

January 2013

Energy Efficient Context-Aware Framework in Mobile Sensing

Ozgur Yurur

University of South Florida, oyurur@mail.usf.edu

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>

 Part of the [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

Scholar Commons Citation

Yurur, Ozgur, "Energy Efficient Context-Aware Framework in Mobile Sensing" (2013). *Graduate Theses and Dissertations*.
<http://scholarcommons.usf.edu/etd/4797>

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

Energy Efficient Context-Aware Framework in Mobile Sensing

by

Özgür Yürür

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Wilfrido Moreno, Ph.D.
Paris Wiley, Ph.D.
Andrew Rajj, Ph.D.
Miguel Labrador, Ph.D.
Fernando Falquez, Ph.D.

Date of Approval:
May 31, 2013

Keywords: Ubiquitous Sensing, Cyber-Physical Systems, Context-Awareness, Human Activity
Recognition, Machine Learning, Learning from Data, Optimal Sensing, Power Efficiency

Copyright © 2013, Özgür Yürür

DEDICATION

To my parents Lütfiye Yürür and Zekeriya Yürür,
and my elder brother Murat Yürür

ACKNOWLEDGMENTS

First of all, I am truly indebted to my supervisor, Dr. Wilfrido Moreno, for his guidance, support, and constant encouragement throughout my Ph.D. studies. I am grateful to Dr. Paris Wiley, Dr. Andrew Raij, Dr. Miguel Labrador and Dr. Fernando Falquez for serving in my committee, and for their precious time and invaluable suggestions. I also want to thank Dr. Carlos Smith for chairing my defense. In addition, I would like to acknowledge Dr. Hüseyin Arslan for his help in my earlier Ph.D studies, and the staff from the Electrical Engineering Department at USF, who has always been very helpful, supportive and kind.

I owe much to my friends and colleagues Ali Görçin, Dr. Ismail Bütün, Gülnur Efe, and Emre Seyyal for their enlightening discussions and support. I am also thankful to Stephanie Aviles and her family, and the Turkish community in Tampa, FL for their kindness and sincere help.

My deepest appreciation goes to my parents and my elder brother for bringing me up, leading me to the right direction, and always encouraging me for pursuing higher degrees in education. It is not possible to thank them enough, but I want them to know that I will be grateful to them throughout my life. I wish my grandparents were able to see that moment as well, which they had been dreaming of. May God rest them in peace. I also want to express my gratitude to all members of Yürür, Kurtulmuş, and Üsküp families.

In the end, the ultimate reality is that this would not be possible without the Will of God. I pray to God to pave our way to ultimate knowledge and help us use it for the real benefit of mankind.

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	v
ABSTRACT	vii
CHAPTER 1 INTRODUCTION	1
1.1 Motivation	3
1.1.1 More Focus on the Importance of Mobile Sensing in the Area of Health Community	4
1.2 Challenges	6
1.3 Dissertation Outline	8
1.3.1 Chapter 2: Context-Awareness: A Survey	9
1.3.2 Chapter 3: Context Inference: Light-Weight Online Unsupervised Posture Detection by Smartphone Accelerometer	10
1.3.3 Chapter 4: Context-Aware Framework: A Basic Design	10
1.3.4 Chapter 5: Analytical Modeling of Smartphone Battery and Sensors, and Energy Consumption Profiles	11
1.3.5 Chapter 6: Context-Aware Framework: A Complex Design	11
CHAPTER 2 CONTEXT-AWARENESS: A SURVEY	13
2.1 Contextual Information	13
2.2 Context Representation	14
2.3 Context Modeling	15
2.4 Context Inference	16
2.5 Context-Aware Middleware Design	19
2.6 Context-Aware Applications	23
2.6.1 Studies for Creating ‘A Generic Framework Design’	24
2.6.2 Studies for ‘Human Activity Recognition’	26
2.6.2.1 Smartphone Based	27
2.6.2.2 Wearable Sensors Based	27
2.6.3 Studies for ‘User Tracking: Transportation Modes and Location Info’	28
2.6.4 Studies for ‘Social Networking’	30
2.6.5 Studies for ‘Healthcare and Well-being’	31
2.6.6 Studies for ‘Monitoring Environment’	32
2.6.7 Studies for ‘Auxiliary Toolkits for Context Inference’	32
CHAPTER 3 CONTEXT INFERENCE: LIGHT-WEIGHT ONLINE UNSUPERVISED POSTURE DETECTION BY SMARTPHONE ACCELEROMETER	34
3.1 Standalone Mode: The Proposed Novel Classification Method	38
3.2 Assisting Mode: Enhancing Existing Classification Methods for On-line Processing	43

3.2.1	Feature Extraction	44
3.2.2	Pattern Recognition Based Classification	44
3.2.2.1	Gaussian Mixture Model	45
3.2.2.2	k -Nearest Neighbors Search (k -NN)	47
3.2.2.3	Linear Discriminant Analysis (LDA)	47
3.2.2.4	Online Processing - Dynamic Training/Supervisory Classes Update	49
3.2.3	Hidden Markov Models (HMMs)	50
3.3	Performance Evaluation	51
3.4	Additional Topic: Context Monitoring Mechanism	55
CHAPTER 4 CONTEXT-AWARE FRAMEWORK: A BASIC DESIGN		58
4.1	Prior Works and Discussions	60
4.2	Construction of the Proposed Framework	62
4.2.1	Basic Definitions	64
4.2.2	User State Representation Engine	65
4.2.3	System Adaptability	70
4.2.3.1	Time-Variant User State Transition Matrix	70
4.2.3.2	Time-Variant Emission Matrix	71
4.2.3.3	System Parameters Updating	71
4.2.3.4	Entropy Rate	72
4.2.3.5	Scaling	72
4.3	Simulations	73
4.3.1	Preparations	73
4.3.2	Process	74
4.3.3	Power Consumption Model	75
4.3.4	Accuracy Model	77
4.3.5	Setups	78
4.3.6	Discussions	78
4.4	The Concept Validation Through a Smartphone Application	81
4.4.1	Observation Analysis	81
4.4.1.1	Construction of Observation Emission Matrix	83
4.4.2	Process	84
4.4.3	Discussions	84
CHAPTER 5 ANALYTICAL MODELING OF SMARTPHONE BATTERY AND SENSORS, AND ENERGY CONSUMPTION PROFILES		89
5.1	Battery Modeling	90
5.2	The Modeling of Energy Consumption in Sensors	97
5.2.1	Preliminaries	99
5.2.2	The Modeling of Sensory Operation	100
5.3	A Case Study: A Real-Time Application by The Smartphone Ac- celerometer Sensor	101
5.4	Sensor Management	104
5.4.1	Discrete-Time Markov Reward Model	104
5.4.1.1	Battery Case	105
5.4.1.2	Sensor Utilization Case	107

CHAPTER 6	CONTEXT-AWARE FRAMEWORK: A COMPLEX DESIGN	108
6.1	Prior Works	112
6.2	The Context Inference Module	113
6.2.1	Inhomogeneous Hidden Semi-Markov Model: A Statistical Machine	114
6.2.1.1	Basic Definitions and Inhomogeneity	115
6.2.1.2	The Working Process	115
6.2.1.3	User State Representation Engine	117
6.2.1.4	Time-Variant User State Transition Matrix	120
6.2.1.5	Observation Emission Matrix	122
6.2.2	The Output of the Context Inference Framework	122
6.3	Sensor Management System	124
6.3.1	Sensor Utilization	124
6.3.2	The Trade-off Analysis: The Description of Action Set	127
6.3.3	Intuitive Solutions	127
6.3.3.1	Method I	128
6.3.3.2	Method II	128
6.3.3.3	Method III	128
6.3.4	Constrained Markov Decision Process (CMDP)	129
6.3.5	Partially Observable Markov Decision Process (POMDP)	131
6.3.5.1	Myopic Strategy and Sufficient Statistics	133
6.4	Performance Analysis	134
CHAPTER 7	CONCLUSION AND FUTURE WORK	139
7.1	The List of Contributions	139
7.2	Research Highlights	143
7.3	Future Works	144
REFERENCES		145
APPENDICES		157
	Appendix A : Some Derivations for CMDP	158
	Appendix B : Copyright Permission for Chapter 4	159
ABOUT THE AUTHOR		End Page

LIST OF TABLES

Table 2.1	Feature selections	19
Table 2.2	Classification algorithms	19
Table 2.3	HAR in mobile devices	27
Table 3.1	Summary of important symbols in chapter 3	38
Table 3.2	Confusion Matrix 1: user state recognition under different classification methods at 100 Hz sampling	53
Table 3.3	Confusion Matrix 2: DT under different sampling frequencies	54
Table 4.1	Filtering user states while no observation received	69
Table 4.2	Current consumption vs. data rate in accelerometer, ADXL346	75
Table 5.1	The power consumption ratio in the sensor drain per each operation cycle: $t_c = 2s$, and the comparison applied based on (50%, 12.5 Hz)	101
Table 6.1	Summary of important symbols in chapter 6	112

LIST OF FIGURES

Figure 1.1	Remote health monitoring	5
Figure 1.2	Challenges in mobile and pervasive sensing	7
Figure 1.3	Dissertation outline	9
Figure 2.1	Context representation	15
Figure 2.2	Meta-data representation	16
Figure 2.3	The stages of context inference	17
Figure 2.4	OSI reference model	20
Figure 2.5	Context-aware middleware	21
Figure 2.6	Context-aware middleware platform	22
Figure 2.7	Context-aware computing	24
Figure 3.1	The proposed system structure for user state classification: standalone or assisting modes	37
Figure 3.2	The proposed decision tree based classification method: standalone mode	39
Figure 3.3	Euclidean distance analysis: user state 'sitting' is the reference point	40
Figure 3.4	The context inference from the accelerometer sensor: (a) a ten-minute recording of three-axial acceleration signals while user posture changes, (b) the corresponding user state representations before smoothing is applied.	51
Figure 3.5	Context monitoring mechanism	55
Figure 4.1	Operation of the proposed framework	63
Figure 4.2	User state recognition/estimation method	65
Figure 4.3	Forward-backward algorithm	66
Figure 4.4	Viterbi algorithm	67
Figure 4.5	Prediction/filtering/smoothing	68
Figure 4.6	An example of user state transitions	70

Figure 4.7	Simulation: entropy rate vs. variant user profiles	79
Figure 4.8	Simulation: power consumption vs. accuracy	80
Figure 4.9	Three-axial accelerometer signals: the two-user-state case	82
Figure 4.10	Experiment: entropy rate analysis	83
Figure 4.11	Experiment: power consumption vs. accuracy	84
Figure 4.12	Experiment: battery depletion	85
Figure 5.1	A battery cell	91
Figure 5.2	Recovery effect (Courtesy of [1–4])	92
Figure 5.3	Rate capacity effect	92
Figure 5.4	The two-well KiBaM	94
Figure 5.5	The KiBaM discharge model: an example	96
Figure 5.6	Sensory operations	98
Figure 5.7	Duty cycling and sensor sampling	99
Figure 5.8	Interrupt Poisson Process (IPP)	100
Figure 5.9	The battery depletion due to the accelerometer sensor	103
Figure 6.1	The operational work-flow of the proposed framework	110
Figure 6.2	The properties of context-aware framework in mobile computing	111
Figure 6.3	Semi-Markovian feature by sensor samplings	117
Figure 6.4	User state representation engine: recognition and estimation models	118
Figure 6.5	Power consumption rate according to variant sensory operation methods	126
Figure 6.6	Power consumption rate in response to user profile	134
Figure 6.7	Recognition accuracy rate in response to user profile	135

ABSTRACT

The ever-increasing technological advances in embedded systems engineering, together with the proliferation of small-size sensor design and deployment, have enabled mobile devices (e.g., smartphones) to recognize daily occurring human based actions, activities and interactions. Therefore, inferring a vast variety of mobile device user based activities from a very diverse context obtained by a series of sensory observations has drawn much interest in the research area of ubiquitous sensing. The existence and awareness of the context provides the capability of being conscious of physical environments or situations around mobile device users, and this allows network services to respond proactively and intelligently based on such awareness. Hence, with the evolution of smartphones, software developers are empowered to create context aware applications for recognizing human-centric or community based innovative social and cognitive activities in any situation and from anywhere. This leads to the exciting vision of forming a society of “Internet of Things” which facilitates applications to encourage users to collect, analyze and share local sensory knowledge in the purpose for a large scale community use by creating a smart network which is capable of making autonomous logical decisions to actuate environmental objects. More significantly, it is believed that introducing the intelligence and situational awareness into recognition process of human-centric event patterns could give a better understanding of human behaviors, and it also could give a chance for proactively assisting individuals in order to enhance the quality of lives.

Mobile devices supporting emerging computationally pervasive applications will constitute a significant part of future mobile technologies by providing highly proactive services requiring continuous monitoring of user related contexts. However, the middleware services provided in mobile devices have limited resources in terms of power, memory and bandwidth as compared to the capabilities of PCs and servers. Above all, power concerns are major restrictions standing up to implementation of context-aware applications. These requirements unfortunately shorten device battery lifetimes due to high energy consumption caused by both sensor and processor operations. Specifically, continuously capturing user context through sensors imposes heavy workloads in hardware

and computations, and hence drains the battery power rapidly. Therefore, mobile device batteries do not last a long time while operating sensor(s) constantly.

In addition to that, the growing deployment of sensor technologies in mobile devices and innumerable software applications utilizing sensors have led to the creation of a layered system architecture (i.e., context aware middleware) so that the desired architecture can not only offer a wide range of user-specific services, but also respond effectively towards diversity in sensor utilizations, large sensory data acquisitions, ever-increasing application requirements, pervasive context processing software libraries, mobile device based constraints and so on. Due to the ubiquity of these computing devices in a dynamic environment where the sensor network topologies actively change, it yields applications to behave opportunistically and adaptively without a priori assumptions in response to the availability of diverse resources in the physical world as well as in response to scalability, modularity, extensibility and interoperability among heterogeneous physical hardware.

In this sense, this dissertation aims at proposing novel solutions to enhance the existing tradeoffs in mobile sensing between accuracy and power consumption while context is being inferred under the intrinsic constraints of mobile devices and around the emerging concepts in context-aware middleware framework.

CHAPTER 1

INTRODUCTION

Growing sensor deployment and computing technologies in mobile devices have enabled researchers to pervasively recognize the individual and social context that device users encounter with. Hence, the inference of daily occurring human-centric actions, activities and interactions by a set of mobile device based sensors has drawn much interest in the research area of ubiquitous sensing community¹. A human behavior is highly dependent on perception, context, environment and prior knowledge of most recent event patterns. The understanding of human activity is based on the discovery of the activity pattern and accurate recognition of the activity itself. Therefore, researchers have focused on implementing computational pervasive systems in order to create high-level conceptual models to infer activities, and low-level sensory models to extract context from unknown activity patterns. However, the creation of a generic model to represent a true nature of human behavior in this process stands as a big challenge. In this regard, the construction of a framework within the realm of middleware technologies belonging to the context aware sensing systems has been put forward to provide a required model for recognition of daily occurring human activities via observations acquired by various sensors built in mobile devices. These activities are inferred as outcomes of a wide range of sensory applications utilized in such diverse implementation areas ranging from environmental surveillance, assisting technologies for medical diagnosis/treatments to the creation of smart spaces for individual behavior modeling. Key challenges that are faced in this concept are to infer relevant activity in a system that takes raw sensor readings initially and processes them until obtaining a semantic outcome under some constrictions. These constrictions mostly stem from the difficulty of shaping exact topological structure and modeling uncertainties in the observed data due to both minimizing energy consumed by physical sensor operations and analyzing sensory data that is in process [5, 6].

¹Paradigm of ubiquitous sensing is also described as pervasive computing, mobile computing, context-aware sensing, ambient intelligence or more recently, everyware.

Today's mobile devices have been becoming increasingly sophisticated and the latest versions of them are now equipped with a rich set of powerful small size built-in sensors such as accelerometers, ambient light sensors, GPSs, magnetic compasses, Wi-Fi, etc. These sensors can directly or indirectly measure various information belonging to the physical world surrounding the mobile device; thereby, the ubiquitous use of the mobile devices in the society creates a new exciting research area for context-aware sensory data mining applications. Specifically, smartphones could provide a large number of applications within the defined research area. Since human beings are involved in a vast variety of activities within a very diverse context, and the usage of mobile phones are getting more integrated into human lives, a specific context acquired through built-in sensors can be extracted by a smartphone application. Then, a desired information within the context can be inferred by successful computing implementations. These applications can be classified under two different categories: personal/human-centric and participatory/community/opportunistic sensing. In personal sensing applications, the device user is the point of interest. For instance, the monitoring and recognition of the user related posture and movement patterns for a personal fitness log or for health care reasons is an active research topic in this field. On the other hand, participatory sensing relies on the multiple deployment of mobile devices to interactively and intentionally (e.g., also autonomously in opportunistic sensing) share, gather and analyze local knowledge which is not solely based on human activity, but also based on the surrounding environment. Hence, participatory sensing requires a collection of sensory data obtained through multiple user participation in order to result in a large-scale phenomenon, which cannot be easily measured by a single user participation. For example, delivering an intelligent traffic congestion report while drivers are providing their speeds and exact locations. In summary, the generic idea of all possible sensing applications is to orchestrate the increasing capabilities of the mobile devices (e.g., computing, communication and networking, and sensing) through running software on an existing hardware platform at a right time and place in order to enable services to infer meaningful information for the benefit of individual and community use.

However, the middleware services provided in mobile devices have limited resources in terms of power, memory and bandwidth as compared to the capabilities of PCs and servers. Above all, power concerns are major restrictions standing up to implementation of context-aware applications. This requirement unfortunately shortens device battery lifetimes due to high energy consumption caused by both sensor and processor operations. Specifically, continuously capturing user context

through sensors imposes heavy workloads in hardware and computations, and hence the battery power rapidly depletes. Therefore, mobile device battery lifetimes are reduced while operating sensor(s) constantly. One solution is to take precautions on sensory operations while putting them into sleeping mode to reduce power consumption. However, this precaution turns into an accuracy problem that the middleware services may produce while providing information to the applications. This precaution also triggers another important topic that researchers have been studying, which is based on finding optimal solutions to balance the trade-off existing between power delivered by the mobile device battery and accuracy in operation of applications. Hence, the key goal lies in discovering the best characteristics of the target complex spatial phenomenon being sensed, meeting the demands of applications, and satisfying the constraints on sensor usage.

1.1 Motivation

The evolution of mobile devices outfitted with powerful sensors leads to progress in the advancement process of the Internet of Things [7, 8]. The integration of sensing and advance computing capability of these sophisticated devices produce sensory data and exchange information among local or system-wide resources by feeding the Internet at a social scale within the concept of personal or participatory sensing. This situation will direct the concept of the Internet of Things to shift into a collection of autonomous, ambient intelligent and self-operated network nodes (e.g., independently acting smartphones) which are well aware of surrounding context, circumstances and environments. With these capabilities, the new network architecture would enhance data credibility, quality, privacy and share-ability by encouraging participation at personal, social and urban scales. It also would lead to the discovery of knowledge about human lives and behaviors, and environmental interactions/social connections by leveraging the deployment capacity of smart things (e.g., smartphones, tablets) in order to collect and analyze the digital traces left by users.

Most sensors currently available on mobile devices are designed to perform some specific applications. For example, accelerometers for detecting screen orientation, a microphone for voice conversations, a camera for capturing images and a GPS for displaying location. However, by introducing intelligence, situational awareness and context recognition into these devices, and by giving the right architecture within the context of ubiquitous sensing in order for enhancing and systematizing the existing methodologies, built-in sensors could be re-purposed and could act as proactive sensor nodes. Thus, smartphones could be used as instruments to collect data and provide

meaningful observations about user behaviors and surrounded environment such as measurement of activity by accelerometer, ambient sound environment by microphone and estimation of time and location user spends indoors and outdoors by GPS. In addition, external sensors, such as biomedical sensors (e.g., ECG, BVP, GSR, and EMG), can also be deployed with a wearable strap on human bodies. Hence, more than one sensor (multiple sensory systems) would be available in ubiquitous sensing for health. Information obtained from different sensors can be cross-linked and presented as a new valuable input. For instance, GPS and accelerometer actualizes Geographic Information Systems (GIS) by potentially providing insight as to how the proximity of recreational facilities affects physical activity levels, or how the relative accessibility of grocery stores and fast-food restaurants influence a diet program. Wi-Fi can be leveraged to determine relative proximity of individuals to each other or fixed locations. Bluetooth, as well as ZigBee, has been used for ambulatory data collection of more traditional signals, such as blood pressure, heart rate, respiration, and blood glucose level for monitoring community health.

1.1.1 More Focus on the Importance of Mobile Sensing in the Area of Health Community

The usage of mobile devices within the context of the ubiquitous sensing is not a new research area. Previously, this technology had been successfully integrated in zoology and veterinary medicine to study the feeding habits and social behaviors of some types of animals from zebras to whales; whereas the adaptation of this technology to human health has only recently begun. With the tremendous development of smartphones and implementation of relevant context-aware applications, it becomes possible to acquire insights about benefits of ubiquitous sensing in the health industry, see Figure 1.1, especially in terms of its role in the social and physical environment.

The goal of community health programs is to improve the overall quality of life by promoting cognitive, physical, and social/emotional well-being [9, 10]. Continuous observation of routine behaviors reflecting physical and physiological health situations can be the source of prediction of future health problems. The conventional model for collecting behavioral data in the health sciences relies on collected data in laboratory settings and/or through periodic surveys/reports. However, this model has several drawbacks as follows:

- excessive time and resource requirements to gather simultaneous data from individuals

- occasionally measured, randomly taken, manually obtained behavioral data sometimes fails to present real and finer details in health states
- too much effort to be suitable for constantly running long-term monitoring



Figure 1.1: Remote health monitoring

On the other hand, automatic sensing of the physical health of individuals based on mobile devices is an active research area. Progress in mobile sensing in terms of recognizing the social and cognitive well-being of humans will bring so many benefits to clinicians, patients, and researchers. Mobile sensing systems infer very detailed measurement information of people's social and behavioral attitudes in their environments over extended time periods. The new health care system will definitely promote some improvements such as:

- monitoring of health status and well-being of individuals. In case of emergency, quick help can be obtained from a primary care and medical home team.
- a detailed analysis of how individuals interact with each other, which can lead to better understanding of behavioral factors that influence their social and cognitive well-being. As a result, clinicians can select appropriate interventions.
- increase in early diagnosis, behavioral interventions, and self-monitoring to improve social and cognitive well-being through automatic tracking and detailed analysis of behavior.

Generally speaking, mobile sensing would be useful for quantifying social wellness from the behavioral indicators, and also for better understanding of how a mobile technology can lead to advancement in health assessment and interventions.

1.2 Challenges

Mobile devices supporting emerging computationally pervasive applications will constitute a significant part of future mobile technologies by providing highly proactive services requiring continuous monitoring of user related contexts. However, a major challenge standing up to sensor-rich devices is resource-limitation, see Figure 1.2. Specifically, continuously capturing user context through sensors imposes heavy workloads both physically and computationally during the operation of mobile devices, thereby drains the battery power rapidly.

To better understand this issue, an application example in [11] can be examined. Accordingly, the accelerometer sensor built-in HTC Touch Pro is employed at a fixed sampling frequency. When the phone receives data samples from the accelerometer, the overall power consumption on device increases by 370 mW; whereas, the accelerometer, Kionix KXSD9, should consume less than 1 mW while being active according to the data sheet. Even if the accelerometer itself wastes very little power to operate its functionality, the phone, together with its main processor and other hardware components, causes much more power consumption while being in process to accomplish a contextual sensory data extraction. Another example can also be given by a provided study in [12]. It is reported that today's mobile devices are not feasible to employ all sensors at the same time by giving an example of Nokia N95 mobile phone with a fully-charged battery. It is experimentally examined that the phone battery would get totally depleted within six hours if GPS is switched on permanently, even it is not being actively used; whereas the same battery supports a conventional telephone conversation up to ten hours.

