over the base framework to capture some more sensor values from phone, timestamp, latitude, longitude, Wi-Fi devices around and data from users' Google calendar. The data was captured in MySQL database on the server. Then we classified it in open source tool Weka, which is a collection of machine learning algorithms for data mining tasks. We compared performance for Naïve Bayes, Decision Trees, LibSVM algorithms. We used different test options to find accuracy for our data viz. use of training and test set, cross-validation, percentage split. Our final model captured different sensor values, timestamp, geo-location, data from users' Google calendar, information about Wi-Fi, Bluetooth devices around. We capture most of the sensor values from android phone. Since we capture data every two, four or twelve minutes according to users' preference, we decided that capturing just one reading of a sensor value from the device would not be helpful. This is because values for sensors like noise, accelerometer, and proximity, and etc. change frequently, smart phones can sometimes miss to capture some value. Also, we need to capture some sensor values over duration of time to get idea of the pattern. Therefore, we capture changing values of sensors for 1 minute in each data capture cycle. We carried out experiments with three android phones viz. two Nexus Ones and one Droid Incredible. Two of the users' were students, one first year master's student and another one research assistant in second year of master's. Third one was a post doctorate working with the lab. All of them were carrying the phone all the time with them and log all the places and activities they do. They logged activities which were generalized by us before and also put some more places and activities which they have been performing peculiarly. Since, same place can be a workspace for one person and can be a school for other. Therefore, we allow individualistic tagging. This made us get a log of finer activities and get more data. All this data capture was in a plain text file. Then we wrote a java parser which can understand the data and average over number of values captured for sensors. We put this data for classification in Weka and compared performance on different algorithms.

We used Intel Core i5 CPU 2.53GHz, 64 bit Windows 7 OS machine to parse the data and put Mysql version 5.2.31 CE database. We used a java program to work on the text file containing raw data and output the comma separated file.

4.1.2 Database Structure:

When user starts any activity, he selects one or set of places and activities and hits start. The application will capture all the attributes related to the context of a person and put it on a data log file residing in the phone memory. We take out the file after every day or so and give input to the parsing program which can converts it to a comma separated values file. We use scripts to put that data in the MySQL relational database management system for further manipulation of data.

Following table shows all the data which is collected by our application with some fields added by the parser to help classification. Timestamp has been divided in some fields like time of day, day of week, weekend (yes or no), since it helps the machine learning algorithm to classify better. It has been observed that sometimes users' forget to tag the activity and place. Therefore, we tag each record with user added field which mentions if the place and activity is been selected by user or the default (last value recorded) is been saved. Each of the Wi-Fi Id which is captured by the user's device in some point of time is been used as one of the attribute for classification. Therefore, we have 679 Wi-Fi Ids as an attribute. If any new Wi-Fi id is been observed, parser will set the undefined flag. All the Bluetooth devices have been classified in two categories viz. paired and not paired since paired devices give more information about the known people (e.g. People working in same group, found in most of the meetings, friend circle, etc.).

Data	
Timestamp	Day of week
Weekend (True/False)	Place
Activity	User Added (True/False)
Orientation (Azimuth, Pitch, Roll)	Magnetic Field
Accelerometer (Gx, Gy, Gz)	Light
Connected Wi-Fi ID	Wi-Fi devices List
631 Wi-Fi IDs (True/False)	Undefined Wi-Fi ID (True/False)
Latitude	Longitude
Altitude	Location Bearing
Location Speed	Proximity
Geocode	Calendar data
Paired Bluetooth devices	Unpaired Bluetooth devices

Table 4.1: Collected Data

4.2 Experimental Settings:

4.2.1 Toy Experiment:

We already introduced the toy experiment in last section. We will go over the details in this section. The data was collected by three android phones and was sent to the server twice every minute. The server collected data in the MySQL database which was then converted to comma separated values file and put for classification in Weka. The server machine was Ubuntu machine on internet. Following tables shows the statistics of data collected. Each device number corresponds to a different phone with the exception of 45 and 46 which belong to the same phone. Sensors recorded for each example were: latitude, longitude, Wi-Fi count, Wi-Fi ids, battery percentage, light (some nulls observed), proximity, and user present (some nulls observed), Google calendar data. Number of examples (records) per event is last column of table. For example, device 44 only got four sessions recorded, three of them at the office and one at home (three on Fridays and one on a Monday). The number of records corresponds to sensor readings (every half minute) during those sessions, which in fact do not vary that much. We used the calendar data to understand the label to certain extent programmatically. Though not all users' had the calendar data synched with their schedule. Users' then manually tagged the instances.