Besides the arising concern about increasing power consumptions, the analysis and inference process of contextual sensory data has many drawbacks as well. Many studies can be found in which a framework is proposed to capture and evaluate sensory data. Most of the studies rely on the recognition of user activities and the definition of common user behaviors. The applied methods in relevant studies are based on statistical models, predefined feature extractions and classification algorithms. However, none of these studies engage themselves to model a common framework in order to construct a generic structure for future context-aware applications. They would rather

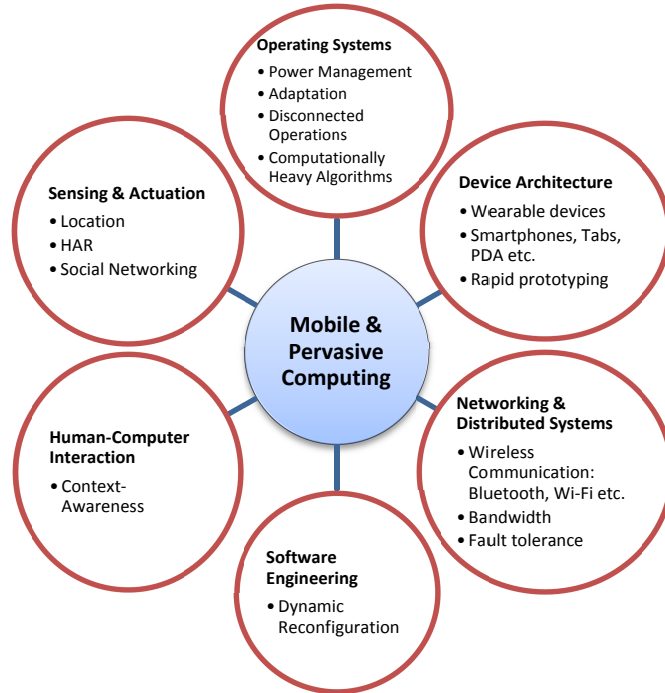


Figure 1.2: Challenges in mobile and pervasive sensing

have canalized solutions to solve their own unique problems instead of proposing a generalized approach. Therefore, these studies mostly focus on a specific sensor while looking for possible target applications in order to exploit the contextual data. A generic framework which fulfills requirements set by all types of context-aware applications was not identified. This problem often stems from the difficulty in building a reliable data set in order to represent a specific recognition interest since the obtained sensory data may vary under different circumstances (e.g., human speech with a variant background noise or placement of the mobile device). As a result, context inference algorithms would not be practical toward varying sensing conditions and eventually tend to perform poorly. Hence, the robustness becomes an important system attribute to consider. For instance, the robustness turns into a severe problem in participatory sensing applications while reasoning different inference assumptions from multiple co-located mobile devices on a specific sensing task. One solution to the robustness problem can be considered by taking advantage of cloud computing technologies which enable to share information and ensemble situational resources between co-located mobile devices by cooperating them to boost up the sensing performance, robustness, and by delivering more effective achievement on a common sensing task.

Another important system attribute that needs to be considered is preventing usage of supervised learning strategy in context inference algorithms. Most systems take pre-defined models or classifiers by creating a specific training data after several repetitions of experiments, which returns into processing large amount of data classes, thereby it forces the analysis of the data to be carried out under offline processing. Obtaining training data classes to feed statistical models, classification or machine learning algorithms in a supervised learning strategy is an expensive real-time operation for mobile devices because it is impractical in terms of computational manner while acquiring and analyzing sensory data, managing resources by storing training samples, handling scalability problem by labeling data, and regulating bandwidth problem by transferring large information. Therefore, the utilization of sensors must be lightweight and unobtrusive. Also, the applied classification and machine learning algorithms used to process sensory data must be trainable without requirements of computationally expensive human-intervened supervisory learning mode.

In conclusion, new generation mobile devices running context-aware applications should provide a scalable and energy efficient context monitoring architecture. The designed architecture should actively manage operations between context aware applications and computing resources, and coordinate multiple applications effectively to leverage the limited resources, and provide the quality service to users.

1.3 Dissertation Outline

The overview of the dissertation, seen in Figure 1.3, is as follows: Chapter 2 includes comprehensive information around the dissertation topic. The definition of context, and the stages to infer context together with context modeling problem are introduced in this chapter. The chapter also introduces the context aware middleware services, and their important properties such as transparency, adaptability, and reflectiveness. Finally, the chapter summarizes recently released significant context aware applications, and categorizes them under the interested context. Chapter 3 proposes a novel context inference algorithm by processing the raw sensory data. The smartphone accelerometer sensor is used for providing relevant sensory data and of course the context. Chapter 4 presents a novel context aware framework design within the middleware services. The framework aims at recognizing the user related activities, discovering the activity profile and tendency towards a change in the profile, and estimating the instant activity if a relevant sensory information does not exist. Chapter 5 investigates the battery behavior towards changing loads, and forms an analyt-

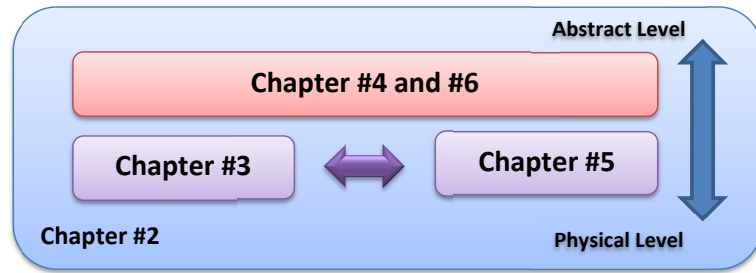


Figure 1.3: Dissertation outline

ical model to investigate variant battery discharge profiles. Having the knowledge of the battery’s nonlinear behavior, a detailed look into sensory operations is provided, and energy consumption profiles for the corresponding operations are correlated with the battery discharge. Chapter 6 presents another novel context aware framework model design, but this time the concept is designed as inhomogeneous, which considers user activity profiles are time-variant. In this chapter, the adaptability problem caused by the inhomogeneity is solved by an entropy rate analysis on user activity profiles. A sensor management system is also integrated to implement different sensory operation methods in order to achieve an efficiency in power consumption while a context-aware application is running. Finally, Chapter 7 points out future research directions and integrated topics related to enhancement of the middleware services which this dissertation proposes.

1.3.1 Chapter 2: Context-Awareness: A Survey

The existence of context allows personalization of network services and is useful for mobile device users. In addition, *awareness* of the context provides the capability of being conscious of physical environments or situations around mobile device users, and makes these services respond proactively and intelligently based upon such awareness. With the evolution of smartphones and the increased computational power within these devices, developers are empowered to create context aware applications for innovative social and cognitive activities in any situation and from any location. This leads to the exciting vision of forming a society of “smart spaces, or “Internet of Things” [7, 8]. Hence, the key idea behind context-aware sensing applications is to encourage users to collect, analyze and share numerous local sensor knowledge for large scale community use.

The latter creates a knowledge network which is capable of making autonomous logical decisions to actuate environmental objects and also to assist individuals.

1.3.2 Chapter 3: Context Inference: Light-Weight Online Unsupervised Posture Detection by Smartphone Accelerometer

This chapter proposes a light-weight online classification method to detect the user centric postural actions, such as sitting, standing, walking and running, by smartphones. These actions are named as user states since they are inferred after the analysis of data acquired through the accelerometer sensor built-in smartphones. To differentiate one user state to another, many studies can be found in the literature. However, this study differs from others by offering a computational light-weight and online classification method without knowing any *a priori* information. The proposed method not only provides a standalone solution in differentiation of user states, but also assists other widely used classification methods by generating training data classes and/or input system matrices. In addition, this chapter intends to improve these existing methods for online processing. Finally, the proposed method still makes a solid differentiation in user states even where the sensor is being operated under slower sampling frequencies².

1.3.3 Chapter 4: Context-Aware Framework: A Basic Design

This chapter presents a novel framework which is based on Hidden Markov Model (HMM) statistical model and learning from data concepts. The framework either recognizes or estimates user contextual inferences called ‘user states’ for future context-aware applications. Context-aware applications require continuous data acquisition and interpretation from one or more sensor reading(s). Therefore, device battery lifetimes need to be extended due to the fact that constantly running built-in sensors deplete device batteries rapidly. In this sense, a framework is constructed to fulfill requirements needed by applications and to prolong device battery lifetimes. The ultimate goal of this chapter is to have accurate user state representations and to maximize power efficiency. Most importantly, this research intends to create and clarify a generic framework to guide the development of future context-aware applications. Moreover, topics such as user profile adaptability and adaptive sampling are presented. The proposed framework is validated by simulations and implemented in a

²The content of this chapter is documented in parts in [13, 15]

real-time application. According to the results, the proposed framework shows an increase in power efficiency of 60% for an accuracy range from 75% up to 96%, depending on user profiles³.

1.3.4 Chapter 5: Analytical Modeling of Smartphone Battery and Sensors, and Energy Consumption Profiles

The usage of mobile devices, such as smartphones, is constrained by battery lifetimes. With the ever-increasing computing power and hardware development in the mobile devices comparing to the slow growth in the energy densities of the mobile device based battery technologies, topics such as the extension of battery lifetimes and the estimation of energy delivery by the batteries have been focused recently in the research area of mobile computing. Hence, modeling of the power consumption profiles on mobile device usage with the knowledge of the battery behavior becomes very important for energy optimization and management of resource constrained mobile computing systems. This chapter studies the battery modeling under the scope of the battery non-linearities with respect to variant battery discharge profiles. In addition, the energy consumption behaviors of some smartphone sensors are analytically modeled. Especially, a real time application is provided for the accelerometer sensor to investigate the energy consumption behavior in detail. Energy consumption profiles are created by assigning different pairs of duty cycles and sampling frequencies in the sensory operations. Finally, a Markov-reward process is integrated in order to model the energy consumption profiles and represent the energy cost by each profile as an accumulated reward in the process. The accumulated reward is also linked to the battery modeling to make a connection between the usage pattern on sensors and the battery behavior. In conclusion, with the understanding of nonlinearity observed on the batteries with respect to variant operational methods in sensors, a tolerable power consumption balance is achieved while employing context aware services in resource constrained mobile devices⁴.

1.3.5 Chapter 6: Context-Aware Framework: A Complex Design

New generation mobile devices have become inevitable to be employed within the realm of ubiquitous sensing. Especially, smartphones have gained importance to be used for Human Activity Recognition (HAR) based studies since it is believed that recognizing the human-centric activity patterns accurately enough could give a better understanding of human behaviors, and also more

³The content of this chapter is documented in parts in [16], and the relevant permission is attached in B

⁴The content of this chapter is documented in parts in [17, 18]

significantly, it could give a chance for assisting individuals in order to enhance the quality of lives. However, the integration and realization of HAR based mobile services stand as a big challenge on resource-constrained mobile embedded platforms. In this manner, this chapter proposes a novel Discrete Time Inhomogeneous Hidden Semi-Markov Model (DT-IHS-MM) based generic framework to address a better realization of HAR based mobile context-awareness. In addition to that, the chapter provides power efficient sensor management strategies including three intuitive solutions, and also Constrained Markov Decision Process (CMDP) and Partially Observable Markov Decision Process (POMDP) based optimal solutions in order to respond the tradeoff defined between the accuracy in context-aware services and the power consumption caused by the service operations. In conclusion, the proposed tradeoff solutions achieve a 50% overall enhancement in the power consumption caused by the physical sensor with respect to overall 95% accuracy rate thanks to the provided adaptive context inference framework⁵.

⁵The content of this chapter is documented in parts in [19]

CHAPTER 2

CONTEXT-AWARENESS: A SURVEY

Many researchers have studied context-awareness through mobile devices within the area of ubiquitous sensing, thereby the context awareness and use of this context in benefit of individual or community scale have taken a significant role in mobile computing platforms. *Context-aware systems aim at using a mobile device (e.g., a hand-held smartphone or attached/wearable device) integrated with smart sensors to monitor and measure individual and environmental phenomena with the purpose of assisting or evaluating human lives in order to achieve a desirable quality of life.*

The existence of context makes network services able to be personalized and useful to mobile device users. In addition, *awareness* of such context provides the capability of being conscious of physical environment and situation around mobile device users, and makes these services respond proactively and intelligently based upon such awareness. With the evolution of smartphones and the increased computational power within these devices, developers are empowered to create context aware applications for innovative social and cognitive activities in any situation and from any location. Hence, the key idea behind context-aware sensing applications is to encourage users to collect, analyze and share numerous local sensor knowledge for the purpose of large scale community use by creating a knowledge network that is capable of making autonomous logical decisions to actuate environmental objects and assist individuals.

2.1 Contextual Information

The ubiquity of mobile devices and proliferation of wireless networks will allow everyone permanent access to the Internet at all times and all places. With the development and deployment of new sensor technologies into mobile devices, these devices gain environmental intelligence, thereby providing the capability to sense, reason and actuate the physical world. In the real world, being aware of context and communicating it is a key part of human interaction. A context is defined as *a data source which can be sensed and used to characterize the situation of an entity.* In other words,

the context describes a physical phenomenon in a real world environment. Hence, the context can be described in a different way according to how the sensor is being used. The context can also be defined as a characterization of a specific entity situation such as user profile, user surrounding, user social interaction, user activity etc. For instance, let us define the entity by user and the context by location information, in this sense, context becomes a much richer and more powerful concept, particularly for mobile users in order to make sensor network services much more personalized, and more useful. Therefore, context awareness refers to the capability of an application being aware of its physical environment or situation and responding proactively and intelligently based on such awareness [20].

2.2 Context Representation

The property of context-awareness can be applied to mobile device based applications and systems in order to reduce human intervention by enabling automatic proactive assistant services. Many context aware applications provide this assistance by using solely logical context which is obtained via data mining techniques (e.g., stored information in profiles, databases or social websites). However, with the proliferation of sensor technologies, external physical factors (e.g., temperature, light, location etc.) are added into context aware systems.

A sensor, in context aware applications, is described not only as a physical device, but also as a data source which could be used for context representation. The collected contextual information may range in a wide sense in terms of specification and representation of a phenomenon in real world onto an entity in the cyber world, see Figure 2.1. Hence, the sensors can be classified under the following categories:

- *Physical sensor*: a sensor which can capture almost any data belonging to the physical world (e.g., GPS: location, accelerometer: activity etc.).
- *Virtual sensor*: a source of information from software applications and/or services and an expression of a semantic data obtained through cognitive inference (e.g., location info by manually entered place pinpoint through social network services or computation power of devices etc.).
- *Logical sensor*: a combination of physical and virtual sensors with additional information obtained through various sources by user interactions (e.g., databases, log files etc.).

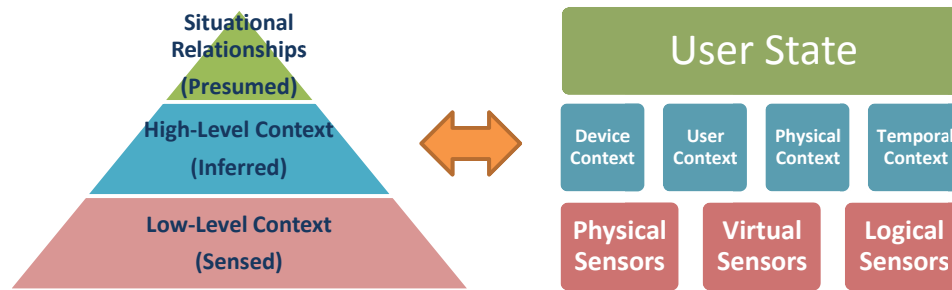


Figure 2.1: Context representation

Sensors are accepted as low-level context which is directly referred to raw data. According to levels of abstractions, high-level context is inferred from low-level context(s), which is called context interpreting. Hence, the definition of semantic meta-sensor/meta-data/meta-context implies a level of abstraction [21, 22] rising from the low-level context, also called context providing, see Figure 2.2. Unlike the sensors, the context can be divided into the following categories:

- *Device context*: e.g. net connectivity, communication cost and resources.
- *User context*: e.g. profile, geographic position, neighbors, and social situation.
- *Physical context*: e.g. temperature, noise level, light intensity, traffic conditions.
- *Temporal context*: e.g. day, week, month, season, year.

2.3 Context Modeling

Being associated with the variant context sources, accurate representation of context with a high certainty under different conditions of measuring range and sampling methods, is very important in assuring the quality of contextual information. In this sense, for the context aware applications, the definition of a desirable context is important. A clarified and consistent definition of context is a key property to ensure applications adapt to heterogeneous environment which surrounds users and inform users of such variant awareness in the defined context. The context acquisition then comes after by gathering raw sensory data through sensors. Finally, the context modeling is required to reason and interpret dynamic context representations accurately enough at a high level abstraction in an unobtrusive way. A good context modeling aims at reducing the complexity of applications for robustness and usability as well as improving their adaptability and maintainability for future

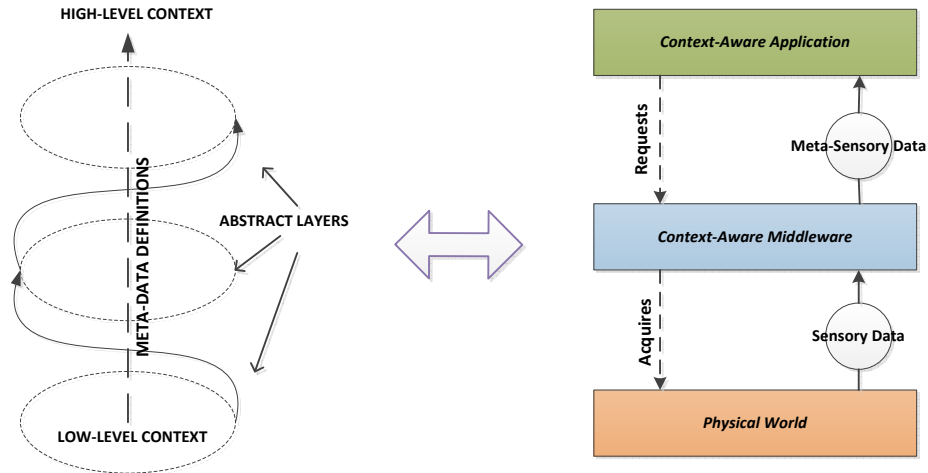


Figure 2.2: Meta-data representation

evolution. In order to accomplish that, it has to consider heterogeneity (i.e., imperfect dynamic nature) and mobility (i.e., asynchronous, timeless data capture) of a large variety of context sources at any level of abstraction. It also considers relationships and dependencies amongst semantic entities such as accuracy in context provisioning versus remaining battery power. Some context models have been proposed, such as Key-Value pair Models, Scheme Models, Graphical Models, Object Oriented Models, Logic Based Models, and Ontology Based Models. Most models create meta-data in form of profiles to represent characteristics, capabilities and requirements of context resources provided by users, devices, and service components. Another system attribute provided by these models while context modeling is to apply policies in order to express behaviors and actions among profiles [23–25].

2.4 Context Inference

One of the most important properties of context aware services is to make context inferences through sensory data. The inference functionalities, such as sensory data acquisition and processing, may show differences in terms of what context modeling or reasoning is applied [26]. However, the context is commonly referred to as low-level (i.e., atomic) context since all required operations are carried out directly from the data obtained through the physical sensors. In addition, the context is referred to as high-level context (meta-context/data), obtained by means of abstractions through the combination of low-level and/or high-level contexts, is called ‘composition’.



Figure 2.3: The stages of context inference

Figure 2.3 shows the stages of context inference process:

- Sensory readings are collected by a sliding window with a specific time length and an overlap value. The length of windowing is an important design merit. Shorter windowing cannot seize the context properly, whereas a wider windowing would create a latency in detections and put additional workload in computations. Hence, the obtained data segments by optimal windowing would provide more relevant information for context classification. In addition, the overlap value is also important in detecting any change in the context.
- Preprocessing (e.g., context filtering or sensory data fusion) could be applied if the raw sensory data is too coarse grained. It may also offer necessary modifications to correct deficiencies in the data due to the limitations of sensory operations (e.g., power concern).
- It is difficult to analyze, to build a classification model, and to infer any context from raw sensory data since it may consist of a large number of variant attributes, irrelevant information and additive noise distortion. Therefore, feature extraction is applied to exploit hidden information inside the sensory data set and to remove a direct effect of the additive noise distortion. It also enables separability in the context classification algorithm while extracting and analyzing the spatial characteristics of sensory data in each sliding window frame, and assisting in identification of different context classes. Also, a feature vector, which constitutes a representation of statistical characteristics in the contextual data, is constructed by using diverse signal processing primitives ranging from time space based features such as mean, standard deviation, correlation etc. to frequency spectrum based features such as entropy, FFT coefficients, power density etc., and also wavelet transforms, see Table 2.1.

- Constructing the feature vector and allowing it to be used by classification algorithms require high computational complexity. There is no necessity to repeatedly and redundantly compute some features, which are not helpful while inferring the context. Hence, the dimension of the feature vector needs to be reduced until it has sufficient statistics, in order to both model the context and allow lower complexity in the computations.
- The diverse characteristics of feature vectors enable us to create training contextual data classes (i.e., structural features) for classification algorithms in order to represent different context entities. Thereby, a training data class is employed by classifiers (e.g., pattern recognition/machine learning based classifiers) to build a classification model, which will allow for an unknown feature vector to test for its membership to any class dependency. The classification process essentially takes one of classifiers to map a feature vector into a training contextual data class. This process is called supervised classification. Various classification algorithms can be used as a classifier to implement a context recognition systems. From Naive Bayesian approaches and Decision Trees to pattern recognition techniques such as Gaussian Mixture Model, k-means, k-Nearest Neighbors (k-NN) search, Support Vector Machines (SVMs), many classification algorithms are involved in this process in order to cluster data classes, and then to infer relevant contexts. In addition, a statistical tool based classification such as Hidden Markov Models (HMMs) or AutoRegressive (AR) Models is widely applied. Furthermore, pattern recognition toolkits such as WEKA provides powerful solutions to the context clustering problem, see Table 2.2 for widely-used classification algorithms. However, since the supervised classifiers need extensive computations to generate models for training contextual data classes and to test for unknown pattern with the trained models, the definition of self- or co-learning based semi- or un-supervised classifiers to actualize proactive context inferences without knowing a prior data class have been actively researched recently [27].
- Lastly, the output of classifiers may not always resolve consistent discrimination in a time sequence of adjacent context inferences. In such cases, a basic smoothing technique which takes a majority voting scheme with a sliding window of a specific history length of context inferences. Hence, any inconsistency (i.e., false truthfulness) can be eliminated.

Table 2.1: Feature selections

Feature Space	Features
Time Domain	mean, standard deviation, variance, magnitude, derivative, min-max, amplitude, histogram, interquartile range (IQR), mean absolute deviation (MAD), correlation between axes, peak counting, rms, sign, and kurtosis, zero-crossing rate
Frequency domain	Fourier Transform (FT), Discrete Cosine Transform (DCT), entropy, centroid, maximum frequency, FFT energy, FFT mean and FFT standard deviation
Others	AutoRegressive coefficients, wavelet transforms

Table 2.2: Classification algorithms

Classification Algorithm	References
Decision Tree (DT), Bayesian Network (BN), Naive Bayes (NB)	[6, 12, 28–32]
Multilayer Neural Networks (MNN)/ Meta Classifier Fusion	[33–37]
Fuzzy Logic	[34, 38, 39]
Gaussian Mixture Model (GMM)	[40]
k -means/ k -Nearest Neighbor (k -NN)	[41, 42]
Hidden Markov Models (HMM)	[29, 36, 43, 44]
AutoRegressive Models (AR-M)	[45, 46]
Support Vector Machine (SVM)	[32, 45, 47, 48]
Principal Component Analysis (PCA)	[34, 48]
Linear/Quadratic Discriminant Analysis (LDA/QDA)	[34, 35, 42]
HAAR Wavelet Models	[49]
Classifier ensembles: Boosting and Bagging	[50]
Toolkits: WEKA	[49, 51–53]

2.5 Context-Aware Middleware Design

The growing deployment of sensor technologies in mobile devices and innumerable software applications utilizing sensors to offer a wide range of user-specific services have lead to create a layered system architecture so that the desired architecture can respond effectively towards diversity in sensor utilization, large sensory data acquisitions, ever-increasing application requirements, context-processing software libraries, mobile device based constraints and so on. The ubiquity of these computing devices in a dynamic environment where the sensor network topologies actively change, yields applications to behave opportunistically and adaptively without a priori assump-



Figure 2.4: OSI reference model

tions in response to the availability of diverse resources in the physical world, and also scalability, modularity, extensibility and interoperability among heterogeneous physical hardware [54, 55].