Device	**Date**	**Start time**	**End time**	**Label**	**No(Records)**
44	Fri 2010-11-19	11:17:52	11:55:52	Office	77
44	Fri 2010-12-03	10:19:14	10:57:14	Office	74
44	Fri 2010-12-03	20:02:54	20:43:11	Home	78
44	Mon 2010-12-06	10:44:04	13:59:09	Office	51
45	Fri 2010-12-03	19:21:20	20:07:20	Home	93
45	Fri 2010-12-03	16:00:35	18:16:53	Lab	98
45	Sun 2010-12-05	21:14:22	22:15:21	Home	66
46	Mon 2010-12-06	16:10:16	16:58:47	Class	60
46	Mon 2010-12-06	14:20:16	14:47:16	Elsewhere	54
46	Mon 2010-12-06	15:29:03	15:34:03	Elsewhere/ Class	8
46	Mon 2010-12-06	15:35:42	15:42:12	Lab/ Class	14
46	Wed 2010-12-08	22:45:37	22:59:37	Home	29

T0able 4.2.1.1: Labeled Records

233 examples without label will be pretty much grouped into the following events:

Device	Date	Start time	End time	Label	No(Records)
41	Sat 2010-11-11	12:50:10	14:05:13	TBD	67
43	Wed 2010-12-08	20:01:06	20:50:36	TBD	99
45	Fri 2010-12-03	15:04:29	15:59:20	TBD	67

Table 4.2.1.2: Un-labeled Records

The data collected is discrete since the framework we used was not stable enough to connect to sever all the time and upload data. The application timed out after a certain period of time. But we still tried to work on the data we collected. We totally had 720 labeled samples for this experiment. Following tables shows results after classification. We used different machine learning algorithms to evaluate our data. Table 4.2.4 shows accuracy for student's data which had 720 data samples. We used cross fold method with 10 folds.

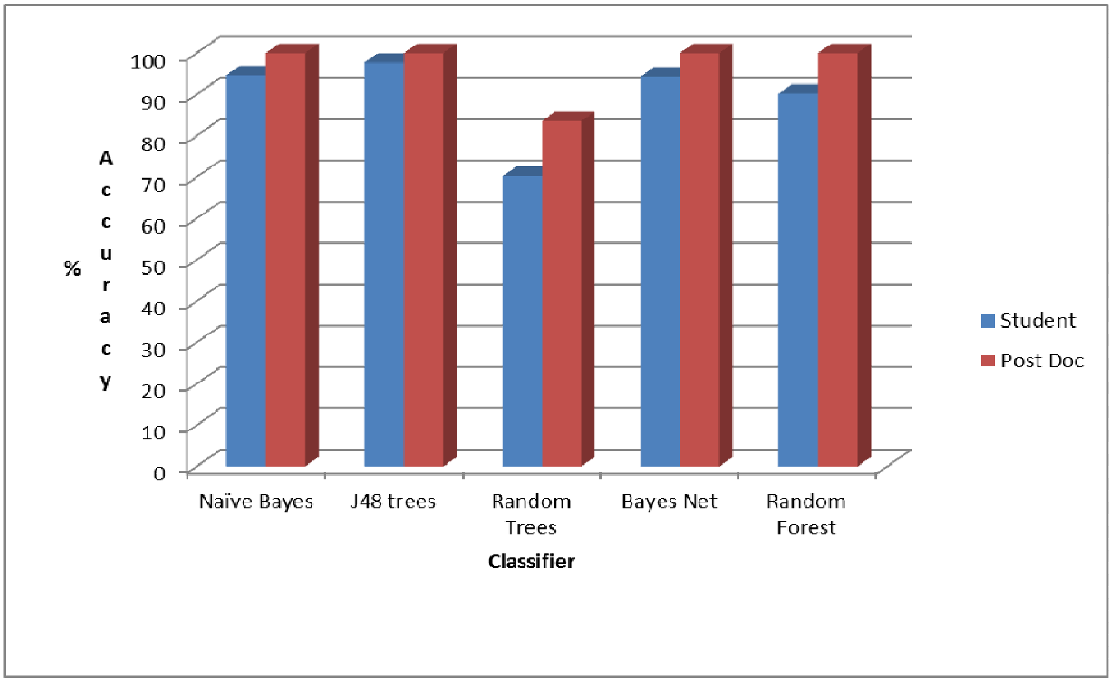
No	Classifier	Correctly Classified		Incorrectly Classified	
		Instances	%	Instances	%
1	Naïve Bayes	607	92.9556 %	46	7.0444 %
2	J48 trees	651	99.6937 %	2	0.3063 %
3	Random Trees	469	71.8224 %	184	28.1776 %
4	Bayes Net	637	97.5498 %	16	2.4502 %
5	Bagging with J48 trees	651	99.6937 %	2	0.3063 %

Table 4.2.1.3: Data Accuracy – Student

No	Classifier	Correctly Classified Instances %		Incorrectly Classified Instances %	
1	Naïve Bayes	280	100 %	0	0 %
2	J48 trees	279	99.6429 %	1	0.3571 %
3	Random Trees	256	91.4286 %	24	8.5714 %
4	Bayes Net	280	100 %	0	0 %
5	Bagging with J48 trees	279	99.6429 %	1	0.3571 %

Table 4.2.1.4: Data Accuracy – Post Doctorate

Following graph shows accuracy for different classifiers for both users.