A conventional middleware takes place of the Session and Presentation Layers, see Figure 2.4, by providing a higher level of abstraction built over the network operating systems, by offering fault tolerant resource sharing, and by masking out the problems to facilitate heterogeneity, stability and efficiency of distributed systems within the ISO/OSI Reference Model. On the other hand, the context-aware middleware, see Figure 2.5, defines an abstract layer between running the operating system and the applications in software systems. It aims at dealing with the heterogeneity of physical world through edge technology connections by adding more specialized mechanisms and services than an operating system provides. It is capable of wrapping (i.e., controlling physical devices and interacting with them to receive data), analyzing and delivering the physical world information (e.g., edge technologies such as sensor networks, embedded systems, RFID or NFC tags) to the application services in a transparent way. Transparency provides the application layer the ability to separate from the internal scope of the middleware and not to interact with the implementation of the lower layers directly; thereby, the application receives the context but does not know the source of the context. In other words, the middleware creates a shielded interface by both enhancing the level of abstraction support needed by the application and intending to hide lower layer operations between the physical layer (i.e., hardware) at the bottom and the application layer (i.e., abstract) at the top. Having a middleware allows the computational burden required for context management to shift from the application to the middleware itself by letting the application developer only deal

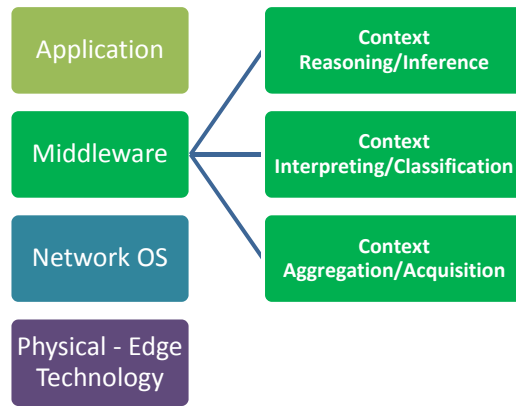


Figure 2.5: Context-aware middleware

with implementing the application logic, and by also letting the application control the entities (i.e., the characterized context) created after context management. Thus, the middleware also allows having robust optimization opportunities within the consideration of system constraints (e.g., relative computational cost associated with entity relevant operations, limited battery power, and insufficient information storage). Most importantly, assigning a middleware takes the responsibility of all context relevant entity management, and provides a complete global access into common resources needed by all applications residing on the same host without any conflict [56–58].

An entity could be anything such as an activity, person, place, object or time that application users interact with. Therefore, it has already noted that the context describes a physical phenomenon in real world environment, whereas the entity describes its projection in cyber world. Moreover, context-awareness in a system allows understanding the context information and using it with its own benefit. By integrating context awareness into the middleware technologies, it must have the capability of recognizing and interpreting the context, characterizing it with entities, allowing interactions among entities and operating the whole system under such conditions where an adaption may require. At this point, the adoption of a term called semantic metadata is defined at a high level of abstraction to represent context as the structure and meaning of entities, and also to present context relevant adaptation strategies, which allows the middleware be suitable to behave dynamically with a minimal human intervention. Having semantic metadata allows for unambiguous specification of context models and lets the knowledge to be shared among entities with no loss of meaning. Thanks to its interoperability and openness, it also allows to infer other complex

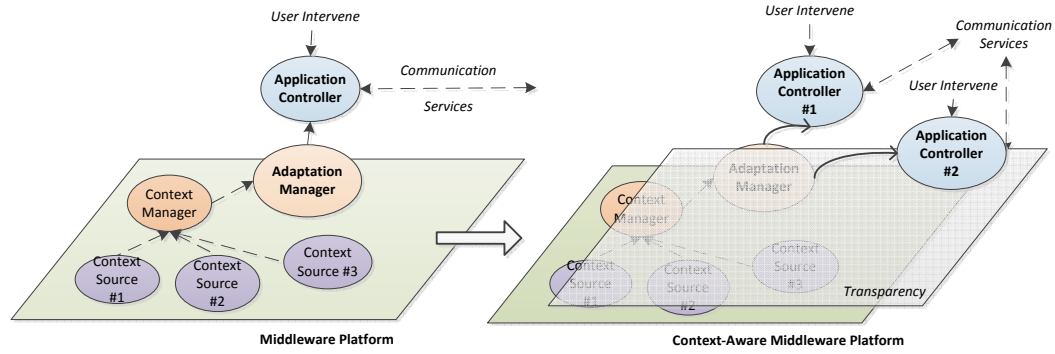


Figure 2.6: Context-aware middleware platform

knowledge at the upper layers in presence of other semantic metadata. However, the complexity of context resources in heterogeneous physical world and the interactions among the diverse context resources, make it difficult to describe the relevant metadata explicitly [59].

Other properties supported by the context-aware middleware technology, seen in Figure 2.6, can be defined as follows:

- The context-aware middleware either runs *standalone* for managing entire physical environments or accepts the existence of an infrastructure, which can deliver required services. This differentiation stems from the diversity caused by the heterogeneous sensing environment.
- The middleware has a *reflective* property which represents the obtained entities through context as semantic *meta-data*. The meta-data may belong to the application, the middleware itself, context, or interconnected (i.e., *composition*). Note that a context can differ (e.g., asynchronously obtainable over different sensor), and inter-operate with other contexts. The reflection allows the middleware to regulate its computations and monitor possible changes in the semantics. It also allows the middleware to model itself through self-representation such that the manipulation of its behavior may be changed. Hence, any change occurred at the meta-level can affect the underlying base level or vice versa.
- Adaptation is an important design merit in ubiquitous sensing which empowers users to customize systems according to individual preferences. Hence, the adaptation in ubiquitous sensing is defined by an autonomous process triggered by a set of requirements

ruled by the ubiquitous computing paradigm, in order to improve quality of service at application layer. This paradigm intends to sense the physical world, reason the obtained context, and react dynamically towards the changing context by reconfiguring its behavior accordingly. The paradigm also supports proactive adaptation, which describes the ability of envisioning future application requirements caused by the context change as well as adjusting the functionality accordingly to prevent or minimize direct application interaction with neither interfering nor modifying the application logic. In contrast, an application itself has the ability to altering its functionality with respect to context changes in reactive adaptation.

- The middleware can constitute entities from physical/virtual contexts for the sake of general usage requested by all types of applications, and it can provide specific information delivery for an application (i.e., service provider).
- The middleware must run smoothly with the underlying operating system. Since mobile applications run on resource limited devices with low memory size, slow CPU frequency, and low power supply, the light-weight middleware systems need to be considered in order to impose light computational load.

2.6 Context-Aware Applications

New generation mobile phones are now equipped with sophisticated sensors. The cameras on mobile phones can be used as video and image sensors; the microphone on the mobile phone used for voice conversations, and an acoustic sensor; and the embedded GPS receivers on the mobile phone can provide location information. Other embedded sensors such as gyroscopes, accelerometers, and proximity sensors can collectively be used to determine users physical activities. Moreover, additional sensors can be easily interfaced with the phone via wireless or wired connections to sense various context obtained for medical applications or activity recognition applications.

In this section, context-aware applications are categorized according to the application design field, see a generic Figure 2.7 for context-aware computing. The categorization begins with some examples to create a general framework design which tries to provide middleware services embracing all type of sensory operations. The rest of categorization continues by introducing different application fields that researchers have been studying extensively.



Figure 2.7: Context-aware computing

2.6.1 Studies for Creating ‘A Generic Framework Design’

The pervasive mobile computing, which captures and evaluates sensory contextual information in order to infer user relevant actions/activities/behaviors, has become a well-established research domain. Most studies rely on recognition of user activities, especially posture detection, and the definition of common user behaviors by proposing and implementing numerous context modeling systems. In addition, researchers have been aware of the need for computational power while trying to infer sensory context accurately enough. However, most works provide partial answers to the tradeoff between context accuracy and battery power consumption. It is hard to say that power saving strategies have been significantly taken at the low-level physical sensory operations. Above all, there is no a generic framework implemented in context aware applications to apply *adaptively changing* dynamic sensor management scheme. In contrast, most work emphasizes, either to set a minimum number of sensors, or to maximize power efficiency by solely applying less complexity in computations or by changing transferring methods of obtained context to the outer network services.

From the stand point of the creation of a generic framework design for middleware services in context-aware applications, it would be notable to mention the following studies:

- “EEMSS” in [12] uses a hierarchical sensor management strategy by powering a minimum number of sensors and applying appropriate sensor duty cycles so that the proposed framework could recognize user states through smartphone sensors while improving device

battery lifetimes. Unfortunately, sensors are considered to have fixed duty cycles whenever they are utilized, and also they are not adjustable to respond differently to variant user behaviors. In addition, energy consumption is reduced by shutting down unnecessary sensors at any particular time. On the other hand, classification of sensory data is based on pre-defined test classification algorithms.

- The hierarchical sensor management system is also studied by introducing “SeeMon” system in [28], which achieves energy efficiency and lessens computational complexity by only performing continuous detection of context recognitions whenever a change occurs during the context monitoring. The framework also employs a bidirectional feedback systems in computations to detect similar context recognitions in order to prevent from redundant power harvesting operations.
- Similarly, Sensay [60] is a context-aware mobile phone, but the form of an external sensor box which is mounted on the users hip area, it receives many different sensory data and eventually determines to dynamically change cell phone ring tone, alert type and un-interruptible user states. However, it classifies user state offline and the system does not have energy efficiency strategy.
- “Darwin” studied in [40] proposes a system that combines classifier evolution, model pooling, and collaborative inference for mobile sensing applications on phones. It is implemented for a speaker recognition application by using efficient but sophisticated machine learning techniques; however, there is no power consideration applied.
- “Jigsaw” presented in [61] is a continuous sensing engine for mobile phone applications, which balances the performance needed by an application and resource demands. The engine employs each sensor under a processing pipeline. It performs all the sensing and classification processing exclusively on the mobile phone, and it uses sensor-specific pipelines that have been designed to cope with the individual challenges presented by each sensor. Duty cycling techniques are attached as an adaptive pipeline process if applicable in order to conserve battery life.
- The study in [62] creates a general framework problem under an energy efficient location based sensing application. It is noted that there are four critical factors affecting energy

efficiency in location-sensing through GPS. These factors are the static use of location sensing mechanisms, the absence of use of power-efficient sensors to optimize location-sensing, the lack of sensing cooperation among multiple applications, and the unawareness of battery level. The middleware solution is given by introducing *substitution* to find an alternative, but as a less power consuming location-sensing mechanism, *suppression* uses less power consuming sensor than GPS when user location is static; *piggybacking* used to synchronize with other location-based applications to infer a collaborative location info, and *adaptation* to adjust system parameters such as time and distance longer when battery level is low.

There are also some studies proposed in [23, 36] in order to providing comprehensive solutions into creating a base framework for context-aware applications.

2.6.2 Studies for ‘Human Activity Recognition’

Recognizing human-centric activities and behaviors has been an important topic in pervasive mobile computing. Human Activity Recognition (HAR) intends to observe human related actions in order to obtain an understanding of what type of activities/routines that individuals perform within a time interval. Thereby, providing accurate information about HAR’s relevant data history could assist individuals in having a better well-being, a fitness level and a situational-awareness [63–65]. For example, patients with diabetes, obesity, or heart disease are suggested to follow a predefined fitness program as a part of their treatment [66, 67]. In this case, the interested information regarding human postures (*lying, sitting, standing, etc.*) and movements (*walking, running, etc.*) can be inferred by the HAR system in order to provide useful feedbacks to the caregiver about the patient’s behavior analysis. In addition, by the attachment of external sensor devices, e.g., *Heart Rate (HR) monitor*, patients with abnormal heart beats can be tracked easily and notified to caregivers in case of emergency in order to prevent undesirable consequences [68]. In practical, HAR has only interest on single person activity detection; however, it can be extended to be a multiple person recognition, which is called Activity of Daily Living (ADL). ADL is a way to describe the functionality status of a person and its interaction with others. Hence, ADL becomes an essential part in the area of community sensing, especially for community health-care concerns (e.g., finding stress level in a group of people [69]).

Table 2.3: HAR in mobile devices

Reference	Platform	Sensors
[6, 12, 32, 36, 37, 40–42, 47, 49]	Smartphone	ACC, BP, BT, GPS,
[29–31, 33, 51, 53]	Wearable Devices	HR, MIC, Prox.,
[28, 31, 34, 35, 44, 45, 51–53]	Mobile Development Boards or PC	Temp., Wi-Fi

ACC: Accelerometer; BT: Bluetooth; MIC: Microphone; HR: Heart Rate Monitor; Prox.: Proximity; Temp.: Temperature

Studies for HAR can be divided into sub-categorizes based on the platform a context-aware system is built on (see Table 2.3):

2.6.2.1 Smartphone Based

Activity recognition basically concerns about human beings and/or their surrounding environment. The constant monitoring of activity recognition was used to carry out by deployment of cameras with high cost or by personal companion devices with no easy use. In addition, the aggregation of monitored data was very complicated and impractical. However, the increasing development in sensor technologies and deployment of small-sized sensors within the mobile devices (e.g., the accelerometer, proximity sensor, magnetometer, GPS and etc.) along with the fact that these devices are carried by people throughout the day makes new generation mobile devices (e.g., smartphones) appear to be an ideal platform to be used with the purpose of human-centric sensing applications. The accelerometer sensor, which can return a real-time measurement of acceleration through all coordinate spaces, is commonly used for HAR due to that fact. It is employed either as a pedometer to measure steps counts and a total calorie consumption or as a monitor to recognize user physical activities such as postures and movements. Most of the measured events/actions/attributes are generally related to the human posture or movement (e.g., using accelerometers or GPS/Wi-Fi/Cell Tower), environmental variables (e.g., using temperature and humidity sensors, microphone and cameras), or physiological signals (e.g., attachment of external devices such as heart rate or electrocardiogram, finger pulse, etc.). In this aspect, there are many studies, refer Table 2.3, proposing to use smartphones to monitor users' daily physical activities according to their lifestyles.

2.6.2.2 Wearable Sensors Based

Wearable sensors, i.e., multiple-sensor multiple-position solutions, have been put forward in order to recognize complex activities and gestures within the HAR concept. It essentially introduces multiple-sensor placements on multiple locations of a human body to well capture specific

target activities (e.g., brushing teeth, arm and wrist movements while folding laundry, etc.), which a smartphone cannot detect by itself. With the usage of wearable sensors, the sensory context is extracted from miniature sensors integrated into garments, accessories, or straps. Especially, the traditional accelerometer based HAR solutions cannot provide activity recognitions at finer granularities for the differentiation of some postures such as sitting and lying down since there are some drawbacks observed such as mis-adjustment of the device orientation and position, or insufficient number of sensors to have enough spatial information. Hence, wearable sensors with the utilization of heterogeneous sensors have been an active research area in order to respond a growing demand for HAR systems in the health care domain, especially elder care support, assisting the cognitive disorders, and fitness and well-being management [29, 52, 53].

A smartphone can be used as a center position for external sensor attachments. Heterogeneous sensors are connected to each other and the smartphone with a wired/wireless communication (mostly Bluetooth). Proximity sensors decide the distance between sensor nodes (i.e., topology of sensor placement) by measuring the received signal strength indication (RSSI) of radio frequency in dBm. On the other hand, the deployment of heterogeneous sensors entail high cost and brings about some constraints in computations, since it requires intensive supervised learning based classification algorithms, which are mostly carried out in offline analysis and hence give an impractical solution. The constraints may also stem from sensor degradation, interconnection failures, and jitter in the sensor placement. Therefore, the reduction of sensor dimension is highly important for node interconnection, which makes the system stay unobtrusively.

2.6.3 Studies for ‘User Tracking: Transportation Modes and Location Info’

Location-based sensing [70] aims at tracking people over a period of time by recognizing their activities in terms of specifying transportation modes (e.g., walking, running, vehicle etc. when a user is outside) that they engage with as well as by discovering common places that they would like to visit. After the integration of GPS receivers into smartphones, the data collected by GPS becomes handy for network connected applications. Thereby, GPS is employed as an instrument in these applications in order to inspect for the habits and general behaviors of individuals and communities [71–73].

Investigation of mobility patterns in extracting places and activities from GPS traces have been generally implemented in a hierarchical structure [72–74]. According to the structure, the

lower level begins with the association of GPS traces with street maps, and the structure rises up by inferring and modeling activity sequences; and eventually, the structure ends up with discovering significant places from activity pattern. By taking a log of recent history of transportation modes of individuals throughout the daily life as well as mapping their location history, a general physical activity report can be documented, and also the goals of future activity plan can be reconfigured with the purpose of the health and fitness monitoring. For instance; from physiological perspective, driving behaviors are investigated in [75] by taking a consideration of trip destinations, trip times and driving efficiency.

The localization is inferred by GPS delivered speed and location information together with a large amount of a priori data (e.g., street maps). GPS provides 2D data by setting a resolution value (e.g., generally 10 m) per a certain distance within two successive data points (i.e., unit difference). Hence, the consecutive GPS readings are grouped based on their spatial relationships in order to create distinctive segmentations among GPS traces. Then, GPS traces are associated with available street maps, which are represented as directed graphs where an edge represents a street and a vertex represents the intersection of streets.

GPS cannot penetrate through walls, and thereby the received data gets degraded. Thus, the usage of GPS for location-based sensing is valid for outdoors. Once GPS times out because of the lost satellite signals, Wi-Fi scan is performed for indoors by checking for wireless access points around. Wi-Fi could be used for outdoors as well if applicable since it covers a range of 20-30 m as radius. Actually, smartphones apply a hybrid localization scheme by using GPS with network-based triangulation by leveraging wireless access points for achieving coarse positioning [76]. The network-based triangulation collects information through RF signal beacons from reachable wireless cell towers or Wi-Fi access points or even Bluetooth which is not effective due to the short-distance usage and limitations in data throughput but could be used indoor environment in presence of multiple users around. The received RF signal strength is used to measure a relative distance through the physics of signal propagation among network nodes (e.g., utilization of local and mobile base stations). Hence, by measuring sequential RSSI data, user related transportation modes can be identified. In addition, during the Wi-Fi scan, the MAC address (i.e., BSSID) of wireless access points might have already been tagged as a point of interest, which yields to retrieve location info automatically that user is in a familiar environment (e.g., office, home, gym etc.). Although GPS could detect some postures such as sitting or standing, the accelerometer sensor is rather used for

such static activities due to GPS may not provide a concise solution for differentiation of user state classes at similar speed. Besides, the inefficiency in power consumptions would be more healed in case where the accelerometer sensor is used.

Location-based sensing could also take place within the concept of community sensing by monitoring highways for real-time traffic conditions, and by forecasting probabilistic traffic congestions, thereby it would be possible to re-route the traffic flow. [74, 77–79]. This scenario can also be applied into biking [80, 81]. Bikers can share their routes to demonstrate the noisiness of the bike trails, and also to take ride statistics for a fitness log. Besides, the most significantly, the crowdedness level of metropolitan areas can be investigated in terms of daily visitor density [82, 83]. Unfortunately, some concerns such as privacy, security and resource considerations limit the expansion of location-based sensing applications since cyber-stalking [84] by tracing the revealed user locations could harm users by economically, physically, and legally. In the absence of the relevant concerns, some websites/applications such as [85] can be widely used and would help more in assisting people.

2.6.4 Studies for ‘Social Networking’

The ever-increasing ubiquity of Internet usage has enable people to exchange innumerable different form of information at a global scale. This situation has resulted in explosive growth in the creation of social network platforms (e.g., Facebook, Twitter etc.) where people can describe and share their personal interests and preferences. With the emergence of smartphones equipped with sophisticated sensors, the integration of smartphones and social networks have leveraged the data collection capability and leded the born of exciting context-aware applications as well as the evolution of the Internet of Things. However, the question of how the inference of a human relevant context can incorporate with social network platforms in an autonomous way is still the most exciting research topic in the area of ubiquitous sensing. In this sense, the researchers have been trying to create context-aware systems where diverse large data streams (e.g., image, video, user location, user transportation mode) are automatically sensed and logically fused together with the purpose of social interaction amongst individuals or groups of people. The corresponding research is called *crowdsensing* or *crowdsourcing*. “CenceMe” [86] is the foremost study which is able to infer user relevant activities, dispositions, habits and surroundings, and then to inject these information into social networking platforms. The fusing of social, sensor, and social data for context-aware computing is also studied in [87, 88]. A detailed study for the current evolution and future challenges

of the crowdsensing is given in [89, 90]. In addition, some exciting futuristic project ideas can be examined through www.funf.org.

2.6.5 Studies for ‘Healthcare and Well-being’

With the advancements and increasing deployment of micro-sensors and low-power wireless communication technologies within the Personal/Body Area Network (PAN/BAN), the studies conducted under ubiquitous computing have grown interest in the healthcare domain. Besides the high demands for applying and understanding HAR based systems, the integration of monitoring and analyzing vital sign data (e.g., heart rate, blood sugar level and pressure level, respiration rate, skin temperature, etc.) through sensors also more likely enable to change assessment, treatment and diagnostic methodologies in healthcare domain since the traditional methodologies have been based on self-reports, clinic visits and regular doctor inspections [91].

With the integration of these emerging technologies in the healthcare domain, the sensor-enabled autonomous mobile devices can help caretakers continuously monitor patients, record their wellbeing process, and report any acute situation in case where abnormal condition is detected. Thereby, it would be more easier and efficient to monitor and manage the lifestyles and well-beings of the patients with chronic diseases, the elderly people, the rehab taking patients, the patients dealing with the obesity, the patients with cognitive disorders, children. It would be even more significantly to monitor and rescue the emergent vitals and status notifying soldiers in the combat zone.

The home-based health care monitoring by mobile based devices is defined under smart home applications. The studies in [92–94] are carried out in order to create a smart home environment for treatment procedures of patients (e.g., having cardiac problem [95], or diabetics). The studies are based on collecting data through different wearable physiological sensors (e.g., body temperature, heart rate, blood pressure, blood oxygen values, respiration level, and ECGs) and also reporting feedbacks remotely to the healthcare givers. The wearable sensors including accelerometers, heart rate monitors and many others have been also studied in [96–99] in order to recognize activity patterns while measuring fitness level and discovering frequentness of body movement against obesity and weight loss programs [100], diagnosing insidious diseases (e.g., hypotension) [101], and understanding emotional states (e.g., stress level) [102, 103]. Besides, smartphones can be used as a reminding systems [104] for aging related cognitive disorders such as Alzheimer treatment. Also, like indicated

in a well-known study, UbiFit [105], smartphone can capture user relevant physical activity level and correspond the obtained information to personal fitness goals by presenting feedback reports back to the user.

There are many commercial products available at the market to give ubiquitous computing solutions in the healthcare domain. These products mostly focus on assisting people by controlling dietary programs/weight management, discovering fitness level, measuring burnt calorie or energy level, counting step numbers, and recognizing activities. Philips Directlife, FitBit Zip and Body-Media GoWear are some devices produced for tracking activity patterns, counting steps, measuring calorie burnt, and calculating distance traveled. In addition, Impact Sports ePulse proposes heart pulse monitoring system. Many other products can also be found for measuring heat flux, galvanic skin response and skin temperature.