Graph: Toy Experiment

4.2.2 Data Collection Part 1:

We had new data collection application (Figure 3.2.1) up on our android smart phones for the data collection phase. We evaluated the data after capturing it for almost two weeks. We used three android phones for this evaluation. Data collected in this case was pretty continuous data as compared to the one before. Though we have some gaps in between the data since users' forget to start the application or phone is down on battery. First we will concentrate on the data collected for a student who works as research assistant in school and attends one lecture and few meetings in a week. The data logs activities done on weekdays and weekends also. Following Table shows range of activities collected by the application and also mentions the number of instances.

No	**Activity**	**No of Instances**
1	Working/Studying	218
2	Sleeping	195
3	Reading	39
4	Driving/Transporting	25
5	Chatting/Talking on Phone	49
6	Coffee/Snacks	54
7	Walking	26
8	Cleaning	26
9	Cooking	17
10	Lunch	19
11	Class Taking Notes	9
12	Class Listening	6
13	In Meeting	7
14	Watching TV	9
15	Watching Movie	1
16	Talk Listening	3
17	Other/Idle	4
18	Dinner	3
19	Shopping	1

Table 4.2.2.1: Data collection Part 1: Activity statistics (Student)

The data has been collected for thirteen days which includes data for two weeks. Following table puts down the statistics about this data.

No	**Date**	**No of Records**
1	03/01/2011	65
2	03/02/2011	12
3	03/03/2011	63
4	03/04/2011	48
5	03/05/2011	24
6	03/06/2011	38
7	03/07/2011	26
8	03/08/2011	40
9	03/09/2011	71
10	03/10/2011	60
11	03/11/2011	88
12	03/12/2011	62
13	03/13/2011	52
	Total	649

Table 4.2.2.2: Data collection Part 1: Data Period statistics (Student)

The data we worked on for this part was raw data collected by smart phone from different sensors and users' calendar. The parsing algorithm written by use would take the raw data from the text files and try to transform it into different feature sets and values. According to our collection program, each of the instances can have multiple records for the sensor values. This parser will average over the values and make the output comma separated file. We also use some techniques to clean up the data. Discrepancy detection for different attributes is been done at this data transformation stage.

After the data transformation, we tried to discretize the data to large extent which helped to get better results. We used binning and concept hierarchy techniques to filter the data at unsupervised attribute level in Weka. Discretization techniques divide the number of values for a continuous attribute into intervals which reduces and simplifies the data. Use of such techniques helped us to have a concise, easy-to-use knowledge-level representation of mining results. We used concept hierarchy technique to represent low-level concepts with higher-level concepts (such as timestamps represented as time of day, day of week, night, morning, etc.). Such kind of generalization loses some data but this is been considered as consistent representation of data which is easier to interpret. Binning methods smooth the data consulting the neighborhood of values. We use equal frequency beans for most of the attributes (sensor values) in our dataset.

We put this data for classification in Weka. We run different algorithms on it with different test options. Table 4.4.2.3 shows performance of this data for some algorithms. We tried to compare five classifiers here. We did not analyze the whole confusion matrix but only few major activities which have been conflicted with others. We can see that except decision trees all others performed well in classifying the correct activity. Strong independence assumptions played a significant role in here.

No	Classifier	Correctly Classified Instances %	Incorrectly Classified Instances %
1	Naïve Bayes	549 77.2152 %	162 22.7848 %
2	J48 trees	343 48.2419 %	368 51.7581 %
3	Random Trees	705 99.1561 %	6 0.8439 %
4	Bayes Net	612 86.0759 %	99 13.9241 %
5	Random Forest	711 100 %	0 0 %

4.2.2.3: Data collection Part 1: Performance of machine learning algorithms

Following table shows the statistics for the confusion matrix.

Total	Main Activity	Conflicted	Conflicted
54	Coffee/Snacks	Working/Studying 12	Sleeping 5
218	Working/Studying	Coffee/Snacks 5	Sleeping 8, Chatting 8
39	Reading	Working/Studying 19	Sleeping 4
26	Cleaning	Working/Studying 10	Sleeping 2
195	Sleeping	Working/Studying 9	
17	Cooking	Working/Studying 5	Sleeping 3, Cleaning 2
49	Chatting/Talking on Phone	Working/Studying 14	Sleeping 2 ,Coffee/Snacks 2
6	Class-Listening	Class-TakingNotes 2	
3	Talk-Listening	Class-TakingNotes 1	Working/Studying 1
1	Watching Movie	Sleeping 1	
3	Dinner	Working/Studying 3	
9	Watching TV	Working/Studying 3	Sleeping 6
1	Shopping	Working/Studying 1	

4.2.2.4: Data collection Part 1: Confusion matrix

All we observed that these algorithms worked on the input data taking nominal values like Wi-Fi Ids, Bluetooth devices, etc. These values were a set of Wi-Fi/ Bluetooth

Ids which can be ordered in any way. Also, at the same place we may not see the exactly same set at another time. All the machine learning algorithms cannot handle this situation of "bag of words". Also, in this part of experiments, we did not work on the data cleanup. We just took the data we collected form the phone without any noise removal. Therefore, the accuracy we see here cannot be the real accuracy for activity recognition. And the poor performance of decision trees can be real.