2.6.6 Studies for ‘Monitoring Environment’

Environmental monitoring, on one hand, aims at sensing and collecting information about the surrounding environment by basically providing a personalized environmental scorecards at the human level; on the other hand, it creates an impact toward environmental exposure by contributing solutions to environmental solutions at the community level. The surrounding environment is either a small scale area (e.g., indoor) or a large one (e.g., outdoor). For indoor environments, applications to monitor HVAC systems and building maintenance are studied [106, 107]. For instance, one can use a smartphone to measure the temperature inside a room, and then the smartphone can adjust the heat, cool, or ventilate automatically in order to change air balance in a smart home environment. On the other hand, it would be more reasonable to apply environmental monitoring in the context of community sensing. The studies in [108–114] provides applications for environmental monitoring to track and notify the hazardous exposure in the environment, such as carbon emission level, air pollution, waste accumulation, water intoxication level etc. In addition, noise pollution and ambiance fingerprinting (fusion of sound, light and color) are other topics that have been studied in this content [115, 116].

2.6.7 Studies for ‘Auxiliary Toolkits for Context Inference’

As mobile phone sensing becomes richer and more sophisticated, the obtained context through sensors becomes more complex and challenging to reason into an inference. Therefore,

context-aware applications needs mobile classifier development tools, such as “Kobe” [117], “WEKA” [118], and “The Context Toolkit” [119] in order to deal with low-level context acquisition from raw sensory and to infer high-level semantic outcomes data while exhibiting efficient utilization among available resources and achieving an optimal balance among energy, latency and accuracy tradeoffs.

CHAPTER 3

CONTEXT INFERENCE: LIGHT-WEIGHT ONLINE UNSUPERVISED POSTURE DETECTION BY SMARTPHONE ACCELEROMETER

The understanding of human activity is based on the discovery of the activity pattern and accurate recognition of the activity itself. Therefore, researchers have focused on implementing pervasive computing systems in order to infer activities from unknown activity patterns, which are called the extracted context by mobile device based sensors. The existence and awareness of the context provides the capability of being conscious of physical environment or situation around mobile device users, and makes network services respond proactively and intelligently based on such awareness. Especially, the ever-increasing technical advances in embedded systems, together with the proliferation of growing development and deployment in MEMS technologies, have enabled smartphones to be re-purposed in order to recognize daily occurring human based actions, activities and interactions which mobile device users encounter with surrounding environment. It is believed that recognizing human related event patterns, called *user states*, accurately enough could give a better understanding of human behaviors, and also more significantly, could give a chance for assisting individuals in order to enhance the quality of lives. Therefore, the inference of a vast variety of human activities in a computationally pervasive way within a very diverse context acquired by a series of sensory observations has drawn much interest in research area of ubiquitous sensing. However, the evolution of the ubiquitous sensing on the resource-constrained mobile devices in terms of battery power, memory and bandwidth have empowered Cyber-Physical Systems (CPS) [120] to emerge as a promising solution for the dynamic integration of highly complex and rich interactions among the modeling computational virtual world and the exploiting heterogeneous physical world.

In this sense, many studies have been done to detect the user centric postural actions within the concept of Human Activity Recognition (HAR) [29, 31, 34, 121–124] using accelerometers, Wi-Fi, GPS or other smartphone sensors. Note that the smartphone accelerometer is solely studied for HAR based analysis in this chapter, which makes the analysis more challenging due to the lack of

multiple sensory data fusion, and also having easily distorted signals with respect to the orientation of smartphone.

Hidden Markov Models (HMMs) [125–127] or AutoRegressive (AR) models [45] are the foremost methods amongst statistical tool based classification models to detect user related physical activities by exploiting the context obtained via wearable or built-in mobile device sensors. However, these studies mostly allow predefined and user-manipulated system parameter settings, such as arbitrary formation of state transition matrix in HMMs, or build filtering coefficients in ARs which is not suitable for online processing due to the increasing computational workload while enlarging data size.

On the other hand, most studies rely on creating feature vectors at first to exploit signal characteristics of sensory data, and then classify these vectors according to specific data classes. A feature vector consists of many signal processing functions starting from mean, standard deviation, correlation to frequency and wavelet transform models [30, 51, 52, 128]. After creating a high dimensional feature vector, pattern recognition algorithms are applied to find out the hidden context inside the feature vector. The major drawback of these algorithms stems from an *offline* decision process, which is carried out as follows: First, all sensory observations are recorded; second, feature vectors are constructed by partitioning data records with a predefined window length; third, the selection of default feature vector for specifying a *training* data class is *user-manipulated*, which means the feature vector is extracted from a specific (e.g., visually observed) window of full recorded data; the final step is called *testing* to use a template matching algorithm by mapping instantly constructed feature vectors into default feature vectors. Pattern recognition techniques for clustering diverse data classes, such as k -means [41], k -Nearest Neighbors (k -NN) search [42], Support Vector Machines (SVMs) [45], are involved in this final step in order to infer the context. Unfortunately, the given clustering techniques might not be efficient while processing large data clusters and also SVMs cannot deal with multi-class classification directly. The multi-class classification problem is usually solved by decomposition of the problem into several two-class problems. Furthermore, pattern recognition toolkits such as WEKA [118] are also used to obtain classification results [26, 129].

Towards this end, this chapter proposes an online solution which intends to perfectly exploit acceleration signals within the fast Decision Tree (DT) classifiers without setting any predefined/fixed thresholds over any specific acceleration spaces in order to differentiate user activities

such as sitting, standing, walking, and running. The proposed classification method provides the following properties, which also make this chapter differ from some studies [34, 45, 47, 52, 123, 124] under a similar name:

- *unsupervised learning*: no priori information, no fixed thresholds, no initial training data classes;
- *adaptive*: robust solution to a changing orientation of the device;
- *light-weight*: efficient tree-based classification by applying sufficient signal processing toolbox: no redundant computational workload;
- *online*: instant context inference;
- *assisting*: working standalone and/or assisting other classification algorithms by creating training data classes or input matrices;
- *updating*: computational efficient update/add/delete process on training data classes.

This chapter also enhances some widely used supervised classification methods using Gaussian Mixture Models (GMM), k-NN, Linear Discriminant Analysis (LDA) for online processing by providing training data classes without a prior offline process as well as supporting the observation analysis defined in statistical based tools such as HMMs.

The whole system structure is described in Figure 3.1. According to the structure, a sequence of sensory data is collected by a sliding window. To be able to infer the hidden context (i.e., user states: sitting, standing, walking and running) inside the sequence, there are two modes suggested: standalone or assisting mode. Standalone mode uses the proposed classification algorithm. The proposed classification method provides a light-weight, online and unsupervised context inference solution by using a sufficient number of statistical/signal processing toolbox functions. It also both produces prior information, which is training data classes or system matrices, to represent user states and decides feature extraction functions to be used in assisting mode. Note that assisting mode is enabled after standalone mode finishes the class definitions for user states since the supervised learning based classification algorithms are employed in assisting mode. Assisting mode receives the sensory data sequence as input as well as having the prior information provided by the standalone mode. The input undergoes a feature extraction process, whose functions are already defined by

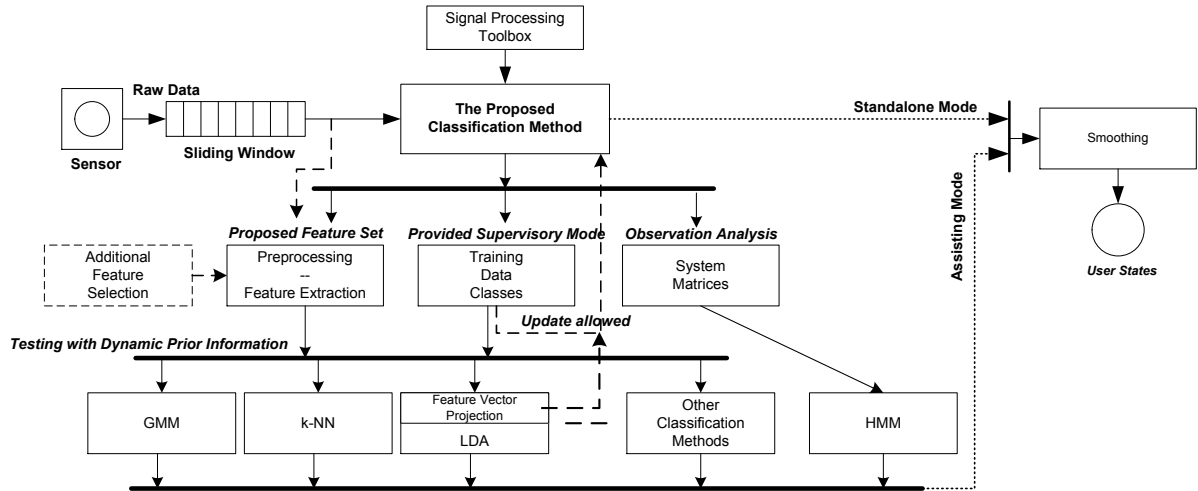


Figure 3.1: The proposed system structure for user state classification: standalone or assisting modes

standalone mode, in order to build a corresponding feature vector. Note that additional features can be added into this process. In presence of feature vector and prior information, context inference is made by diverse classification methods such as GMM, k -NN, LDA, or by statistical tool based HMM. In addition, another powerful property provided by the structure is to be able to update prior information dynamically in a computational efficient way for online processing whenever standalone mode makes a solid differentiation in user states; thereby, adaptability can be satisfied toward changing user behavior profile or sensor signal characteristics.

The whole system structure is described in Figure 3.1. According to the structure, a sequence of sensory data is collected by a sliding window. To be able to infer the hidden context (i.e., user states: sitting, standing, walking and running) inside the sequence, there are two modes suggested: standalone or assisting mode. Standalone mode uses the proposed classification algorithm from this chapter. The proposed classification method provides a light-weight, online and unsupervised context inference solution by using a sufficient number of statistical/signal processing techniques. It also both produces prior information, which are training data classes or system matrices, to represent user states and decides feature extraction functions to be used in assisting mode. Note that assisting mode is enabled after standalone mode finishes the class definitions for user states since the supervised learning based classification algorithms are employed in assisting mode. Assisting mode receives the sensory data sequence as input as well as having the prior information provided by the standalone mode. The input undergoes a feature extraction process, whose functions are already

Table 3.1: Summary of important symbols in chapter 3

Symbol	Definition (Section where the symbol is first used)
$\{x, y, z\}$	the accelerometer sensor readings (i.e., three-axial info) (3.1)
i, j	indexes for the accelerometer axes (3.1)
u, t	time indexes (3.1)
L	the length of sliding window (3.1)
f	active sliding window at current time (3.1)
f^P	previously active sliding window (i.e. $L/2$ samples earlier) (3.1)
x	feature vector (3.2.1)
s, s^*, \acute{s}	indexes for user state classes (3.2.1)
n, m	indexes for a data point in a class/feature vector (3.2.2.1)
W	feature projection matrix (3.2.2.3)
y	feature projection vector (3.2.2.2)
μ, m	mean values (3.1)
σ	standard deviation (3.1)
Λ, S	covariance matrices (3.2.2.3)
A, B	between and within class scatter matrices (3.2.2.3)
a, b	user state transition and observation matrices (3.2.3)
n, \acute{n}	the number of samples (3.2.2.3)

defined by standalone mode, in order to build a corresponding feature vector. Note that additional features can be added into this process. In presence of feature vector and prior information, context inference is made by diverse classification methods such as GMM, k -NN, LDA or HMM. In addition, another powerful property provided by the structure is to be able to update prior information dynamically in a computational efficient way for online processing whenever standalone mode makes a solid differentiation in user states; thereby, adaptability can be satisfied toward changing user behavior profile or sensor signal characteristic.

3.1 Standalone Mode: The Proposed Novel Classification Method

Accelerometer sensor retrieves three-axial acceleration data $\{x, y, z\}$ at each sampling time. Sensory readings are collected by a sliding window with a length of L and an overlap value of 50%. The length of windowing is an important design merit. The shorter windowing cannot seize the activity pattern properly, whereas the wider windowing would create a latency in detections and puts additional workload in computations. In addition, the overlap value is important as well to detect user state transitions in activity pattern.

The window at current time is called ‘‘active frame’’, and it is denoted by $f_{\{x,y,z\}}(\tau)$ where $\tau \in [t-L+1, t]$ and t is the time index and L is the total number of samples for each axis. Hence, in case where $L/2$ number of new samples inserted into the frame due to the overlapping, the proposed classification method begins to operate by receiving inputs as shown in Figure 3.2. The inputs are considered as two data sets: the active frame and the previously active frame, which is denoted as, $f_{\{x,y,z\}}^P(\tau)$ where $\tau \in [t-3L/2+1, t-L/2]$, superscript p represents for ‘‘previous time frame’’.

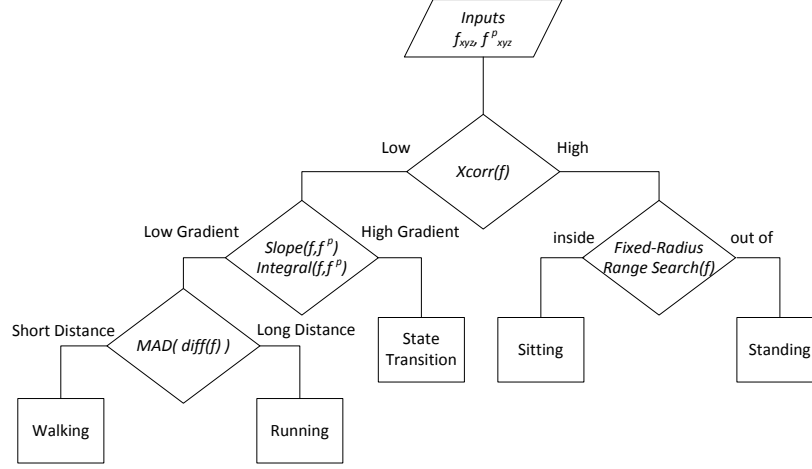


Figure 3.2: The proposed decision tree based classification method: standalone mode

For the preprocessing, a noise cancellation filter, e.g. a LMS filter, could be used before any signal processing technique is applied in order to reduce possible distortion over sensory readings. However, the proposed method would show that it can still produce valid results with/without the noise effect.

The applied method begins with normalizing each axis into unit power in order to analyze signals over the similar base (e.g., $\hat{f}_x = f_x(t) / \left(\sqrt{\sum_{\tau=t-L+1}^t f_x(\tau)} \right)$), and then taking cross-correlations among acceleration axis pairs in the active frame,

$$R_{ij}(u) = \begin{cases} \sum_{\tau=0}^{L-u-1} \hat{f}_i(\tau+u) \hat{f}_j(\tau), & u \geq 0 \\ R_{ij}(-u), & u < 0, \end{cases} \quad (3.1)$$

where u and τ time indexes, $i, j \in \{x, y, z\}$ and $i \neq j$.

If high correlations are obtained among axis pairs, $\max\{|R_{ij}|\} > \varepsilon$ where $\varepsilon \in [0.75, 1]$, user state is identified as either sitting or standing. Otherwise, it is notified that user state could be either walking/running or be in transition. Note that the applied method seeks for the highest correlations at first for specifying a starting reference point, which defines a *training data frame* for the relevant user state, so that the learnings from future sensory samplings will be more accurate. Generally speaking, experiments show that $|R_{xy}|$ mostly satisfies a highest correlation; whereas, $|R_{xz}|$ and

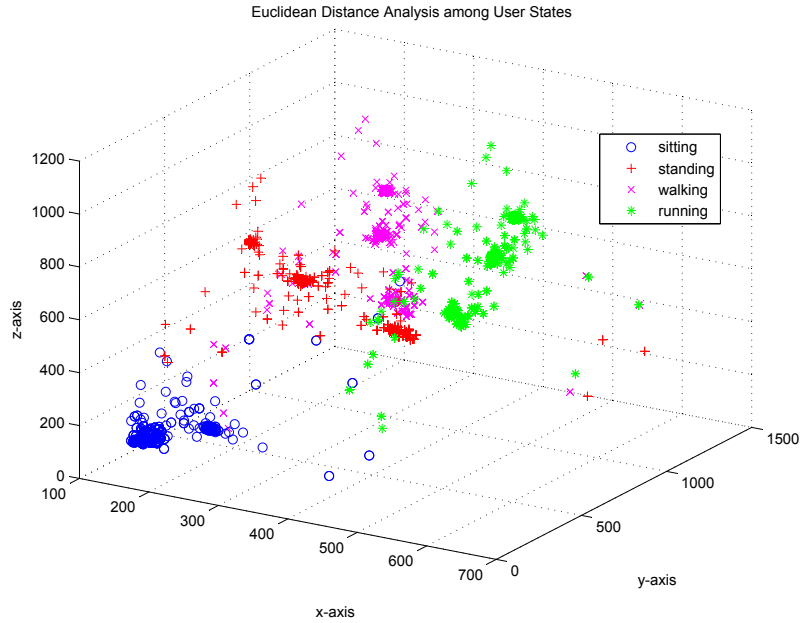


Figure 3.3: Euclidean distance analysis: user state 'sitting' is the reference point

$|R_{yz}|$ do not. Also, the training data frame of each user state can be updated whenever a clear classification result is obtained for the classification of corresponding user state.

Figure 3.3 gives the information of how user states occupy the Euclidean space. By analyzing the figure which consists of three-axial acceleration data recorded through ten minutes at 100 Hz, user state sitting can be differentiated from other user states easily over the Euclidian space by assigning a data set of user state sitting as the reference point, while the similar conclusion cannot be made that easy for user states standing, walking and running. In other words, these three user states can be put in a same group in comparison with user state sitting.

The differentiation between sitting and standing relies on Euclidean distance analysis among three-axial accelerations since relevant data samples for these two user states are scattered distinctively over the coordinate spaces. The Euclidean distance between two random points h and q on a coordinate space is given by $|hq| = \sqrt{(h_i - q_i)^2 + (h_j - q_j)^2}$ where $i \neq j$. Hence, pairwise Euclidean distance vectors between the active frame and the training data frame through each three-axial accelerations are calculated.

After that, a radius r is defined for the training data frame which belongs to the recent recognized user state. Since the dispersion in accelerations would be distinctive to select a proper

user state, the variance values of accelerations must be taken into account in order to see how far sampling points within the training data frame is spread out from the mean. Thus, the magnitude (i.e., the norm) of a vector containing standard deviation of each axis gives out a required radius: $r = \sqrt{\sum \sigma_x^2 + \sigma_y^2 + \sigma_z^2}$. Note that the geometric mean of standard deviations could also be used for defining the radius, $r = \sqrt[3]{\sigma_x \sigma_y \sigma_z}$.

In order to identify user state “sitting” from “standing” in presence of Euclidean distances and the defined radius, a distance based learning function is used for clustering user states. Range Search algorithm (i.e., Fixed-Radius Near Neighbors [130]) is implemented for this purpose. This algorithm finds all points inside the pairwise Euclidean distance vectors within a radius r centered at the mean. By applying a brute-force approach, given the set H and a distance $r > 0$, all pairs of distinct points $h, q \in H$ such that $|hq| \leq r$ is found. In addition, the radius needs to be smoothed whenever a new radius value becomes available in case where a perfect correlation satisfied among axes so as to have a better spreading circumference.

The points which stay in and out of radius r are considered to create a separation in relevant user state detections. However, to this end, the proposed method does not know yet which user state is sitting or standing due to the placement, i.e., orientation, of the smartphone. The absolute decision will be made whenever user state walking/running is recognized since user state standing lies over almost the same signal level that user state walking/running does.

Note that instead of Range Search algorithm, SVMs could be another choice to differentiate two user states from each other by using linear discrimination. SVM denotes two *user states* as binary data *classes*. The objective is to create a hyperplane which sets a rigid margin among data classes to achieve an optimal linear distance separation. The hyperplane $w^T x + w_0$ is either $\geq \text{class}\{+1\}$ or $< \text{class}\{-1\}$ where w and w_0 are defined as weight and bias respectively. The hyperplane boundary satisfies $w^T x + w_0 = 0$ for a data point x .

On the other hand, in case where correlations observed among axis pairs by (3.1) are not sufficient, i.e., $\max\{|R_{ij}|\} < \varepsilon$ where $\forall i$ and $i \neq j$, the integrals of both zero-mean versions of the inputs are taken to check if both inputs are on the same signal level. Range Search algorithm is also applied for this purpose by receiving the absolute values of the both integral results, $|\sum_{\tau=t-L+1}^t (\tau - (t - L))(f_i(\tau) - \mu_{f_i})|$ and $|\sum_{\tau=t-3L/2+1}^{t-L/2} (\tau - (t - 3L/2))(f_i^p(\tau) - \mu_{f_i^p})|$ where $i \in \{x, y, z\}$ and μ is the mean. In addition, a radius for the algorithm is defined by the geometric mean of integral results belonging to f^p , $\sqrt[3]{\prod_{\forall i} |\sum_{\tau=t-3L/2+1}^{t-L/2} (\tau - (t - 3L/2))(f_i^p(\tau) - \mu_{f_i^p})|}$. If the

signal level is similar for both inputs, user state becomes either walking or running. Otherwise, user state is in transition, thereby the previous user state is taken as the current user state.

To differentiate “walking” and “running”, it would be a reasonable start to take the first order differentiation of the relevant acceleration samples in order to better exploit the contextual information since these user states exhibit more frequently changing variations in acceleration data. For this purpose, the first order regression coefficient is defined by

$$\dot{f}_i(t) = \frac{(f_i(t+1) - f_i(t-1)) + 2(f_i(t+2) - f_i(t-2))}{2(1^2 + 2^2)}. \quad (3.2)$$

The given regression coefficient is mostly introduced in speech signal recognition systems [131, 132] to provide a temporal information about the pitches in speech signals in order for accuracy enhancement.

The next step is to extract the mean absolute deviation (MAD) of the differentiated sensory samplings so as to discern how far the points can reach.

$$(\text{MAD})_i = \frac{1}{L} \sum_{\tau=t-L+1}^t |\dot{f}_i(\tau) - \dot{\mu}_i|, \quad (3.3)$$

where $\dot{\mu}_i$ is the mean value of the modified acceleration instance (3.2) for any axis. Then, Range Search algorithm is also applied on points obtained from (3.3) to differentiate user states walking and running. When the differentiation completes, the user state which has a bigger geometric mean of all three relevant values as in (3.3) is marked as running; other one then becomes walking.

$$\{\mathbf{x}\} = \left[|\mu_{\{x,y,z\}}|, \sigma_{\{x,y,z\}}, \max\{|R_{xy}|\}, \max\{|R_{xz}|\}, \max\{|R_{yz}|\}, \Theta_{\{x,y,z\}}^{\text{slope}}, (\text{MAD})_{\{x,y,z\}} \right] \quad (3.4)$$

It has been already remarked the conditions in which user state would be in transition. The recognition process where user state just goes into or exits from a transition would yield to have a false truthfulness even though the cross-correlation analysis gives the opposite results by showing high correlations. In such cases, the slope of the active frame needs to be calculated. The relevant slope information is given by

$$\Theta_i^{\text{slope}} = \arctan \left\{ \frac{\frac{4}{L} \sum_{l=3L/4+1}^L f_i(l) - \frac{4}{L} \sum_{l=1}^{L/4} f_i(l)}{2} \right\}, \quad (3.5)$$

which basically takes the first order difference of the means of the last and the first 25% portions of the active frame.

According to the retrieved slope information from (3.5), low angles show that user state occurs at a same signal level, whereas high angles show that user state transits into another one. Basically, having low angles shows the evolution of the active frames still lies in the range of $|\mu_{f_i^p} \pm \sigma_{f_i^p}|$ where $\mu_{f_i^p}$ and $\sigma_{f_i^p}$ denote the mean and the standard deviation of the previously active frame. As a conclusion, if user transition is detected, the previous user state is taken as the current user state.