Following tables shows the statistics for the data collected for post doctorate person. If we compare this dataset with the earlier one, we can see the data period of collection is almost similar. Though, some activities and their number of instances differ. For example, if you see data collected of "Sleeping" activity, there is vast difference in numbers of the instances recorded. The reason for this difference was that the second person had his phone out of charge most of the times at night. Therefore, logging sleeping activity is not recorded for all the time. This Error in data collection makes a lot of difference in our analysis.

No	**Classifier**	**Correctly Classified**		**Incorrectly Classified**	
		Instances	**%**	**Instances**	**%**
1	Naïve Bayes	704	90.0256 %	78	9.9744 %
2	J48 trees	631	80.6905 %	151	19.3095%
3	Random Trees	778	99.4885 %	4	0.5115 %
4	Bayes Net	717	91.688 %	65	8.312 %
5	Random Forest	775	99.1049 %	7	0.8951 %

Table 4.2.2.5: Data collection Part 1: Performance of machine learning algorithms

No	Activity	No of Instances
1	Working/Studying	525
2	Sleeping	72
3	Reading	5
4	Driving/Transporting	29
5	Coffee/Snacks	3
6	Walking	14
7	Cooking	11
8	Lunch	9
9	In Meeting	6
10	Watching TV	18
11	Watching Movie	2
12	Talk Listening	8
13	Other/Idle	78
14	Shopping	2

Table 4.2.2.6: Data collection Part 1: Activity statistics (Post Doctorate)

Total	Main Activity	Conflicted	Conflicted
525	Working/Studying	Other/Idle 9	Sleeping 4 , Watching TV 6
9	Lunch	Working/Studying 3	Other/Idle 1
72	Sleeping	Working/Studying 19	Other/Idle 2
11	Cooking	Working/Studying 3	Sleeping 2
78	Other/Idle	Working/Studying 13	Walking 1
18	Watching TV	Working/Studying 7	Other/Idle 1
2	Shopping	Cooking 1	

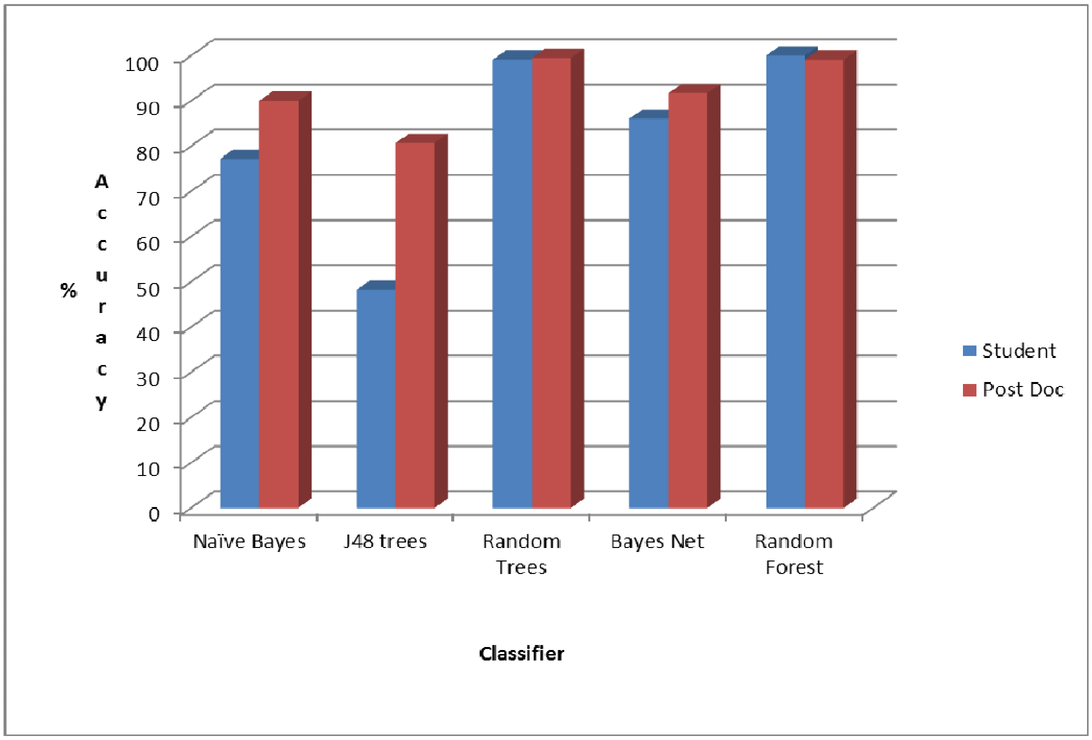
Table 4.2.2.7: Data collection Part 1: Confusion Matrix (Post Doctorate)

No	Date	No of Records
1	3/2/2011	36
2	3/3/2011	26
3	3/4/2011	60
4	3/5/2011	68
5	3/6/2011	77
6	3/7/2011	60
7	3/8/2011	26
8	3/9/2011	63
9	3/10/2011	94
10	3/11/2011	84
11	3/12/2011	35
12	3/13/2011	81
13	3/14/2011	29
14	3/15/2011	54
	Total	793

Table 4.2.2.8: Data collection Part 1: Data Period statistics (Post Doctorate)

If you compare the confusion matrix for this data with earlier one, we could realize that there are less instances of working activity being confused with the sleeping activity. The major reason for this to happen is as follows. The student stays in small apartment where he works, watched TV, drinks coffee and also eats at almost same location (hardly any difference in the sensor values collected). First person has most of his working instances at home than at school. Also, there are lot of instances of sleeping activity recorded which states that he has not missed on collecting data as opposed to the second person. The data collected by second person seems to be good training data since the activities done are associated with place most of the times and therefore predicted with higher accuracy. First data has many instances of same activities at different places (e.g. working in school and at home) and many instances of different activities at same place (e.g. having dinner and studying at same place). This confused the machine learning algorithm at higher extent lowering its performance.

Following graph shows the accuracy for different classifiers for both the users.



Graph : Data Collection Part 1

4.2.3 Data Collection Part 2:

Good accuracy given by Naïve bayes in the first set of evaluation was deceptive. We had captured timestamp and used it as one of the feature. Also, we had captured Wi-Fi devices and Bluetooth devices in the surrounding. This was collected as a set of values. Each time the set cannot have exactly same ids at same location. This condition was not handled by the machine learning algorithm. It was considering each set as one entry. Therefore we preprocessed the data for such attributes. We removed timestamp and split it into five Boolean features namely ‘weekend’, ‘morning’, ‘afternoon’, ‘evening’, ‘night’. We learned the Wi-Fi ids and geographic information for the individual and represented the complete list as features like bag of words analysis. This way some of the overfitting was removed. Also, we tried to generalize some of the activities. We also tried to evaluate our data on SVM (Support Vector Machine), machine learning algorithm. We used LibSVM in Weka but decision trees seem to outperform. Following are the details for our experiment. Bagging with j48 trees seem to outperform in this scenario.

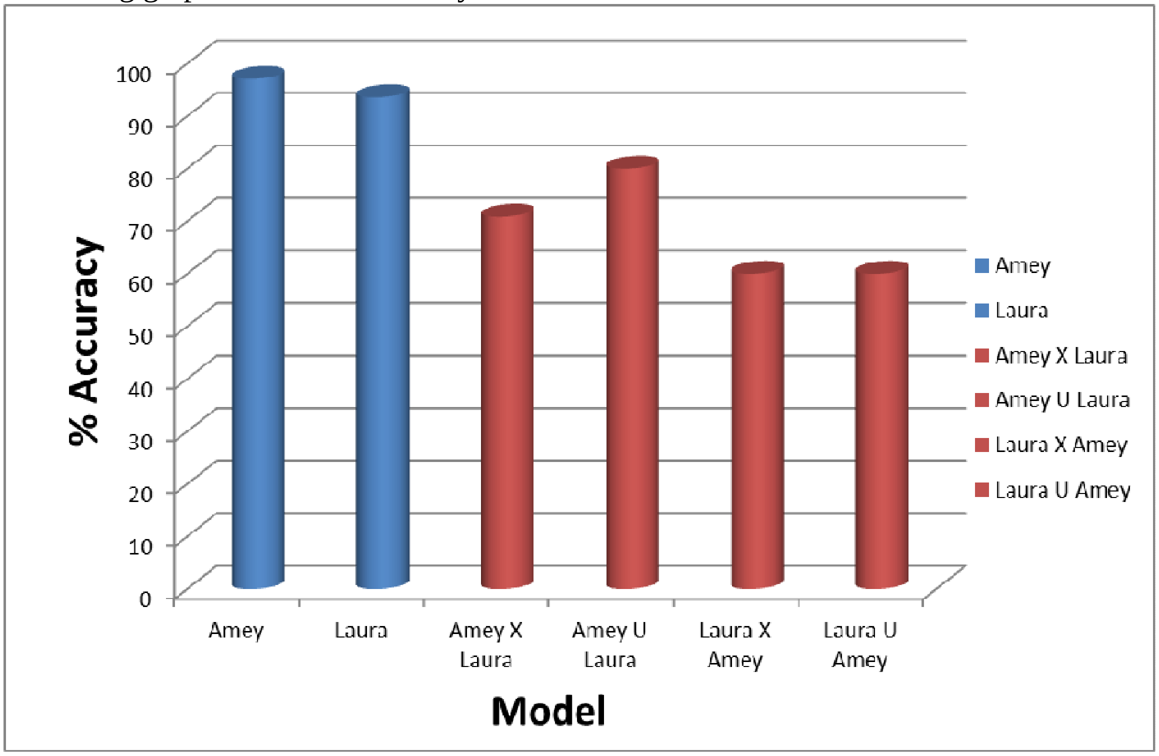
No	Classifier	Percentage split 66%	Cross Validation 10 Folds
1	Naïve Bayes	65.6109 %	64.0553 %
2	J48 trees	83.7104 %	84.0246 %
3	Bagging + J48 trees	86.8778 %	86.7896 %
4	LibSVM	66.5158 %	64.2089 %
5	LibLinear	73.3032 %	74.1935 %