In summary, Figure 3.2 organizes the user state differentiation process from the contextual data and diminishes in a scope of decision tree based classification method. This algorithm initially takes the accelerometer data stored inside the current sliding window. Then, it applies cross-correlation function through pairs of coordinate axes in the sensory data to check for the data consistency among the axes. If there is a high match observed among the axes, then current user state becomes either sitting or standing after Range Search, or SVM, analysis over the Euclidean space. On the other hand, user state could be walking, running or in transition in case where the inconsistency occurs after the cross-correlation checks. The differences obtained from the slope and/or integral information of two subsequent sensory data sequences reveal whether or not user state is in transition. Low gradient in the differences reveals the sensory data has a stable continuity on the previous signal level through axes, i.e., user state is not in transition. Finally, the differentiation between walking and running is determined by the spreading factor of the first order regression coefficients of the sensory data. Accordingly, a long distance spread within the regression coefficients indicates user state running, while the opposite indicates user state walking.

Another significant merit provided by this algorithm is that a sufficient number of functions exploited from statistical/signal processing techniques is used for a user state classification, whereas existing classification methods do not consider this distinction either by setting simple thresholds over a specific acceleration axis while the phone is placed fixed, or by applying the whole feature set required to differentiate all user states in order for only one user state recognition.

3.2 Assisting Mode: Enhancing Existing Classification Methods for Online Processing

The proposed method not only provides a standalone online solution but also assists other classification methods by generating training data classes and/or input system matrices and by making them prepared for online processing. In this case, each user state is represented as a template

data class. The properties of each class are defined by signal processing functions, which is called *feature extraction*. After that, whenever a new data set comes in, diverse classification algorithms match (i.e. map) the new data set with an existing class to be able to identify hidden user state information in the data set. This process is called classification by supervised learning method in the study of machine learning, which allows learning the mixture parameters of the stochastic distribution representing a data set, and also assumes that the new sample comes from a known distribution.

In this section, the extraction of a feature vector, and how the proposed method can assist to other existing classification methods are examined.

3.2.1 Feature Extraction

Extracting features from a given sensory data set is an effective way to identify diverse characteristics of the data set and also to preserve class identity for each user state. In Section 2, features such as mean, standard deviation, cross-correlation, slope, first-order differentiation, integral and MAD have been used. There are also many other features, such as angular velocity, cepstral coefficients, DC gain, energy, frequency-domain entropy, highest peaks of power spectrum density, magnitude, peak-frequency, rms, variance and so on, can be found in the studies [30, 49, 51, 52, 128]. However, the ultimate goal desired for online processing is to define the dimension of feature vector $\{\mathbf{x}\}$ as the smallest as possible. Therefore, the feature vector is constructed for existing classification methods with the same signal processing toolbox that the proposed method in Section 2 uses: feature vector (3.4) with a size of $\mathcal{L} = 15$. Note that the proposed method does not use every features at decision time, however existing classification methods use all of them at every time, which burdens computational complexity in online processing.

3.2.2 Pattern Recognition Based Classification

Some widely used classification methods are introduced, and enhanced for online processing in this section. Also note that the proposed method provides the training data classes for these methods in a form of feature vectors.

3.2.2.1 Gaussian Mixture Model

User state classes can be individually represented in a mixture K -Gaussian model. Any feature vector can be drawn from this model to check for which data class encapsulates a given specific feature vector, i.e., training data set, called *clustering problem*. This is also called density problem which is defined as: given a set of N points in \mathcal{D} dimensions, $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathcal{R}^{\mathcal{D}}$, find the probability density function, $f(\mathbf{x}) \in \mathcal{F}$ on $\mathcal{R}^{\mathcal{D}}$, that is the most likely to have generated the given points.

The relevant Gaussian mixture model is given by

$$f(\mathbf{x}; \theta) = \sum_{k=1}^K q(k, \mathbf{n}) = \sum_{k=1}^K p_k \mathcal{N}(\mathbf{x}_n; \mu_k, \sigma_k), \quad (3.6)$$

where $\mathbf{n} \in \mathbf{N}$, $k \in K$ and K -components parameters are defined as $\theta = \{\theta_1, \theta_2, \dots, \theta_K\} = \{(p_1, \mu_1, \sigma_1), \dots, (p_K, \mu_K, \sigma_K)\}$ and $\{p_k, \mu_k, \sigma_k\}$ are described as the mixing probability, the mean and the standard deviation of the model respectively, and also

$$\mathcal{N}(\mathbf{x}_n; \mu_k, \sigma_k) = \frac{1}{(\sqrt{2\pi}\sigma_k)^{\mathcal{D}}} e^{-\frac{1}{2}\left(\frac{\|\mathbf{x}_n - \mu_k\|}{\sigma_k}\right)^2}, \quad (3.7)$$

with the properties of $\sum_{k=1}^K p_k = 1$, $p_k \geq 0$.

Each user state class represents a cluster which is assigned as a Gaussian model by (3.6) with a mean approximately in the middle of the cluster, and also with a standard deviation showing a measure of how far the cluster spreads out. Therefore, Gaussian parameters θ of each user state class needs to be estimated.

The maximum likelihood parameters $\hat{\theta}$ are estimated iteratively using the Expectation-Maximization (EM) algorithm according to the following steps:

- E Step:

$$p^{(l)}(k/\mathbf{n}) = \frac{p_k^{(l)} \mathcal{N}(\mathbf{x}_n; \mu_k^{(l)}, \sigma_k^{(l)})}{\sum_{k=1}^K p_k^{(l)} \mathcal{N}(\mathbf{x}_n; \mu_k^{(l)}, \sigma_k^{(l)})}, \quad (3.8)$$

where $p(k/\mathbf{n}) = q(k, \mathbf{n})/\sum_{k=1}^K q(k, \mathbf{n})$ is defined as the conditional probability of the component k given data point \mathbf{x}_n .

- M Step:

$$\begin{aligned}
\mu_k^{(l+1)} &= \frac{\sum_{n=1}^N p^{(l)}(k/n) \mathbf{x}_n}{\sum_{n=1}^N p^{(l)}(k/n)}, \\
\sigma_k^{(l+1)} &= \sqrt{\left(\frac{1}{D}\right) \frac{\sum_{n=1}^N p^{(l)}(k/n) \|\mathbf{x}_n - \mu_k^{(l+1)}\|^2}{\sum_{n=1}^N p^{(l)}(k/n)}}, \\
p_k^{(l+1)} &= \frac{1}{N} \sum_{n=1}^N p^{(l)}(k/n),
\end{aligned} \tag{3.9}$$

given initial estimates of $\{p_k^{(0)}, \mu_k^{(0)}, \sigma_k^{(0)}\}$ where l is the step index,

until the logarithmic likelihood function,

$$\lambda(\mathbf{x}; \theta) = \sum_{n=1}^N \log \sum_{k=1}^K p_k \mathcal{N}(\mathbf{x}_n; \mu_k, \sigma_k), \tag{3.10}$$

converges to a local maxima, which is computed by partial differentiations of λ with respect to μ_k, σ_k, p_k .

For all user state classes, the corresponding joint probability of density function (pdf) of the mixture K -Gaussian model is finally given by

$$p(\mathbf{x}_n | s) = \sum_{k=1}^K p_{sk} \mathcal{N}(\mathbf{x}_n; \mu_{sk}, \Lambda_{sk}), \tag{3.11}$$

where this time $\{p_{sk}, \mu_{sk}, \Lambda_{sk}\}$ denote the weight, the mean vector and the covariance matrix of the k -th Gaussian in user state s respectively.

In the supervised learning case, the number of classes and their parameters are known. Hence, with help of the obtained training data classes by the proposed systems, the parameters of (3.8) and (3.9) can estimated through (6.16), and (3.11) can be derived. In the presence of a new feature vector, the desired user state class is identified s^* by the cluster index satisfying the maximum likelihood (ML) estimation,

$$s^* = \underset{s}{\operatorname{argmax}} \{p(\mathbf{x}_n | s)\}, \tag{3.12}$$

3.2.2.2 k -Nearest Neighbors Search (k -NN)

This algorithm assigns the nearest class set x^m , $1 \leq m \leq k$ for the input feature vector by defining a dissimilarity function which measures the nearness between the training data set and the new data points in the feature vector. The dissimilarity function is generally defined by the squared Euclidean distance $d(x, x^m) = (x - x^m)^T(x - x^m)$. However, the Euclidean distance does not take into consideration how the data is spread out, and may let the largest length scale between data points dominate the dissimilarity function. Therefore, the Mahalanobis distance $d(x, x^m) = (x - x^m)^T \Lambda^{-1}(x - x^m)$ where Λ is the covariance matrix could be used to rescale all length of scales to be essentially equal.

The algorithm classifies a feature vector x_n given training data x^m where

$$m^* = \underset{m}{\operatorname{argmin}}\{d(x_n, x^m)\}, \quad 1 \leq m \leq k, \quad (3.13)$$

However, the algorithm becomes more computational expensive in computational manner as the dimension of the feature vector increases. Therefore, Principal Components Analysis (PCA) is introduced [133] to reduce the dimension by replace $\{x\}$ with a low dimensional projection $\{y\}$. Hence, the dissimilarity function is denoted by $d(y_n, y^m)$. However, this method may not be helpful to separate classes from each other while getting the lower dimensions. Hence, another approach called Linear Discriminant Analysis (LDA) is introduced in [35] to reduce the dimension of feature vectors while resulting in separation among classes.

3.2.2.3 Linear Discriminant Analysis (LDA)

In the supervised learning method where distinctive class information is available, the dimension of the feature vector can be reduced even while continuing to obtain a solid classification. A supervised linear projection is defined as $y = W^T x$ where W is the projection matrix, \mathcal{L} is the dimension of y , $\dim\{W\} = \mathcal{D} \times \mathcal{L}$, $\mathcal{L} < \mathcal{D}$. Recall that \mathcal{D} denotes the dimension of the feature vector x . Since each user state class s is defined with a Gaussian model $p(x_s) = \mathcal{N}(x_s; m_s, S_s)$ and the linear projection is being used, the Gaussian model turns into $p(y_s) = \mathcal{N}(y_s; \mu_s, \sigma_s^2)$ where $\mu_s = W^T m_s$ and $\sigma_s^2 = W^T S_s W$.

The most efficient projection matrix creates a minimal overlap among the projected distributions, which can be achieved when the projected means are maximally separated and the projected

variance are not large enough. Hence, the objective equation is known via maximizing the Fisher criterion [134]:

$$\begin{aligned} \frac{(\mu_s - \mu_{\acute{s}})^2}{\pi_s \sigma_s^2 + \pi_{\acute{s}} \sigma_{\acute{s}}^2} \rightarrow \mathcal{F}(W) &= \frac{W^T (m_s - m_{\acute{s}})(m_s - m_{\acute{s}})^T W}{W^T (\pi_s \sigma_s^2 + \pi_{\acute{s}} \sigma_{\acute{s}}^2)^T W}, \\ &= \frac{W^T A W}{W^T B W} \end{aligned} \quad (3.14)$$

where π_s is the fraction of the dataset in class s , $s \neq \acute{s}$, $A = (m_s - m_{\acute{s}})(m_s - m_{\acute{s}})^T$ and $B = \pi_s \sigma_s^2 + \pi_{\acute{s}} \sigma_{\acute{s}}^2$.

In case of more than one dimension and more than two classes, Fisher's projection method (3.14) is generalized by Canonical Variates method, which takes $p(x) = \mathcal{N}(x; \mu_s, \Lambda_s)$ and replace it with $p(y) = \mathcal{N}(W^T x; W^T \mu_s, W^T \Lambda_s W)$. Then, the following matrices are defined:

- Between Class Scatter: $A = \sum_{\forall s} n_s (\mu_s - \mu)(\mu_s - \mu)^T$ where μ is the mean of all data set and n_s is the number of data points in class s .
- Within Class Scatter: $B = \sum_{\forall s} n_s B_s$ where $B_s = \frac{1}{n_s} \sum_{u=1}^{n_s} (x_u^s - \mu_s)(x_u^s - \mu_s)^T = \Lambda_s$.

By assuming B is invertible, the Cholesky factor \tilde{B} is defined by $\tilde{B}^T \tilde{B} = B$. In addition, defining $\tilde{W} = \tilde{B} W$ yields to have $W = \tilde{B}^{-1} \tilde{W}$. Then, the latest Fisher criterion becomes:

$$\mathcal{F}(\tilde{W}) = \frac{(\tilde{B}^{-1} \tilde{W})^T A \tilde{B}^{-1} \tilde{W}}{(\tilde{B}^{-1} \tilde{W})^T \tilde{B}^T \tilde{B} \tilde{B}^{-1} \tilde{W}} = \frac{\tilde{W}^T \tilde{B}^{-T} A \tilde{B}^{-1} \tilde{W}}{\tilde{W}^T \tilde{W}}, \quad (3.15)$$

which subjects to $\tilde{W}^T \tilde{W} = I$.

Since it is symmetric, the term $\tilde{B}^{-T} A \tilde{B}^{-1}$ in (3.15) turns into a special case of the real valued eigen-decomposition form, Schur decomposition, QEQ^T where E is a diagonal matrix which contains the eigenvalues. Hence, \tilde{W} is set to hold relevant eigenvectors. Finally, the projection matrix becomes

$$W = \tilde{B}^{-1} \tilde{W}. \quad (3.16)$$

To differentiate user states, the newest (i.e., lower dimensional) form of data classes and feature vectors are modeled by (3.11) and (3.12).

3.2.2.4 Online Processing - Dynamic Training/Supervisory Classes Update

For an online classification algorithm, one of the most important things is to reduce computational burden and stay away from large amount of data manipulations. Note that computational complexity requires $\mathcal{O}(\eta\mathcal{L}\mathcal{D}^2)$, $\mathcal{O}(\mathcal{L}^2\mathcal{D})$, and $\mathcal{O}((\mathcal{L} + c)^2\mathcal{D})$ times for GMM, k -NN, and LDA with Schur decomposition respectively where η is number of iterations during EM algorithm in GMM, and c is the total number of user state classes. In this sense, the parameters related to GMM, k -NN, and exceptionally LDA algorithms need to be dynamically updated in an efficient way rather than just computing relevant system parameters all over again in case where new training data samples inserted into an existing class or a new data class is added/deleted. Especially, matrix multiplications, which take $\mathcal{O}(c\mathcal{L}\mathcal{D})$ time, need to be manipulated with ease during the update.

Here is the suggested update for the supervised classification algorithms:

- Adding a new data to an existing class i : (\hat{n}_s : the number of added training samples)

- Common properties for all classification methods:

- * $\hat{\mu}_s = \mu_s + \Delta_{\mu_s}$
- * $\Delta_{\mu_s} = ((\sum_{u=n_s+1}^{n_s+\hat{n}_s} \mathbf{x}_u^s) - \hat{n}_s\mu_s)/(n_s + \hat{n}_s)$

- Only additional for LDA:

- * $\hat{\mu} = \frac{((n_s+\hat{n}_s)\mu + \sum_{\forall s} \sum_{u=n_s+1}^{n_s+\hat{n}_s} \mathbf{x}_u^s)}{(\sum_{\forall s} n_s + \hat{n}_s)}$
- * $\hat{A} = \sum_{\forall s} (n_s + \hat{n}_s)(\hat{\mu}_s - \hat{\mu})(\hat{\mu}_s - \hat{\mu})^T$
- * $\hat{B}_s = \frac{1}{n_s + \hat{n}_s} \sum_{u=1}^{n_s + \hat{n}_s} (\mathbf{x}_u^s - \hat{\mu}_s)(\mathbf{x}_u^s - \hat{\mu}_s)^T$
- * $\hat{B} = \sum_{\forall s} (n_s B_s + n_s \Delta_{\mu_s} \Delta_{\mu_s}^T + \sum_{u=n_s+1}^{n_s+\hat{n}_s} (\mathbf{x}_u^s - \hat{\mu}_s)(\mathbf{x}_u^s - \hat{\mu}_s)^T)$

- Adding/Deleting a class $\{s^*\}$:

- Common properties for all classification methods:

- * create/delete $\mu_{\{s^*\}}$ and $\Lambda_{\{s^*\}}$

- Only additional for LDA:

- * $\hat{B} = B \pm n_{\{s^*\}} B_{\{s^*\}}$
- * $\hat{A} = A + n \Delta_{\mu} \Delta_{\mu}^T \pm n_{\{s^*\}} (\mu_{\{s^*\}} - \hat{\mu})(\mu_{\{s^*\}} - \hat{\mu})^T$
- * $\Delta_{\mu} = \hat{\mu} - \mu = \pm n_{\{s^*\}} (\mu_{\{s^*\}} - \mu)/(n \pm n_{\{s^*\}})$

Note that after the update completes, only for LDA algorithm, the linear projection matrix needs to be re-computed.

3.2.3 Hidden Markov Models (HMMs)

Hidden Markov Models (HMMs) based human postural behavior and activity detection is the mostly used statistical tool in HAR objected applications. In HMMs, a system parameter called observation emission matrix is given with the help of (3.11) by

$$b_{s,O_t} = p(O_t/q_t = s) = \sum_{k=1}^K p_{sk} \mathcal{N}(O_t; \mu_{sk}, \Lambda_{sk}), \quad (3.17)$$

where s , O_t and q_t is a user state (i.e., a data class), instant observation (i.e. feature vector), and user state instance in a sequence respectively. The user state inference according to (3.2.2.1) is done by checking for Gaussian membership of observations in reduced dimensions and selected the suitable class according to majority vote in the classification.

On the other hand, for improving HMMs for online processing, a similar matrix like in (3.17) can be constructed easily in light of the proposed method, which is given by

$$b_{s,O_t} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} R_{in}^{Euc.} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + R_{out}^{Euc.} \right) + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} R_{in}^{MAD} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} R_{out}^{MAD} \right), \quad (3.18)$$

where \otimes is the Kronecker product; $R_{in/out}^{Euc.}$ is the percentage of points stay in and out of radius within the Fixed Search algorithm for Euclidean distance analysis to differentiate user states sitting, $\{in\}$, and standing, $\{out\}$; and $R_{in/out}^{MAD}$ is the same algorithm approach but this time based on MAD analysis to differentiate user states walking, $\{in\}$, and running, $\{out\}$.

In addition, another system parameter, user state transition matrix, is also defined as

$$a_{s,\acute{s}} = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$$

for the HMM relevant operations in where s and \acute{s} denote user states, which are ordered as sitting, standing, walking, and running. According to the defined user state transition matrix, there is no

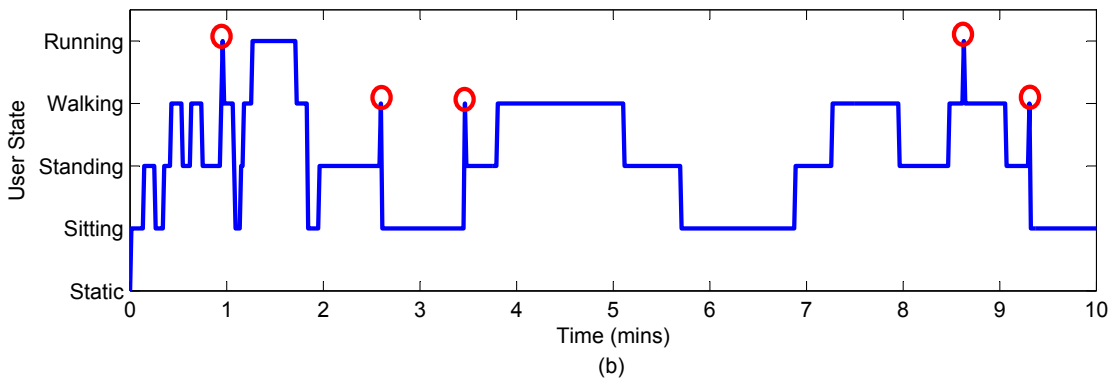
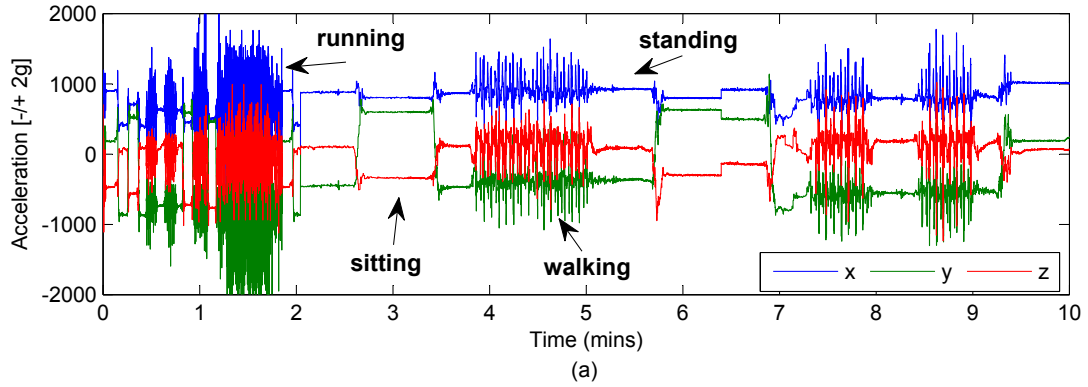


Figure 3.4: The context inference from the accelerometer sensor: (a) a ten-minute recording of three-axial acceleration signals while user posture changes, (b) the corresponding user state representations before smoothing is applied.

possibility to transit from sitting to running, from standing to running, or vice versa. The rest of the elements in the matrix has equal probability with respect to other state transitions.

3.3 Performance Evaluation

To demonstrate the effectiveness of the proposed classification method, experiments are carried out by the Blackberry RIM Storm II 9550 smartphone as target device. The Blackberry Java 7.1 SDK is used for programming implementations and Eclipse is used as software development tool. Storm II consists of 3-axial accelerometer named ADXL346 from Analog Devices. Figure 3.4 shows a ten-minute recording of the collected sensory readings, and draws the track of user state recognitions with respect to changing context in the readings.

The target device is considered to be put in trousers pocket. A change in orientation of the device, such as rotation, is an important design drawback for the most of classification algorithms, especially for those which solely rely on exploiting specific axis information. Since the sensor is not placed fixed, it would produce some distortion over acceleration axes. Upward or downward position of the device causes x-axis flipped to y-axis or vice versa. Therefore, the device is placed fixed in many studies in order not to change the orientation which yields to have false truthfulness in desired activity recognitions. The applied classification method in this chapter is not affected much by rotation of the device. As a worst case scenario, in case where the device changes its rotation, the update process will handle the adaptation problem.

With the utilization of the proposed classification method, the accelerometer sensor is sampled at $f_s = \{100, 50, 25\}$ Hz. The samplings are windowed with the window specifications of $L = 3f_s$ and 50% overlap value. Exceptionally, when $f_s = 100$ Hz is taken, L could be $2f_s$ since the lower sampling frequencies cannot resolve the inference problem when the window size is not adequate. The proposed method infers user states from the obtained context through the accelerometer sensor with almost a perfect accuracy as shown in Figure 3.4. In addition, user's quick movements while state transitions may lead to false statements in recognition process, especially any user transition between sitting/standing/walking might be detected as walking or any user transition from/to walking might be detected as running. In such cases, a basic *smoothing* technique is applied by taking a majority voting scheme by a sliding window with a specific length of user state history to prevent from having false truthfulness in user state recognitions. Hence, the red circle in Figure 3.4.b is corrected since the preceding and proceeding user states are different than what is perceived.