Table 4.2.3.1: Data Collection Part 2: Performance of Machine Learning Algorithms

No	Activity	No of Instances
1	Working/Studying	392
2	Sleeping	157
3	Walking	52
4	In Class	36
5	Outdoors	1
6	In Meeting	6
7	Talk-Listening	3
8	Other/Idle	1
9	Shopping	3

Table 4.2.3.2: Data Collection Part 2: Activity Statistics

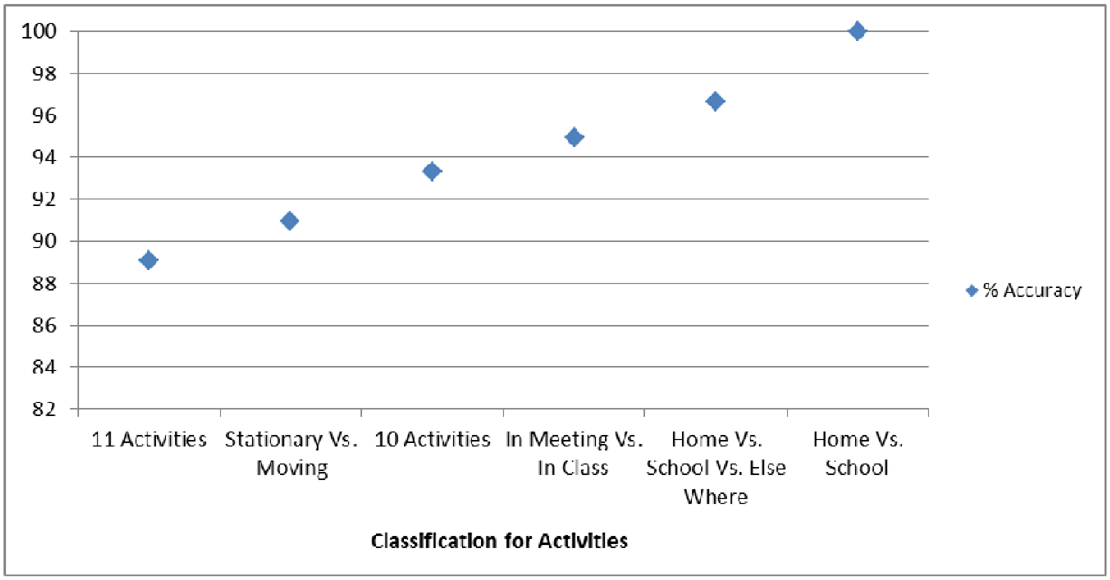
Following graph shows the accuracy



Graph: Data Collection Part 2

4.2.4 Data Collection Part 3:

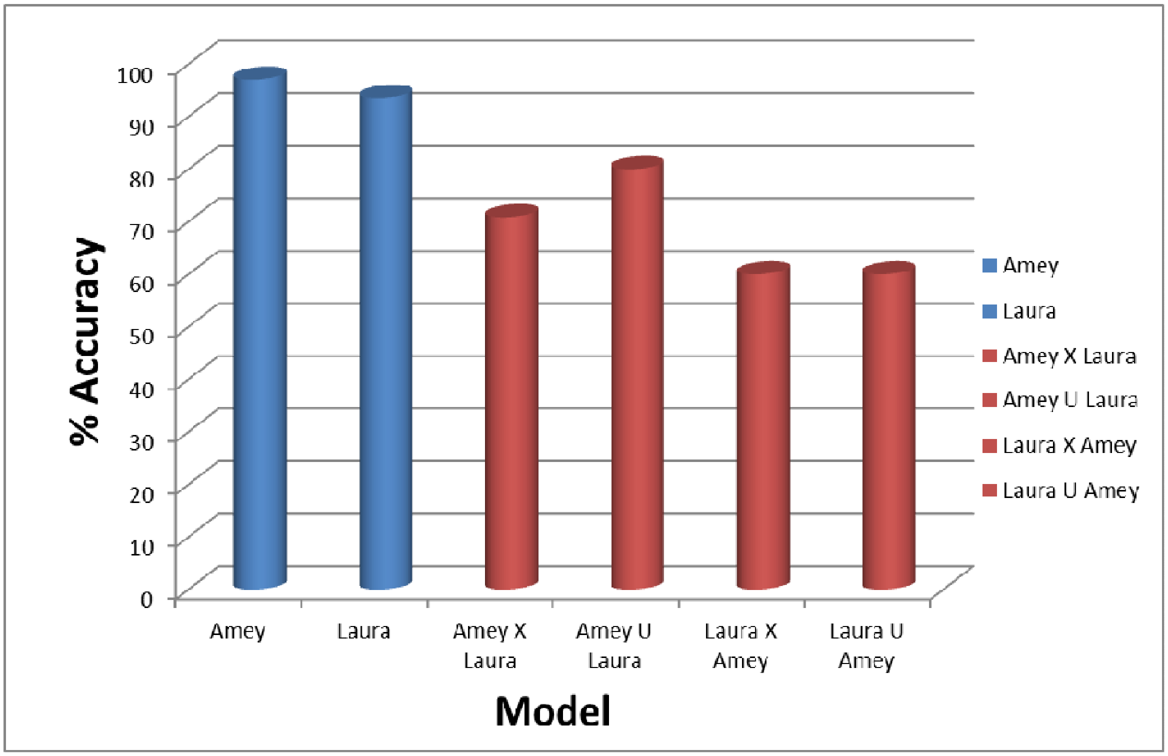
We found that it was hard to predict the activities to the finer extent because of occurrences of different activities at the same place and less information from smart phone. Therefore, we tried to find out accuracy for simple use cases. Our model could find out "At School", "At Home" or "Else Where" with 96.67% with decision trees classifier. We get 100% accuracy for "At School" or "At Home" as compared to MIT's model [11] which finds it with similar accuracy, using Hidden Markov Models. Also, we tried to cleanup data by removing noise from some records. Following graph shows the performance.



Graph: Performance of Machine Learning Algortithms

General Model:

Since our model is developed for an individual user, we needed to find the extent of generalization. We got very less accuracy for fine grained activities. Therefore, we tried to evaluate accuracy for the activities which can be predicted almost correctly if we train the classifier with one person's data and test it with others. Following graph shows the accuracy for such generalization. This accuracy is for 'Walking', 'Sleeping', 'Lunch', 'In Meeting', 'Watching a Movie' activities. Since Wi-Fi ids and Geo-codes have been handled as bag of words, we use intersection and union of them and evaluate the accuracy. Following graph shows the accuracies for percentage split at 72% for different combinations of data.



Graph: General Model

We also experimented on finding activity of the user with the help of time and location. We considered following attributes: Day, morning, afternoon, evening, night, activity, latitude, and longitude. We evaluated this model for ten and eleven activities for a student.

No	Model	All Attributes	Time and Location
1	11 Activities	90.42	70.69
2	10 Activities	93.39	74.61

Table 4.2.4.2: Comparison of Performance for new models

4.3 Data Modeling

Following diagram shows the upper level place ontology developed.

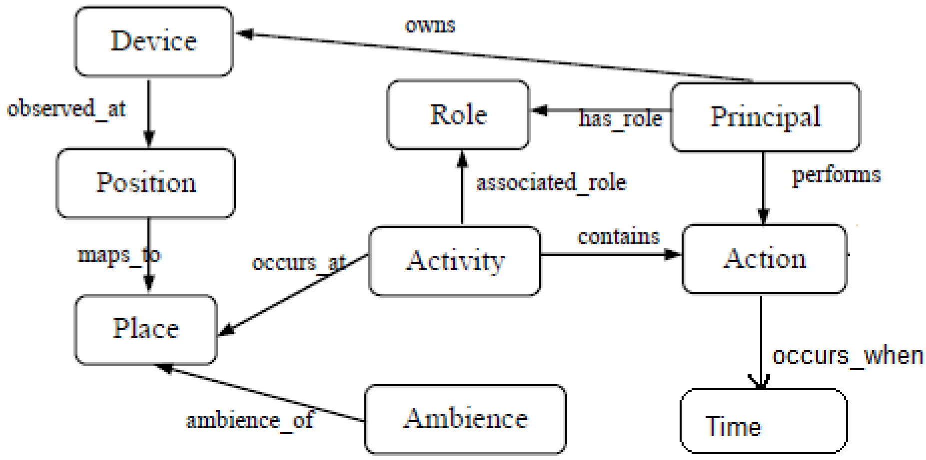


Figure 4.3.1: High Level Ontology

Following are the core concepts of this ontology:

- Place – a concept that captures the semantic notion of a location. Examples include class, restaurant, gym, etc.
- Principal – a person or resource whose place can change over time. An example of a non-human principal is a projector which can be in a class, a seminar, or even a campus_cinema_theater.
- Action – a task performed by the principal. Examples include silencing_the_cell_phone, rejecting_call, sending_text, etc. Actions are temporal concepts.
- Role – the character(s) played by the principal in a given place. Example roles in a class would be teacher, student, teaching assistant, etc.
- Activity – an association entity that ties together actions and roles. Examples include listening_to_lecture, delivering_the_lecture, working_out etc.

- Ambience – concepts describing the environment of the principal. Examples include noise level, temperature, etc.
- Device – a resource which helps us to capture the context of a person who carries it by capturing context information from the sensors and other user specific sources (calendar, etc.) of information.
- Position – a geographic location which is been captured by the device carried by a person. It maps to the place concept. Examples include latitude, longitude and geo-location.
- Time – concepts describing temporal aspect of the action. Example is timestamp captured by device.