The experiments are carried out after obtaining long time accelerometer data recordings from three different individuals. User state recognition analysis is examined through Figure 3.2, (3.12), (3.13), (3.16) and (3.18) for each classification method. According to results, Table 3.2 shows the confusion matrix for user state recognitions under different classification methods at 100 Hz accelerometer sensor samplings. The proposed classification method, labeled as DT, achieves a great differentiation in user states recognitions because the applied methodology well analyzes the acceleration signals in order for defined postural movements, and exploits sufficient features in order to make such differentiations in user states. In addition, other existing methods succeed very reasonable truthfulness in user state recognitions. It is because that online processing allows them to have updated training classes for the clustering problem. More significantly, it also achieves

Table 3.2: Confusion Matrix 1: user state recognition under different classification methods at 100 Hz sampling

Method at 100 Hz		a	b	c	d
DT	a	99.5	0.5	0	0
	b	0	98.5	1.5	0
	c	0	1.3	97	1.7
	d	0	0	8.5	91.5
GMM	a	96.2	3.3	0.5	0
	b	1.9	93	5.1	0
	c	0.5	3.5	90	6
	d	0	0	8.5	91.5
k -NN	a	98.5	1	0.5	0
	b	0	96.5	3.5	0
	c	0	1.8	95.5	3.7
	d	0	0	9.5	90.5
LDA	a	96.2	3.2	0.6	0
	b	3.2	90.5	6.3	0
	c	0.5	4.5	88.5	6.5
	d	0	4.1	11.2	85.7
HMM	a	99	1	0	0
	b	0	98.2	1.8	0
	c	0	1.75	96.5	1.75
	d	0	0	8.5	91.5

a: sitting, b: standing, c: walking, d: running

adaptability towards some heterogeneities observed over accelerations signals such as distortion due to the rapid movement, or change in orientation. Only LDA algorithm seems to have worse outcomes in the user state recognition problem. The result may cause from the size of feature vector. Since the feature vector in this study is set to a length of 15, it may not resolve the clustering problem after size reduction due to the projection. Note that most studies mentioned in Section 3.2 have a very large size of feature vectors, e.g. size of over 40 features. In summary, existing classification methods fall behind in comparison with the proposed method even if they benefit from all elements of a feature vector while differentiating user state classes.

Moreover, Table 3.3 shows how the proposed classification method responds under lower sampling frequencies. The method still achieves a solid differentiation between sitting and standing; whereas, there are some confusion between walking and running. Due to lower number of samplings obtained by sensory operations, and also actions such as walking/running has variant signal characteristic, it would be difficult to have a pure data frame for each user state representation. Hence, clustering problem becomes more challenging to be resolved.

Table 3.3: Confusion Matrix 2: DT under different sampling frequencies

DT at f_s		a	b	c	d
50 Hz	a	99.1	0.9	0	0
	b	0	96.1	3.9	0
	c	0	3.6	87.7	8.7
	d	0	0	15.5	84.5
25 Hz	a	98	2	0	0
	b	0	92.5	7.5	0
	c	0	10.6	79	10.4
	d	0	0	22	78

a: sitting, b: standing, c: walking, d: running

In conclusion, the proposed method achieves solid user state differentiations under variant sampling frequencies by showing an above 90% overall accuracy at each f_s , and it also compares the outperforming result by showing an 10% overall enhancement for each user state at 100 Hz with the widely used GMM, k -NN, LDA and HMM based classification, which applies *supervised* and computational *heavy-weight* algorithms using high dimensional feature vectors and training data classes, and which also be light-weighted for online analysis by this chapter. In addition, the proposed method infers user states from the obtained context through the accelerometer sensor with almost a perfect accuracy at 100 Hz, as shown in Figure 3.4.b. For the lower sampling frequencies, the method still makes high accurate differentiations in user states.

A major challenge in extracting and providing user contextual information is based on monitoring the context continuously. Constantly context monitoring in a sensor-rich mobile phone imposes heavy workloads which cause some limitation in computing and battery power. Therefore, there should be a mechanism to organize required processes for analyzing the context in a such way that redundant repetitions of the same contextual information can be avoided.

Once a transition occurs into a new user state, there is no necessity to recognize and notify the same context redundantly again and again as long as user state remains unchanged. A mechanism can be considered to elaborate computational complexity during the processing of a sensor data; as a result, a decision on the continuity of user state can be made at an early age of full processing pipeline by achieving significant computational overhead save.

As illustrated in Figure 3.5, detection of an user contextual information is recognized by employing an intelligent computational method. It is aimed to perform an inexpensive strategy

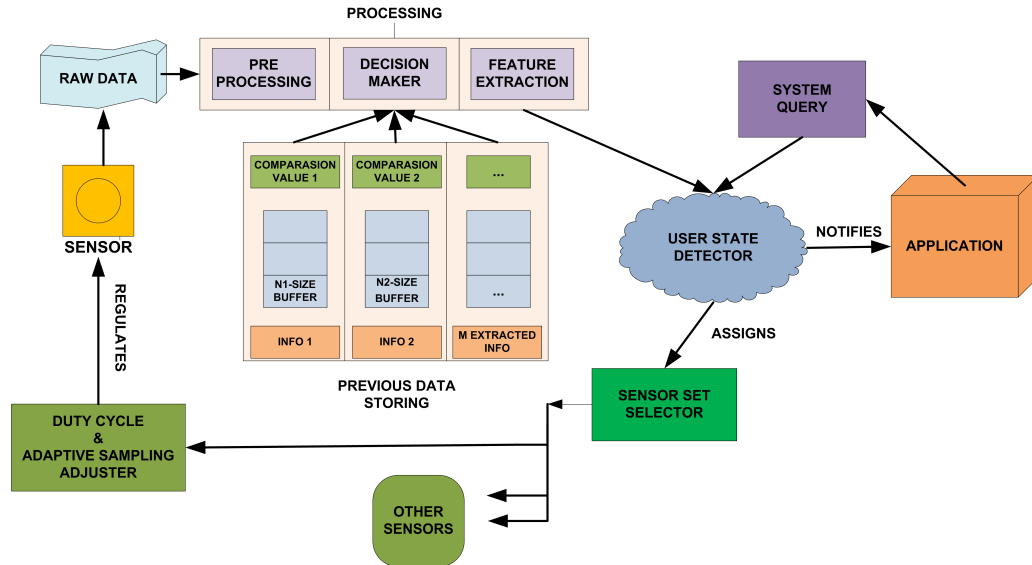


Figure 3.5: Context monitoring mechanism

without completely exploiting a raw sensor data to make the decision on continuity of current user state.

The conventional user state recognition method tries to process and analyze raw sensor data, exploit context information, and compare it with a classification algorithm based on a decision tree logic. In this method, basically every raw sensor data is carried through a processing pipeline, and ends with an user state decision. In proposed approach, an intelligently pipelined context monitoring mechanism is introduced to reduce high cost for required operation sequences. Main objective of the mechanism is to notify user states to a running application only when an user state transition occurs.

A sensor produces a discrete raw data periodically. Data entries into a processing pipeline; output of the pipeline is to inform possible change of user states or not since the data is found so identical to previous ones which were buffered.

3.4 Additional Topic: Context Monitoring Mechanism

The pipeline starts with a preprocessing structure. This structure basically filters out the required information from raw sensor data. The information varies from one sensor to another. For example; accelerometer and microphone gives away only one and required information whereas Wi-

Fi and GPS returns a package that includes different kind of information; therefore, there might be more than one monitored information sent by sensors. Different number of contextual information forces to have same number of information buffers. Each buffer has same kind of contextual data which belongs to previous samplings. Buffers are designed to gain flexibility and speed on decision making, and also most importantly to help in decrease of the number of redundant computational operations which can yield to stop the pipeline working before feature extraction algorithms are applied. For example; let us imagine that accelerometer converts an analog signal and returns related discrete digital data a hundred times in a second; since any user state cannot be changed drastically that the sensor misses transitions, there is no point of processing all algorithms at each sensor readings, and then send detected user state information to running application all the time. At that point, a tradeoff might come up for buffer size that resolve the accuracy of decision on matching with current user state.

Buffers consist comparison variables which are obtained from statistical analyses on the previously stored contextual information. Comparison variables can be standard deviation, variance or any other result of applied statistical tools. At the middle stage of processing pipeline, decision maker classifications are applied among current and previous contextual information to interrogate the existence of future processes such as feature extraction. Feature extraction block is the place where a new and different contextual information is exploited to bring about a possible user state transition. For instance; If environment was detected as in silence before, 'quite', and if newly obtained data indicates there is a changing situation in environment context, feature extraction block analyzes the data and send out resulting outcome to user state detector block to decide on which user state will be selected, would it be 'loud', 'music' or 'speech'?

User state detector block decides user state transitions, and inform them to applications. Another important property belonging to this block is to assign required sensor set for recognition of current user state and its possible transitions.

Duty cycle and adaptive sampling adjuster block basically keeps tracking length of intervals in which sensor readings do not yield to cause to miss any user state transition. Therefore, depending on how long time is spent on an user state, duty cycles and adaptive sampling periods can be adjusted to secure energy efficiency.

As last, application is notified when a user state transition occurs; it, as naturally, queries the mechanism at any time such as that a user defined event is wanted to be queried to check whether

it can be happening or not. In summary, this chapter provides online classification methods to detect the mobile device based user centric postural actions (user states are defined as sitting, standing, walking, running) through the smartphone accelerometer sensor by developing a novel Decision Tree (DT) based method in unsupervised learning mode as well as by improving some widely used existing classification methods (GMM, k -NN, LDA, HMM) for online processing. The proposed classification method exploits the context provided by the acceleration signals effectively using an adequate usage of signal processing techniques and Fixed-Radius Neighbors Search algorithm (SVM may also be used instead) within the concept of DT classifiers. Hence, there is no need for creating high dimensional feature vectors to be fed into pattern recognition algorithms for clustering the desired user state classes, which allows algorithms to gain the lightweight property. Furthermore, the proposed method does not need any a priori information about user state classes; whereas, it can provide training data classes to assist other classification methods, which employ feature vectors. To demonstrate the effectiveness of the proposed classification method and to prove its ability to make a solid differentiations among user states even under different sampling frequencies, the experiments are carried out successfully through the accelerometer sensor built-in a smartphone. In summary, the proposed method achieves solid user state differentiations under variant sampling frequencies by showing an above 90% overall accuracy at each f_s , and it also outperforms GMM, k -NN, LDA and HMM based classifications by showing an 10% overall enhancement for each user state at 100 Hz. Thereby, this chapter aims at providing a solution for the tradeoff existing between context accuracy and computational power in resource-constrained mobile sensing platforms.

CHAPTER 4

CONTEXT-AWARE FRAMEWORK: A BASIC DESIGN

The understanding of human activity is based on discovery of the activity pattern and accurate recognition of the activity itself. Therefore, researchers have focused on implementing pervasive systems in order to create high-level conceptual models to infer activities and low-level sensory models to extract context from unknown activity patterns. In this sense, a successful research has been conducted in the area of human-centric ubiquitous sensing.

Specifically, smartphones could provide a large number of applications within the defined research area. Human beings involve in a vast variety of activities within a very diverse context. A specific context can be extracted by a smartphone application, which acquires relevant data through built-in sensors. A desired activity within the context is then inferred by successful algorithmic implementations. Unfortunately, all of these operations put a heavy workload on the smartphone processor and sensors. Constantly running built-in sensors consume relatively much more power than a smartphone does for fundamental functions such as calling or text messaging. Therefore, mobile device batteries do not last a long time while operating sensors simultaneously.

However, some application fields like medical (see [66, 67]) or personal sensing (see [63–65]) could use mobile devices as measurement devices. Suppose that a patient with heart disease needs an application to keep track of his/her heart beats by a wearable sensor. The relevant application would monitor and evaluate data samples, more importantly, it could notify health centers in case an emergency situation occurs [68]. Another example could be marked by studies which recognize stress level since a balanced stress level has been studied to be an important factor to improve quality of life [69]. Consequently, context-aware applications are becoming essential in our day to day life which in return implies a greater power consumption required by smartphones.

In this sense, a framework is required to create a control mechanism for sensor utilizations and to help context-aware applications work their functionalities properly.

This chapter proposes an inhomogeneous (time-variant) Hidden Markov Model (HMM) based framework in order to represent user states by defining them as an outcome of either the recognition or estimation model. The framework also applies different sensory sampling operations in order to examine the device battery lifetime. With this method, sleeping time for a sensor can be extended by entering its idle operating mode frequently. While the sensor is in idle mode, the user might either keep or change his/her current activity. Thereby, a statistical model is required to track *time-variant* user activity profiles in order to predict the best likely user state that fits into instant user behavior. As a result, user states are either recognized as an inference of actual sensor readings or as an estimation of the missing inference. The ultimate goal of this chapter is to find a proper way to balance the trade-off while representing user states accurately as well as making sure that power efficiency is maximized.

The followings highlight some key elements introduced by this chapter:

- This research intends to create and clarify an effective HMM based framework to guide the development of real-time operating future context-aware applications.
- The framework accepts that user behavior changes in time; thereby, it assumes that HMM parameters are inhomogeneous (i.e., time-variant). As a result, there is no *a priori* assumption revealed on system parameters such as how HMM evolves in terms of the information belonging to the probability of HMM state transitions.
- Adaptability problem is introduced for inhomogeneous user behaviors, and a relevant solution is given by the uncertainty of entropy rate. It is shown that the convergence of entropy rate could be used to notify a specific user behavior.
- An effective online unsupervised user state recognition algorithm for postural actions ‘sitting’ and ‘standing’ for context extraction through accelerometer data is provided in order to infer relevant activities within the provided context, and also to conduct the observation analysis of HMM.
- By changing sampling frequencies in sensory operations, the trade-off analysis is examined. The connection between the context inference under different sampling frequencies and its effect on the convergence of entropy rate is discovered. Accordingly, the slower sampling methods could cause a failure in the convergence of entropy rate. Thereby, the discovery

could be used to make a prediction on the actual accuracy of a HAR based application performance in a real-time environment since there is not an exact solution to compare the actual user behaviors with the recognized behaviors. With this discovery, a feedback mechanism can be built to monitor the convergence of entropy rate while change sensory sampling operations. If the convergence stabilizes, then the period of sensory samplings is extended; or vice versa.

- Missing observations occurred during the slower sensory samplings are found by Naive Bayesian approach within HMMs.
- A proper simulation model is built around the accelerometer sensor so that the trade-off between accuracy in user state representations and efficiency in power consumption could be examined in detail.
- A smartphone application using the accelerometer sensor is implemented to show the effectiveness of the proposed framework.

The outline of the chapter is as follows: Section 4.1 includes a summary of prior works in this field including a discussion and recommendations on the ongoing research. In Section 4.2, the proposed framework is constructed in light of the defined purposes above. Section 4.3 and Section 4.4 are reserved for results obtained by simulations and a smartphone application in order to validate the proposed framework.

4.1 Prior Works and Discussions

Many studies can be found in which a framework is proposed to capture and evaluate sensory contextual information. Most of the studies rely on recognition of user activities and definition of common user behaviors. For instance, gesture recognition of users is well studied using video cameras. The applied methods in relevant studies are based on statistical models, which this chapter also intends to use. However, none of these studies engage themselves to model a common framework in order to construct a base structure for future context-aware applications. They would rather have canalized solutions to solve their own unique applications instead of a generalized approach. Therefore, these studies mostly focus on a specific sensor to discover possible target applications in order to exploit contextual data. A generic framework which fulfills requirements set by all types

of context-aware applications was not identified. In contrast, all of the studies put the importance on evident trade-off, which is triggered by reducing power consumption while being intended to continuously receive accurate sensor contextual data.

From the literature search, it is important to refer to Wang *et al.* [12, 135] who proposes a sensor management system, which is called Energy Efficient Mobile Sensing System (EEMSS). This system models user states as a discrete time Markov chain and improves a device battery life by powering a minimum set of sensors and also by applying duty cycling into sensor operation. However, sensors have fixed duty cycles when they are active, and they are not adjustable to different user behaviors. Also, given system is predetermined and not time-variant. Another study which analyzes hierarchical sensor management systems is introduced with a name of "SeeMon" in [28]. This system achieves energy efficiency and satisfies less computational complexity by performing continuous detection of context recognition when changes occur during context monitoring. Moreover, [33] demonstrates advantages of a dynamic sensor selection scheme for accuracy-power trade-off in user state recognition. From a power efficiency standpoint, Rachuri *et al.* [6] uses different sampling period schemes for querying sensory data in continuous sensing mobile systems to evaluate accuracy-power trade-offs.

Statistical tools based, such as Hidden Markov Models (HMMs), posture and activity detection is proposed in many studies [29, 125–127]. These studies mostly lack some key issues as follows: First of all, observations are considered directly as sensory data without applying any signal processing techniques. This consideration stems from the general definition of HMMs, which are constructed with Gaussian featured observation sequences. In addition, state transition matrices are manipulated in order for systems to infer predefined user states given observation sequences. In addition, these matrices are generally arbitrarily formed, and mostly based on visual observations. Therefore, it would be so *difficult to answer* some questions such as "*How would these systems act against different user profiles?*" and "*What if a sensor could be used for different context extraction?, how would observations be formed?*."

Other methods used for activity detection benefit from signal processing and pattern recognition techniques. Relevant studies such as [34, 121–124, 136, 137] construct a framework which is designed to control the acquisition and interpretation of data from one or more sensor(s). This inference framework is implemented for a range of context aware workloads by using diverse algorithmic primitives. The framework analyzes spatial characteristics of sensory signals based on either time

space or frequency spectrum. High-dimensional feature matrices are constructed by including many signal processing functions ranging from angular velocity, correlation, energy, entropy, magnitude, mean, min-max values, standard deviation to frequency domain entropy, peak frequency analysis and power spectrum density. Then, these large feature matrices undergo pattern recognition techniques such as k-means clustering, nearest neighbors search, and Support Vector Machines (SVM). The major drawback of these studies stem from *offline* decision process executed as follows: First, total sensory observations are recorded. Second, a feature vector is constructed over partitioned data records with a predefined window length. Third, default feature matrices are *user-manipulated*, which are extracted from a specific window of recorded data to be able to describe specific states. The final step is using a template match, i.e. *supervised learning*, by comparing default feature matrices with instantly constructed feature matrices. With this applied methodology, it is not clear how resource-constrained mobile devices provide highly complex signal processing operations by handling large size data manipulation.

In addition, the use of entropy rate as optimization criteria within the concept of HMMs to address a solution in order to reduce power consumption is introduced in [138] and implemented in [139]. These studies accept offline decision process to attain an optimal sampling policy through recorded HMM observations with the known transition probabilities among states. The policy aims at selecting most informative sub-sequences of observations by measuring the uncertainty of expected entropy rate with respect to a power consumption constraint. By this policy, the time intervals where the most informative sub-sequences are selected define the length of idle times in sensory operations. Unfortunately, the policy gives a deterministic sampling strategy, and the given strategy is valid for time-invariant systems by providing example of defining measurement epochs to build a temperature time series during the day. However, the optimal strategies would work only if the weather season does not change since it accepts the transitions probabilities among temperature measurements are fixed; thereby, this solution does not satisfy the trade-off requirement needed by HAR based applications, which rely on actively changing distinctive user behaviors.

4.2 Construction of the Proposed Framework

A Hidden Markov Model (HMM) [140] can be applied to a system which aims to recognize user states, as shown in Figure 4.1. In this system, sensor readings (i.e., extracted user contexts through mobile device based sensors) are seen as inputs. These readings undergo a series of sig-

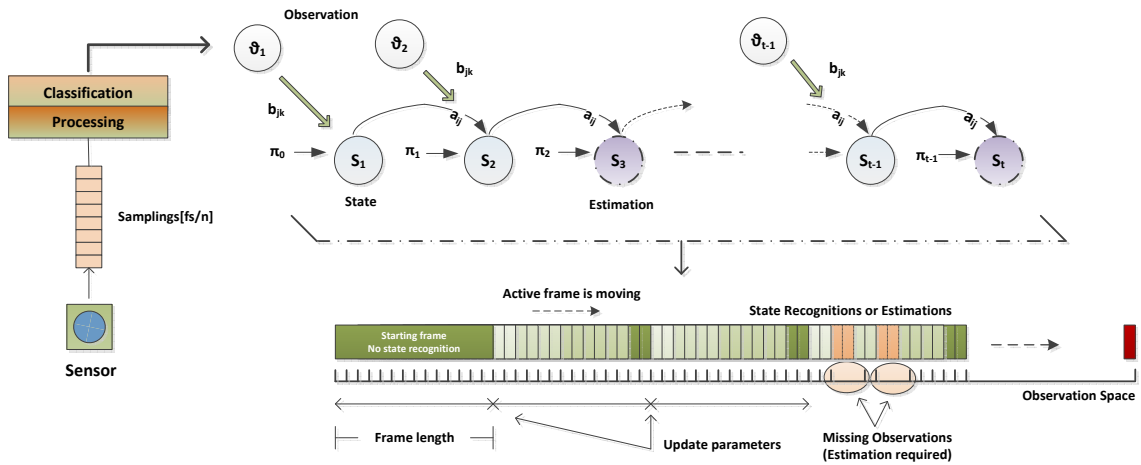


Figure 4.1: Operation of the proposed framework

nal processing operations and eventually end up with a classification algorithm in order to provide desirable inferences about user behaviors/profiles for context-aware applications. A required classification algorithm differs in terms of explanation of extracted user context through a specific sensor. Outcomes of the algorithm are represented in a matrix whose elements show probability weights for possible user state selections. Finally, a selection criteria is set to differentiate between user states.

Classification algorithms produce observations (i.e., *visible states*), ϑ , of HMM. Among observations, only one observation is expected to provide the most likely differentiation in selection of instant user state representation. This observation is marked as instant observation, which also indicates the most recent element of observation sequence of HMM. On the other hand, user states are defined as *hidden states*, S , of HMM since they are not directly observable but only reachable over visible states. Therefore, each observation has cross probabilities to point any user state. These cross probabilities build an emission matrix, b_{jk} , which basically defines decision probabilities of picking user states from available observations.

In addition, a user state might not be stationary since a general user behavior changes in time. Thus, it is expected from a user state either to transit into another user state or to remain same. These occurrences build a time-variant user state transition matrix, a_{ij} , which defines transition probabilities among the user states.

Note that one sensor model is considered for the sake of generalization. In reality, to recognize a user state, it might require multiple sensory observations.

4.2.1 Basic Definitions

First order HMM is proposed for creating a statistical tool to show dependencies of states at discrete time t that are influenced directly by a state at discrete time $t - 1$ ¹ [141, 142]. Discrete time is used to specify sensor readings which occur periodically in a system.

HMM is characterized by the following elements:

- *Hidden Process* : A set of hidden states is defined as a discrete time process with a finite space of N ,

$$S_{[1:t]} = \{S_1 = i, S_2, \dots, S_t\}, \quad \forall i \in \{1, \dots, N\}.$$

Recall that the system can revisit a user state at different system times and also not every user state needs to be visited.

- *Initial Hidden State Probability* : An irreducible and aperiodic Markov chain that begins with its ergodic distribution:

$$\pi_i = Pr(S_0 = i), \quad \begin{cases} \forall i \in \{1, \dots, N\}, \\ \pi_i \geq 0, \\ \sum_{i=1}^N \pi_i = 1. \end{cases}$$

- *State Transition Probability* : a is described as $\{N \times N\}$ state transition matrix where each element a_{ij} of a is equal to a transition probability from state i to state j ,

$$a_{ij} = Pr(S_t = j | S_{t-1} = i), \quad \begin{cases} \forall i, j \in \{1, \dots, N\}, \\ a_{ij} \geq 0, \\ \sum_{i=1}^N a_{ij} = 1. \end{cases}$$

There is no requirement that transition probabilities must be symmetric ($a_{ij} \neq a_{ji}$) or a specific state might remain the same in succession of time ($a_{ii} = 0$).

¹In an n -th order Markov process, state at t has dependencies with states at $\{t - n, t - n + 1, \dots, t - 1\}$.

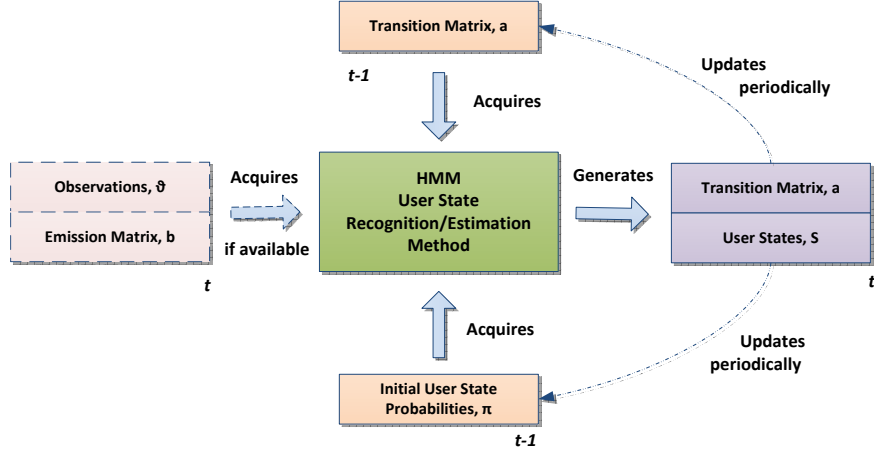


Figure 4.2: User state recognition/estimation method

- *Visible Process* : A set of observations is defined as a discrete time process with a finite space of K ,

$$\vartheta_{[1:t]} = \{\vartheta_1 = k, \vartheta_2, \dots, \vartheta_t\}, \quad \forall k \in \{1, \dots, K\}.$$

- *Observation Emission Probability* : b is described as $\{N \times K\}$ observation emission matrix where each element b_{jk} of b is equal to a cross probability between hidden state and emitted observation,

$$b_{jk} = Pr(\vartheta_t = k \mid S_t = j), \quad \begin{cases} \forall j \in \{1, \dots, N\}, \\ \forall k \in \{1, \dots, K\}, \\ b_{jk} \geq 0, \\ \sum_{k=1}^K b_{jk} = 1. \end{cases}$$

HMM parameters are usually denoted as a triplet $\lambda = \{a_{ij}, b_{jk}, \pi_i\}$.

4.2.2 User State Representation Engine

User state representation engine infers an instant user behavior in light of prior knowledge of a human behavior pattern and availability of sensory observation at decision time. If sensory observation exists, the applied process is called recognition method; otherwise, estimation method.

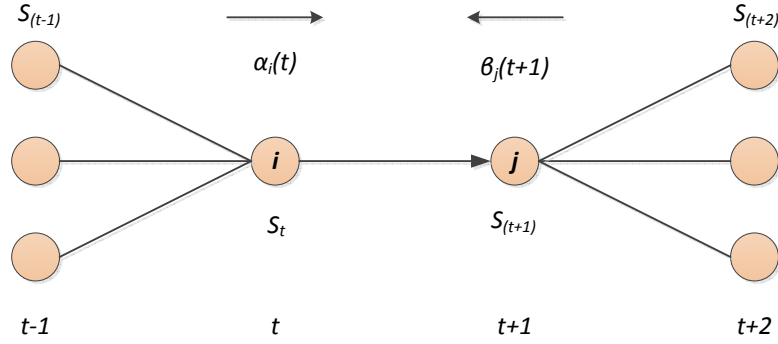


Figure 4.3: Forward-backward algorithm

In other words, estimation method is applied whenever power efficiency is taken into consideration by a system, seen in Figure 4.2.

The probability of a specific observation sequence is given by

$$\begin{aligned}
 Pr(\vartheta | \lambda) &= \sum_{i=1}^N Pr(\vartheta | S, \lambda) Pr(S | \lambda) \\
 &= \sum_{i=1}^N \pi_0 \prod_{t=1}^T Pr(\vartheta_t | S_t) Pr(S_t | S_{t-1}) \\
 &= \sum_{i=1}^N \pi_0 \prod_{t=1}^T a_{ij} b_j(\vartheta_t = k)
 \end{aligned} \tag{4.1}$$

By implementing Bayes' theorem together with (4.1), a sequence of hidden states can be formulated by

$$Pr(S | \vartheta, \lambda) = \frac{Pr(\vartheta | S, \lambda) Pr(S, \lambda)}{Pr(\vartheta, \lambda)} \tag{4.2}$$

Then, inference of a hidden state at time \$t\$ in (4.2) is given by

$$Pr(S_t = i | \vartheta_{[1:\tau]})$$

where \$\tau = t - 1, t\$ or \$T\$ is termed as "predicted", \$\mathcal{P}\$, "filtered", \$\mathcal{F}\$, and "smoothed", \$\mathcal{S}\$, probabilities of \$S_t\$, depending on observation sequence of \$\vartheta_{[1:t-1]}, \vartheta_{[1:t]}\$ or \$\vartheta_{[1:T]}\$ respectively.

The filtered probabilities of \$S_t\$ have already been named user state recognition method in this chapter. Thus, the relevant probability of an instant user state recognition can be found by either evaluation method or decoding.

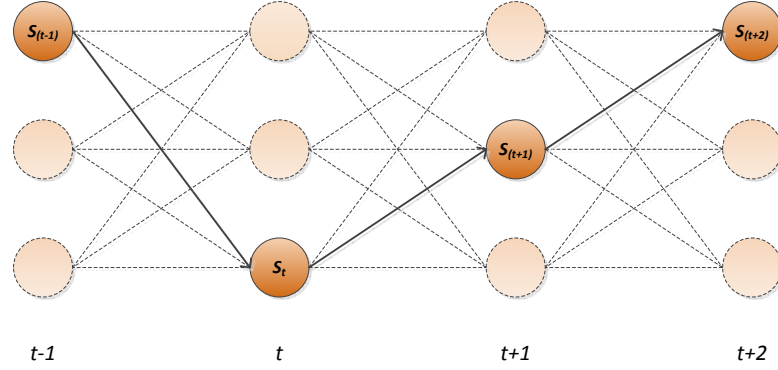


Figure 4.4: Viterbi algorithm

Evaluation method, a.k.a. forward and backward algorithms [140], see Figure 4.3, is proposed to find the most likely one-step ahead or one-step back user state in a hidden chain. Evaluation method relies on updating probability weights iteratively that decides which user state is selected to represent instant user activity.

The corresponding probability weights are α for forward iterations and β for backward iterations. α_t is the probability of current occurrence of a user state, S_t , generated from one-step previous occurrence, S_{t-1} . On the other hand, with reverse analogy, β_t is defined as the probability of transition occurrence of S_{t-1} from S_t .

From inferences of (4.2) since it has inductive calculations by using ϑ_t , S_t and S_{t-1} each term, α and β are expressed as follows:

- $\alpha_t = \sum_j \alpha_{t-1} a_{ij} b_j(\vartheta_t = k)$ with condition of $\alpha_0 = 1$ for initial state, π_0 at $t = 0$.
- $\beta_t = \sum_j \beta_{t+1} a_{ij} b_j(\vartheta_{t+1} = k)$ with condition of $\beta_T = 1$ for final state, π_T at $t = T$.

The term of $a_{ij} b_j(\vartheta_t = k)$ represents a user state transition triggered by given observation.

Since recognition process of user states evolves in real time by context-aware applications, forward algorithm is employed to assign a proper user state in order to specify current user activity whenever a new observation is made.

$$\hat{S}_T = \arg \max_{1 \leq i \leq N} [\alpha_T] \quad (4.3)$$

selects the most likely instant user state representation with existing probability of $Pr(\hat{S}_T) = \max_{1 \leq i \leq N} [\alpha_T]$.

Table 4.1: Filtering user states while no observation received

Filter at $t - 1$	$Pr(S_{t-1} = j \vartheta_{t-1})$
	\downarrow
Prediction at t	$Pr(S_t = j \vartheta_{t-1})$
	$\downarrow \vartheta_t$
Filter at t	$Pr(S_t = j \vartheta^t)$

User state estimation method uses predicted probabilities. Therefore, a posteriori probabilities need to be derived from given existing system parameters, see Table 4.1 and Figure 4.5.

Predicted probabilities are found by

$$\begin{aligned}
 \mathcal{P}_t(j) &= Pr(S_t = j | \vartheta_{[1:t-1]}, \vartheta_t) \\
 &= Pr(S_t = j, \vartheta_t = k | \vartheta_{[1:t-1]}) \\
 &= \sum_j a_{ij} \mathcal{F}_{t-1}(i) \sum_k b_j(\vartheta_t = k) \\
 &= \sum_j a_{ij} \mathcal{F}_{t-1}(i)
 \end{aligned} \tag{4.5a}$$

where $\mathcal{F}_t(i) = \alpha_t(i)$ is taken.

Alternatively, $\mathcal{P}_t(j)$ can be found by assigning the most likely visionary observation instead by accepting there is a missing observation.

$$\mathcal{P}_t(j) = \sum_j a_{ij} \mathcal{F}_{t-1}(i) b_j(\vartheta_t = k_{predicted}) \tag{4.5b}$$

$\mathcal{P}_t(j)$ can be extended as $\mathcal{P}_t(j, k)$ to show a dependency on future predicted observations.

$$\mathcal{P}_t(j, k) = \sum_j a_{ij} \mathcal{F}_{t-1}(i) b_j(\vartheta_t = k) \tag{4.5c}$$

Then, the most likely observation is selected according to assigning each possible observation as a final node to observation sequence as indicated in (6.11) while calculating probability of sequence existence

$$(\vartheta_t = k_{predicted}) = \arg \max_k \sum_j \mathcal{P}_t(j, k) \tag{4.6}$$

with probability of $Pr(\vartheta_t) = \max_k \sum_j \mathcal{P}_t(j, k)$.

4.2.3 System Adaptability

The most important feature of context-aware applications is being capable of adapting themselves to user behaviors. User context differs in time and the corresponding user state also does. Since user behavior shows various patterns from one user to another, a sequence of user states for each user will be arranged in a different formation with respect to variant user state transitions occurring throughout time. For instance; one user might remain in same user state for a long time; whereas, others might be more active by changing their user states frequently. Therefore, it cannot be expected from relevant user state transition matrix to remain stationary. User state transition matrix evolves in time and must adapt itself to user choices.

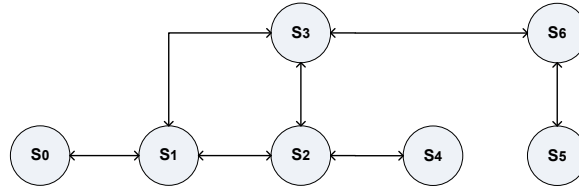


Figure 4.6: An example of user state transitions

4.2.3.1 Time-Variant User State Transition Matrix

User state transitions can be represented as simple random walk on a graph. A finite, simple and undirected graph is a finite collection of vertices, V , and a collection of edges, E , where each edge connects two different vertices and any two vertices are connected by at most one edge [141, 142], such as the graph shown in Figure 4.6.

Let's notate $v_1 \sim v_2$ as vertices v_1 and v_2 are adjacent or an edge connects the two vertices.

User states in HMM are vertices of the graph. The chain chooses a new user state randomly from among neighbor user states adjacent to current user state.

Transition matrix for this chain is given by

$$a_{ij} = \frac{1}{d(v_i)}, \quad v_i \sim v_j$$

where $d(v_i)$ is the number of vertices adjacent to $d(v_i)$. For example, if $d(v_i)$ is 0, a_{ii} becomes 1.

This transition matrix is a default system parameter which is predefined in application for all users. Starting from that point, transition matrix will be evolving in time in response to user behaviors/profiles.

4.2.3.2 Time-Variant Emission Matrix

The system always receives an updated version of the emission matrix at each discrete time when a new context is conceived thanks to the resulted classification algorithms. Therefore, the emission matrix always includes an active condition of cross-probabilities among user states and observations.

4.2.3.3 System Parameters Updating

The system has to update the user state transition matrix periodically according to obtained observations and recognized/estimated user states. There is no finite solution to find an optimal set of updated a and b . Boltzmann method or numerous weight adjustments methods can be found in literature for this purpose. In this study, a generalized expectation maximization algorithm is considered for updating process, which is a.k.a. forward-backward algorithm or Baum-Welch method [140, 144]. This method relies on updating weights iteratively to estimate new system parameters.

The process of how weights are calculated is explained and described in Section 6.2.1.3. With the combination of α and β , system parameters can be estimated. For this purpose, γ_t is defined to present the probability of transition from S_{t-1} to S_t in the presence of observation sequence, ϑ_t .

$$\gamma_t = \frac{\alpha_{t-1} a_{ij} b_{ij} \beta_t}{Pr(\vartheta_t | S_t, \lambda)} \quad (4.7)$$

Updated version of elements belonging to user state transition matrix is found using (4.7), which is defined by division the expected number of transitions from S_{t-1} to S_t at any time to total expected number of transitions from any user state to S_t :

$$\hat{a}_{ij} = \frac{\sum_{t=1}^T \gamma_t}{\sum_{t=1}^T \sum_k \gamma_t} \quad (4.8)$$

(4.8) is repeatedly calculated until the difference between newly estimated a_{ij} and current a_{ij} is less than a predetermined converge criteria. However, this algorithm runs only once when

needed in this study because it consumes considerable computational power. Plus, an update for system parameters is invoked regularly to be able to satisfy adaptation to user behaviors faster; therefore, a_{ij} would converge in anyway sooner or later.

4.2.3.4 Entropy Rate

Entropy is a measure of uncertainty in a stochastic process. It quantifies the expected value of information contained in a specific realization of any random variable. The value of entropy rate shows predictability of a random distribution. As long as entropy rate gets closer to 0, outcomes of distribution become more predictable. In this study, entropy rate is used to track the user state transition matrix. Since the transition matrix is set fixed as default for all users, the system has to adapt to unique user behaviors in time by tracking changes in transition matrix. When entropy rate converges into a stable value, it points out that the system could manage to set required adjustment on adaptation [145, 146].

Entropy rate for a HMM is given by

$$\begin{aligned}
 H(S_t) &= \lim_{t \rightarrow \infty} H(S_t | S_{[1:t-1]}) = \lim_{t \rightarrow \infty} H(S_t | S_{t-1}) \\
 &= \sum_i H(S_t | S_{t-1} = i) Pr(S_{t-1} = i) \\
 &= \sum_i \left(- \sum_j a_{ij} \log a_{ij} \right) \pi_i = - \sum_{ij} \pi_i a_{ij} \log a_{ij}
 \end{aligned} \tag{4.9a}$$

With time-variant property, entropy rate becomes

$$H(S_T) = \frac{1}{T} \sum_{t=1}^T H(S_t) \tag{4.9b}$$

in time sequence of T.

4.2.3.5 Scaling

Computational complexity while calculating system parameters causes a crucial underflow problem. When time goes by during evolution of HMM chains, both $\alpha_t(i) \xrightarrow{t} 0$ and $\beta_t(i) \xrightarrow{t} 0$ starts to head to zero at an exponential rate since a_{ij} and b_{jk} include elements being lower than 1.

Therefore, both variable needs to be scaled.

$$\hat{\alpha}_{(i,t)} = \frac{\alpha_{(i,t)}}{\prod_{\tau=1}^t \sum_i \alpha_{(i,\tau)}} \quad (4.10a)$$

$$\hat{\beta}_{(i,t)} = \frac{\beta_{(i,t)}}{\prod_{\tau=1}^t \sum_i \alpha_{(i,\tau)}} \quad (4.10b)$$

where $\kappa_t = 1/\sum_i \alpha_t(i)$ is defined as a scaling coefficient or a scaling factor in study of [140].

4.3 Simulations

Simulations are carried out in Matlab in order to examine the defined trade-off between power consumption and accuracy in user state representations in light of the proposed framework for context-aware applications. Some case scenarios are created so as to indicate the framework is still valid under different system parameters. For the sake of simplicity, two-user state consisting HMM is considered. However, more complex models can be applied as well by using same system approach.

4.3.1 Preparations

The required HMM is constructed with an initial user state probability, π_0 :

$$\pi_0 = \begin{bmatrix} 0.95 & 0.05 \end{bmatrix} \quad (4.11a)$$

and an initial user state transition matrix, $a_{ij}^{default}$:

$$a_{ij}^{default} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad (4.11b)$$

$a_{ij}^{default}$ is set fixed for all user profiles initially. It is expected from user state transition matrix to adjust itself according to different user profiles so that adaptability problem arisen in context-aware applications could be solved.

In order to claim that system provides adaptability, different user profiles are generated by assigning various probabilities to user state transition matrix, a_{ij} .

$$a_{ij}^{[1:6]} = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}, \begin{bmatrix} 0.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix}, \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix}, \begin{bmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{bmatrix} \quad (4.11c)$$

It can be seen that first five matrices has symmetrical user profile; however, last one is not.

In addition, observation emission matrix is built by

$$b_{jk}^{[1:2]} = \begin{bmatrix} 0.75 & 0.20 & 0.05 \\ 0.05 & 0.20 & 0.75 \end{bmatrix}, \begin{bmatrix} 0.05 & 0.20 & 0.75 \\ 0.75 & 0.20 & 0.05 \end{bmatrix} \quad (4.11d)$$

Given matrices show that three different classification results help a decision process to select a proper user state for instant user activity representation. In other words, same selection criteria is valid every time for which user state is most likely to be chosen in the presence of its cross-relations among observations. For example, given b_{jk} indicates that first and third observations have relatively significant weights for the decision process. During simulation, these matrices remain same in order to make healthy statements on system response to different user profiles.

4.3.2 Process

Figure 4.1 indicates how the proposed system runs. There is a moving frame acting like a queue structure to store new and recent observations. Initially, the frame is being filled with observations until it notifies full. After that, system updates a_{ij} and prepares itself for future operations such as running user state recognition or estimation methods. Since it would cause an insufficient number of observations to select proper user state representations, no operation is executed while initial frame is being filled. Hence, it is not allowed to run energy saving algorithms during initial frame is in progress.

The frame keeps rolling in time by inserting a new instant observation and rejecting the oldest observation simultaneously. Having a frame like queue structure enables to store a short term memory of recent observation history and to preserve redundant waste of computational power. Increase in number of observations yields to put more workload on computations required for applied algorithms during user state representations and update of system parameters. Therefore, length of

Table 4.2: Current consumption vs. data rate in accelerometer, ADXL346

$(V_{DDI/O} = 1.8 \text{ V}, V_S = 2.6 \text{ V})$

Data Rate (Hz)	$I_{DD}(\mu A)$
100	140
50	90
25	55
12.5	40
Autosleep Mode	23
Standby Mode	0.2

the frame is an important system parameter to be set proper. During simulations, this value is set to 60 observations per frame.

System parameters, (4.8) and (4.10), are updated periodically whenever the frame advances as long as its length. This process continues until the system is terminated.

4.3.3 Power Consumption Model

The system makes an observation after a predefined number of sensor samplings are acquired. Then, user state is recognized with the help of extracted context from samplings. Actual sampling time in sensor operation can be extended in order to prolong a mobile device battery lifetime. Sampling intervals are modeled as $I_i = n/f_s$ where n and f_s are an integer value and sampling frequency respectively. I_i defines a waiting time between two consecutive sensor samplings. During simulations, $n = \{1, 2, 4, 8\}$ and $f_s = 100 \text{ Hz}$ are taken.

Given the probability of user states and interval waiting time for a corresponding user state, the expected waiting time for an upcoming sensor sampling is calculated by

$$E[I_t] = \sum_i Pr(S_t = i)(I_t = i) \quad (4.12)$$

Since I_i is considered fixed for all user states throughout simulations, being at any user state does not change a relevant waiting duration to acquire a new sensor sampling.

Reasonably speaking that the expected energy consumption at each sensor sampling is inversely proportional to waiting time (4.12),

$$E[C_t] \propto \frac{1}{E[I_t]} \quad (4.13)$$

For instance; if $n = 1$ is taken for each user state, energy consumption turns out the highest since sensor sampling is being made at each available time slot. Otherwise, whenever $n > 1$, there comes time slots at which no sampling is made.

The least power consuming sensor on today's smartphones is the accelerometer [147]. Therefore, the accelerometer sensor is considered to use in simulations and implementation of a real-time application.

Accelerometer sensor needs to be modeled in order to examine power efficiency achieved at each different sampling strategy. Table 4.2. provides modeling parameters belonging to accelerometer used in application. Drawn current values at specific sampling frequencies give an idea of how much power could be consumed in different sensor operation modes of accelerometer [148].

In this sense, let Ω_{sample} and Ω_{idle} be given as default sensor properties for power consumption per unit time during sensing and during running idle respectively [15]. Note that only drain current value I_{DD} is an effective modeling parameter to determine power consumption through sampling channel under stable drain voltage V_{DD} where power is formulated as $P = VI$ in physics.

Energy consumption during sensing is then formalized as

$$\Theta_{sampling} = \sum_{m=1}^{\#of\ samplings} \delta(m)\Omega_{sample} \quad (4.14)$$

where $\delta(m)$ is Kronecker delta function that indicates sampling is being made.

Also, energy consumption during sensor is in idle mode is given by

$$\Theta_{idle} = \sum_{m=1}^{\# of estimations} \delta(m)\Omega_{idle} \quad (4.15)$$

According to Table 4.2, current flowing at the lowest data rate 12.5 Hz draws roughly 200 times more than current flow in stand-by mode. This information reveals that total energy consumption to acquire samplings is much more greater than the one while the sensor is being idle, $\Theta_{sample} \gg \Theta_{idle}$. Assuming that the sensor becomes idle between two consecutive sampling operations, total energy consumption wasted during a waiting time is ignored.

In summary, in presence of (4.14) and (4.15), the fact that expected waiting time is inversely proportional to number of samplings gives a clue of

$$E[T] \propto \frac{1}{\Theta_{sampling}} \propto \Theta_{idle}$$

Note that computational power required to extract contextual inference from sensor samplings is ignored.

4.3.4 Accuracy Model

The probability of error occurred during either recognition or estimation of user state is calculated by

$$e_t = 1 - Pr(\hat{S}_t) \quad (4.16a)$$

This information yields to find the expected recognition error throughout observation space O with a number of occurrences $\#O$

$$E[e^{rec.}] = \frac{1}{\#O} \sum_{t=1}^O e_t^{rec.} \quad (4.16b)$$

The expected estimation error uses arithmetic mean method. However, it is assumed that t_f is where the first sampling is made. Then, the weight corresponding to an error occurred where the first estimation is made should have lower proportion than the weight corresponding to an error occurred where the last estimation is made because of the fact that while time goes by and an estimation is being made one after another, the accuracy of estimations are expected to degrade. Therefore, the relevant weight for any estimation point is reciprocal of the time distance to next available sampling time slot. In that case, harmonic mean becomes more appealing to finding expected estimation error.

Whenever $n > 1$, the expected estimation error between two consecutive samplings is given by

$$E[e^{est.}] = \frac{1}{\#O} \sum_{t_f=1}^O \left(\frac{1}{I_i - 1} \sum_{t=t_f+1}^{t_{max}=t_f+I_i-1} \frac{t_{max} - t + 1}{e_t^{est.}} \right)^{-1} \quad (4.16c)$$

Finally, accuracy can be found using (4.16) by

$$\varphi\% = \left(1 - (E[e^{rec.}] + E[e^{est.}])\right) * 100 \quad (4.17)$$

4.3.5 Setups

A waiting time for two consecutive samplings is set to $\{n_1, n_2\} = \{1, 2, 4, 8\}$ where $\{n_1, n_2\} \propto I$ with respect to which user state is represented, S_1 or S_2 . That a pair of (n_1, n_2) is selected as $(1, 1)$ means there is no power saving method applied. In contrast, selection of $(8, 8)$ means almost 87.5% of observations inside of the moving frame mentioned in Section 4.3.2 will be estimated according to (4.6).

Total observation number is set to 3600. Recall that the length of the moving frame has been already set to 60. According to which user profile is chosen (i.e., a specific a_{ij}), user states are found by Viterbi algorithm (4.4) with already synthetically generated Markovian observations. These user states are marked as original user states. While the proposed framework is operating, original user states are compared to the user states (4.3) obtained by recognition or estimation methods for the sake of accuracy (4.17).

A simulation starts for each user scenario with default system settings (4.11). In addition, all possible power saving methods are applied repeatedly on observation sequences as long as the proposed system runs and finishes a process of finding proper user state representations.

4.3.6 Discussions

According to simulation results, Figure 4.7 shows how entropy rate could give a clue of identifying a specific user profile, and Figure 4.8 also shows how applied power saving methods affect convergence of entropy rate and accuracy of instant user state representations.