Chapter 5

Conclusion, Discussion, and Ongoing Work

Conclusion

Our work contributed to have a framework setup for activity recognition. We developed an application for android phone which can be used to collect information from several sources automatically. We worked on the data to have a good feature set for activity prediction. We evaluated on different machine learning algorithms. The analysis was in line with most of the researched done in past in same area. E.g. Home vs. Work gave accuracy of 100% which is compared with 95% accuracy given by MIT project which used Hidden Marcov Models. Mid-level detailed activity recognition by Bao and Intille (MIT) also had similar results. Our supervised learning approach proved to be good for mid-level detailed activity predictions. E.g. we had almost 88% accuracy for predicting 9 activities of an individual in University scenario. We also tried to find a set of activities which can be generalized across users.

Limitations

The research was limited to just two android smart phones and some volunteers to collect data for us. This data was not sufficient to collect varied activities since the users were the students in school. There were not different roles to be explored to take advantage of “Roles” in order to find out the activity. Also, occurrence of different activities at same (small) place was a problem. We did not have expertise to process on sound samples or images to help our prediction. Our model just tried to divide the values in different frequency bins.

Ongoing Work

Our goal was to predict the activity of the user according to the context of the user collected from the smart phone carried by user. We had less data to capture the exact activity of the user since most of the observed values helped to predict the place accurately but not the activity. Then also we tried to be modestly accurate to find some set of activities. We tried to check the generalization of this model. Since we could not do much in this regards with just data from three users over some period of time, we would like to collect more data for more users and work on it.

The research involved lot of work on the raw data collected from the smart phone. It was cumbersome to work on such data. Therefore, some automation can be done to work with such data. We concentrated here on machine learning approach. We can surely take help of semantic web to generalize or classify into more specific activities according to the requirement. Also, some rules can help improve the accuracy of the experiments.

Following were some of the improvements which can be done to improve prediction accuracy:

- Learn the sleeping time, working time for a person according time of day, day of week and its habits and suggest classifier some set of activities which the person could be doing.
- Classify activities according to the place like home, school or outdoors. Since we have good accuracy on predicting place, we can come up with small set of activities (Priori Analysis). This can help us to get more accuracy on predicting the activity of the user. We are evaluating how we can combine these priors and machine learning approach together.
- Parse calendar activities to predict kind of activity it falls in. We can use semantic web to help us in this regard.
- Assigning roles to the individual can help us to disregards some of the activities. For example, if a professor is in class and its context can predict that he is in class, we can narrow down all the activities seen in class to some small set according to his role.
- Training for finding threshold values for different sensor values to predict particular activities like audio volume to infer if person is at a party or at a restaurant or in class can improve the results. We need lot of training data and people and devices to carry out such experiments.

Bibliography

[1] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pages 85-90, Santa Cruz, California, December 1994. IEEE Computer Society Press.

[2] Roy Want , Veronica Falcao , Jon Gibbons. The Active Badge Location System (1992) ACM Transactions on Information Systems

[3] Guanling Chen and David Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000.

[4] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pages 304{307, London, UK, 1999. Springer-Verlag.

[5] Stefano Mizzaro, Elena Nazzi, and Luca Vassena. Retrieval of context-aware applications on mobile devices: how to evaluate? In IIiX '08: Proceedings of the second international symposium on Information interaction in context, pages 65{71, New York, NY, USA, 2008. ACM.

[6] Ichiro Satoh. Experience of context-aware services in public spaces. In ICPS '08: Proceedings of the 5th international conference on Pervasive services, pages 81{90, New York, NY, USA, 2008. ACM.

[7] Guanling Chen and David Kotz. Context aggregation and dissemination in ubiquitous computing systems. In WMCSA '02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, page 105, Washington, DC, USA, 2002. IEEE Computer Society.

[8] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, Andrew T. Campbell. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, ACM, New York (2010)

[9] Anusuriya Devaraju, Simon Hoh, and Michael Hartley. A context gathering framework for context-aware mobile solutions. In Mobility '07: Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology, pages 39{46. ACM, 2007.

[10] Weichang Du and Lei Wang. Context-aware application programming

for mobile devices. In C3S2E '08: Proceedings of the 2008 C3S2E conference. ACM, 2008.

[11] Nathan Eagle, Alex (Sandy) Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. Proc Natl Acad Sci (PNAS), USA 2009, Sep 8.

[12] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell, Dartmouth College. A Survey of Mobile Phone Sensing. Communications Magazine, IEEE Sept. 2010.

[13] Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson. Rapid prototyping of mobile context-aware applications: the cyberguide case study. In MobiCom '96: Proceedings of the 2nd annual international conference on Mobile computing and networking, pages 97{107, New York, NY, USA, 1996. ACM.

[14] Tejas Lagvankar, Collaborative data gathering using context-aware mobile Devices, Master’s thesis, University of Maryland Baltimore County, 2010

[15] Laura Zavala, Pramod Jagtap, Radhika Dharurkar, Anupam Joshi, Tim Finin. Privacy Preservation in Mobile Context-Aware Systems. IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY 2011)

[16] Ling Bao and Stephen S. Intille. Activity Recognition from User-Annotated Acceleration Data. 2004.