Firstly, adaptability to any user profile is considered in form of convergence of entropy rate. As stated in Section 4.2.3.4, entropy rate (4.9) is calculated with transition matrix, a_{ij} . Transition matrix which is initially fixed for all user profiles gets adjusted to an assigned user profile when time goes by. With the explanation of entropy rate, small difference in entropy rate reveals high prediction of how much closer transition matrix gets into relevant user profile. Therefore, if entropy rate converges into a specific value, which means time-variant user state transition matrix becomes stationary (i.e., time-invariant). In this sense, Figure 4.7 shows how the system achieves adaptability

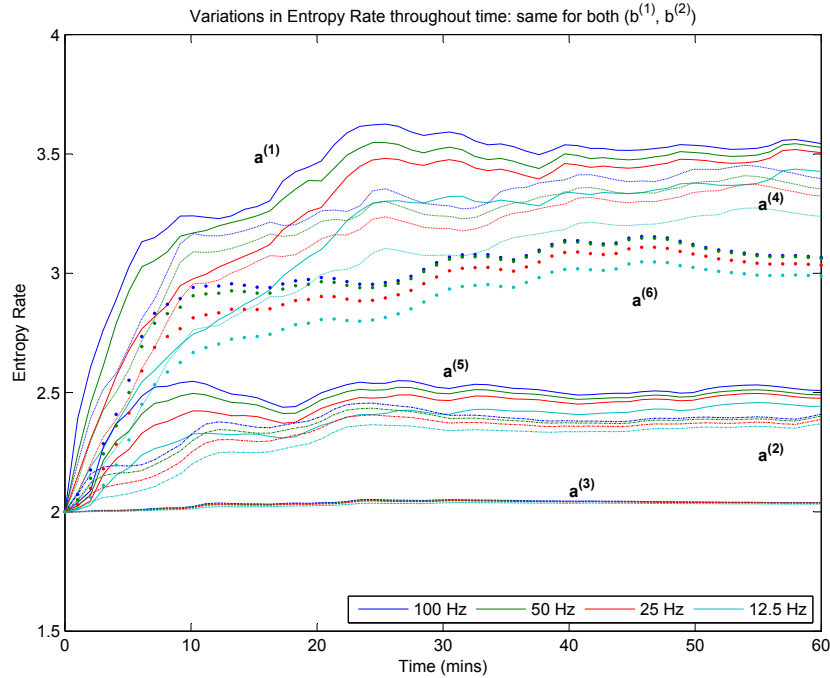


Figure 4.7: Simulation: entropy rate vs. variant user profiles

to different user profiles under variant power saving methods. Discussions on this graph can be made as follows:

- Symmetrically constructed user state transition matrices and their horizontal-flipped forms draw almost similar trajectories with respect to any change in convergence of entropy rate, ($a^{(1)} \approx a^{(4)}$) and ($a^{(2)} \approx a^{(5)}$).
- Observation emission matrices and their horizontal-flipped forms respond in a similar way to updating process of user state transition matrix, ($b^{(1)} \approx b^{(2)}$).
- Entropy rate can be very distinctive factor to identify different user profiles.
- Frequently changing user states like $a^{(3)}$ has the lowest entropy rate since it is for sure that the frequentness of user state transitions will be more predictable.
- Aggressive sampling method (i.e., sampling at 100 Hz) helps user state transition matrix to adapt itself to relevant user profile very quickly and produces clear results to track where entropy rate converges. In contrast, decrease in sampling frequency causes slower adaptability and late convergence.

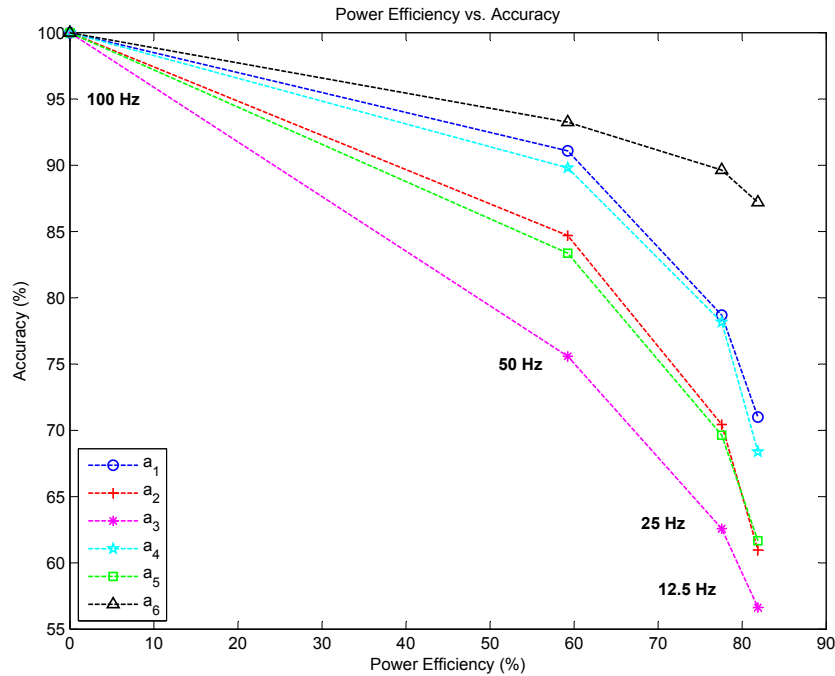


Figure 4.8: Simulation: power consumption vs. accuracy

Secondly, accuracy of user state representations at different sampling methods is examined for all user profiles. Figure 4.8. indicates accuracy rate versus earned power efficiency by sampling methods. The following assumptions can be evaluated from this graph:

- Aggressive sampling achieves the most accurate user state selections; whereas, this method consumes the highest power.
- Other sampling methods elevate power efficiency while satisfying a reasonable accuracy.
 - Sampling at 50 Hz results in power efficiency of 60% in return for an accuracy range from 75% up to 96%, depending on user profiles.
 - At 25 Hz, power efficiency goes up to 78%; however, accuracy downs to from 64% up to 91%.
 - Finally, the slowest sampling method at 12.5 Hz hits up the highest power efficiency by 82% while accuracy falls in a range from 55% up to 87%.

- The lowest accuracy is obtained by the third user profile $a^{(3)}$ since it has equal probability for both user states. In that case, the system becomes uncertain when it comes to a point of decision on proper user state selection.
- Interestingly, symmetrically constructed user state transition matrices and their horizontal-flipped forms have same accuracy model versus power efficiency, $(a^{(1)} \approx a^{(4)})$ and $(a^{(2)} \approx a^{(5)})$.
- Asymmetrically constructed user state transition matrix gives the best solution to the trade-off by achieving accuracy of over 90% for all sampling methods.
- The highest adaptation the system gets to a specific user profile, the better accuracy the system achieves on user state recognitions.

4.4 The Concept Validation Through a Smartphone Application

A real-time application is implemented in order to demonstrate the effectiveness of the proposed framework. Blackberry RIM Storm II 9550 smartphone is chosen as target device. Blackberry Java 7.1 SDK is used for programming language and Eclipse is used as software development tool.

Storm II consists of 3-axis accelerometer named ADXL346 from Analog Devices. This sensor operates at 2.6 V as supply voltage and 1.8 V as drain voltage. Its current flow varies under different operation modes and data rate selections (refer to Table 4.2). Turn-on and wake-up times are determined by data rate, they are approximately defined by $\tau + 1.1$ in milliseconds, where $\tau = 1/(\text{data rate})$.

4.4.1 Observation Analysis

Observation analysis is carried out through a smartphone accelerometer sensor. In Figure 4.9, three-axis acceleration information for postural movements, such as sitting, standing, walking and running, is plotted. Like applied in simulations, only two-user state scenario is taken. These are sitting and standing. Inference related to walking and running is considered as standing since walking and running activities cause more variations over acceleration signal belonging to standing activity.

To differentiate one activity to another, so many studies can be found in literature (refer to Chapter 2). chapters proposing on-line solutions mostly use decision tree based classifications

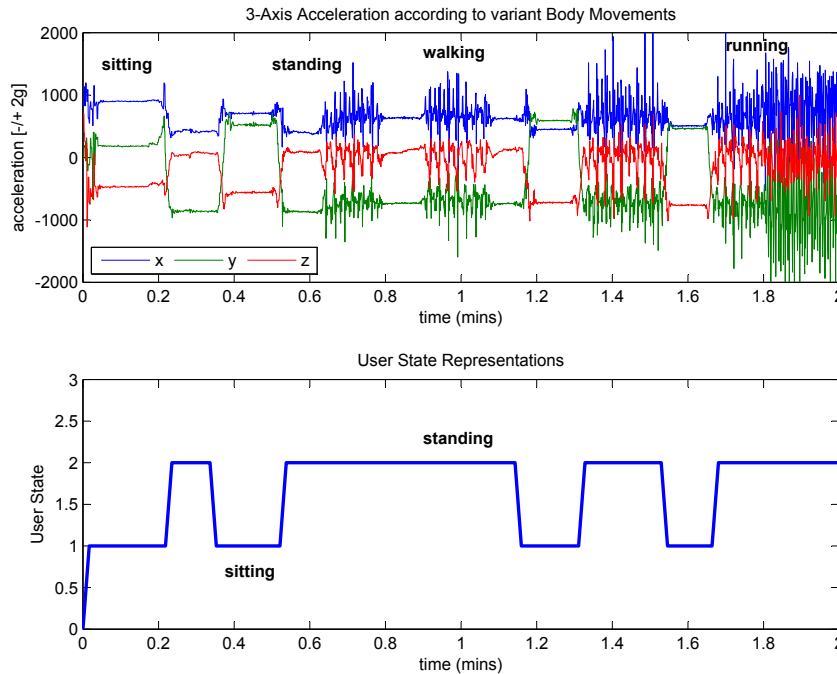


Figure 4.9: Three-axial accelerometer signals: the two-user-state case

with setting predefined fixed thresholds in order to cluster different states. On the other hand, some chapters provide off-line solutions after recording relevant observation data. These solutions propose a creation of high-dimensional feature vectors at first by applying many signal processing techniques in order to extract signal characteristics such as energy, entropy, mean, rms, variance etc. in regardless of analyzing signal deeper in terms of what is really supposed to do so that a proper differentiation could be made. Then, feature vectors undergo classification and/or clustering algorithms to match themselves with one of activities.

Nonetheless, applying complex processing techniques require much computational power. Indeed, it may also cause rapid depletion in any mobile device battery. Differentiation of two-user state scenario likewise applied in simulations may seem less expensive method in computation of feature vectors. However, characteristics of observation data must be well analyzed under any condition in order to stay away from redundant computational workload. In this sense, this chapter proposes a combined usage of HMMs and pattern recognition techniques. The applied method is very applicable to work on-line processing in order to actualize the proposed framework.

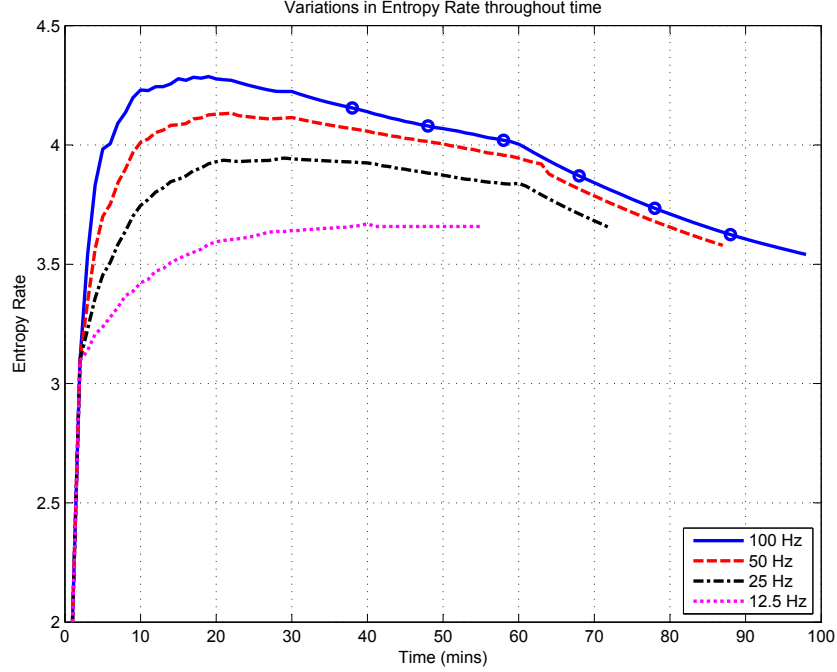


Figure 4.10: Experiment: entropy rate analysis

4.4.1.1 Construction of Observation Emission Matrix

Observation emission matrix b_{jk} is built after an on-line classification method is applied. Applied method is presented in Figure 3.2. Two moving frame structures are used to store both recent history of sensor samplings and recent history of observations. The first frame has a length of 200 sensor samplings. Samplings are acquired by sensors and filled into the frame with a 50% overlap value. Thus, it is aimed that transition from one state to another will not be missed. On the other hand, the second frame has a length of 60 observations.

The recognition process of user state “sitting” and “standing” is explained in detail in Chapter 3; however, the output of this process source the observation emission matrix of the proposed framework as the follows:

$$b_{jk} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \text{prob. of sitting} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \text{prob. of standing} \end{bmatrix} \quad (4.18)$$

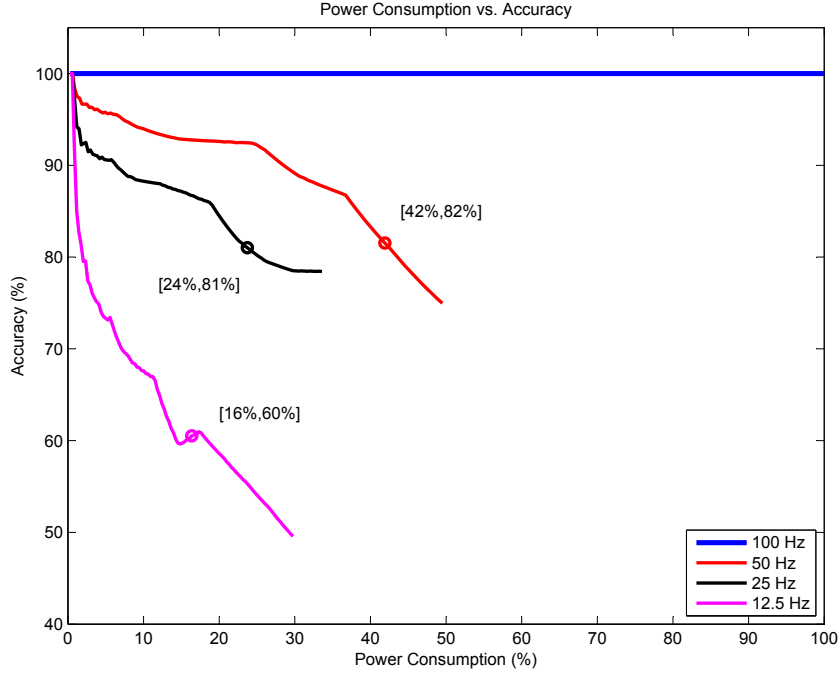


Figure 4.11: Experiment: power consumption vs. accuracy

4.4.2 Process

The system operates likewise how it is explained in Section 4.3.1 and Section 4.3.2. Default system parameters π_0 and $a_{jk}^{default}$ are set initially. The length for the moving frame to store observations is set to 60 observations which equals to one minute in system time. Therefore, system parameters are updated periodically whenever the frame advances as long as its length. Initial time for system construction is set 2 minutes. This duration is basically reserved to let classification algorithms run properly and let system adjust initial adaptability to user. After initial 2 minute long sampling insertions at 100 Hz, any power saving method can be applied. In that case, it takes $\{1, 2, 4, 8\}$ seconds to insert a new observation if $\{100, 50, 25, 12.5\}$ Hz sampling frequencies are adjusted respectively.

Same user activity profile is performed throughout 1-and-half hours application running time for each power saving method.

4.4.3 Discussions

Firstly, Figure 4.10 shows the convergence of entropy rate throughout application running time for each power saving method at different sampling frequencies. Aggressive sampling method

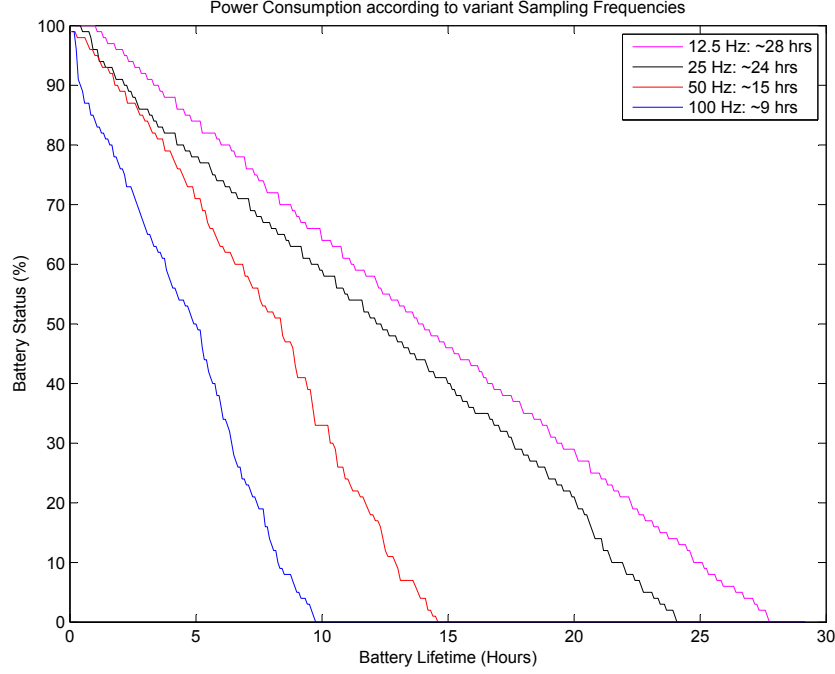


Figure 4.12: Experiment: battery depletion

which takes 100 Hz as sampling frequency draws actual track of entropy rate. Circles over blue line indicate difference in user behaviors. Since user states, such as sitting and standing, are recognized, the frequentness of transition from one state to another cannot be much due to nature of human. Therefore, user state transition matrices over time are desired as

$$a_{ij}^{[1:6]} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.85 & 0.15 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.75 & 0.25 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.6 & 0.4 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{bmatrix}$$

As discussed in simulations, entropy rate converges late while samplings are collected at sampling frequencies except for 100 Hz. That is also the reason why accuracy rate decreases. The most significant result can be obtained from Figure 4.10 is entropy rate cannot sometimes converge into any point where the plot lines belonging to $f_s = \{12.5, 25, 50\}$ Hz stop. When the frequentness of state transitions increases, sampling frequency may not be fast enough to capture the activeness of user profile. Therefore, the system cannot find any proper user state transition matrix to define instant user activity profile. For instance, $a^{(6)}$ has equal probabilities for both residual frequentness in user state and its transition to other user state. Thereby, the specific user activity profile may be enough for being recognized at 100 Hz but not at other sampling frequencies. Note that at 12.5 Hz,

87.5% of observation space is estimated. Another significant point is the blue line on the graph gets closer to $a^{(6)}$ in Fig. 3.a since both transition matrices in application and simulation start to converge into same pattern. Therefore, it is pointed out that the convergence of entropy rate belonging to an HMM based HAR framework could distinguish a specific user activity profile. In addition to that, the linkage between the entropy rate and the accuracy of the recognitions could be made as follows: assume that a simple threshold is defined for entropy rate in (4.2.3.4) by $\varepsilon_{e_p} = [\mu_{e_p} - \sigma_{e_p}, \mu_{e_p} + \sigma_{e_p}]$ where μ_{e_p} and σ_{e_p} are mean and standard deviation of e_p respectively. Hence, an accuracy notifier is set to inform whether or not current recognitions in user states are done accurately by

$$\phi(s, t) = \frac{1}{(t - s + 1)} \sum_{\tau=s}^t 1_{(e_p^\tau \in \varepsilon_{e_p})}$$

Secondly, by using the accuracy notifier, the trade-off between accuracy in user state representations and power consumption is given in Figure 4.11. Aggressive sampling method has the best accuracy value of 100% whereas power consumption has the worst efficiency. Sampling at 50 Hz frequency (red line) provides 82% accuracy against 42% power consumption as long as relevant entropy rate converges. Accuracy rate is over 50%, which means applied HMM works properly and estimates missing observations properly enough. When an user profile has less frequent transitions among user states, accuracy rate is seen over 90%. On the other hand, at a time slot where entropy rate loses information, the sampling method at 100 Hz frequency has consumed almost 87% of its power. Thus, efficiency in power consumption at 50 Hz frequency becomes $(87 - 42) * 100 / 87 = 51\%$. Power efficiency can be healed more if adaptability is satisfied earlier since a drastic fall is seen in the beginning of curves on the graph. In addition, when entropy rate converges in a fixed value, the curves become a concave-shaped because number of false user state recognitions are expected to decrease. After losing entropy rate, the system retains previous user state transition values and continues running. Therefore, a rapid decrease can be seen after a circle is marked on any curve. Other sampling methods has (81%, 67%) and (60%, 70%) pairs of accuracy rate versus power efficiency for 25 Hz and 12.5 Hz respectively where they are applicable. From application results, it can be concluded that if adaptability to user profile is sufficiently satisfied, different sensor sampling methods can be applied adaptively by periodically checking for the convergence of entropy rate. The lower variance value entropy rate gets, the slower sampling period the system can adjust; or vice versa.

Thirdly, Figure 4.12 demonstrates battery lifetimes for each power saving method. The target device has one-year old 1400 mAh battery named DX-1. Note that an used phone is examined in order to depict a regular daily usage. The application starts running with fully charged battery at each sampling method. While the application is running, the device is in stand-by mode and connected to a 3G network. According to the results, the target device battery depletes after approximately {9, 15, 24, 28} hours while operating at {100, 50, 25, 12.5} Hz sampling frequencies respectively. In a stand-by mode, the device itself lasts up to 55 hours. Note that the Blackberry Java 7.1. SDK only reveals remaining battery status information.

In simulations, power consumption per unit sampling at each sampling frequency, which is proportional to (4.13), is modeled as

$$\Theta_{n*12.5Hz} = n * \Omega_{sample} + (8 - n) * \Omega_{idle} \quad (4.19)$$

In order to compare power consumption between simulations and application, the following inequality derived from (4.19) needs to be approximately satisfied.

$$\frac{\Theta_{100Hz}}{\frac{1}{9} - \frac{1}{55}} \approx \frac{\Theta_{50Hz}}{\frac{1}{15} - \frac{1}{55}} \approx \frac{\Theta_{25Hz}}{\frac{1}{24} - \frac{1}{55}} \approx \frac{\Theta_{12.5Hz}}{\frac{1}{28} - \frac{1}{55}}$$

This inequality with a small error margin proves a valid connection between simulations and application in terms of achieved power efficiency. Relevant error stems from computational workload on processor. With the help of this inequality, accelerometer sensor is modeled with sufficient success during simulations. This information reveals that many sensors whose operation method is similar to accelerometer, such as microphone, gyroscope, pulse meter and so forth can be modeled, in a same way which this chapter proposes.

In summary, this chapter presents a generic system framework within the area of mobile device based context-aware applications. This research focuses on the inhomogeneity and the user profile adaptability while examining the trade-off between accuracy in contextual inference from sensory data and required power consumption due to data acquisition and computational processing. The inhomogeneity is characterized by time-variant system parameters, and the user profile adaptability challenge is modeled using the convergence of entropy rate in conjunction with the inhomogeneity. Simulations are run and a smartphone application is implemented in order to demonstrate how entropy rate converges in response to distinctive time-variant user profiles under different sensory

sampling operations. In addition, user state representations are either recognized or estimated while various energy saving strategies are being applied. During the recognition process, a sufficient number of signal processing techniques is applied to find out the best context-exploiting methods on the sensory signal instead of applying computationally harsh pattern recognition methods. Moreover, the power saving consideration is taken at the low-level sensory operations rather than just applying less complexity in computations or changing transferring methods of data packets in the application. In summary, the present research aims at creating and clarifying a HMM based framework to guide the development of future context-aware applications by proposing an effective method to achieve a fine balance in the defined trade-off analysis.

CHAPTER 5

ANALYTICAL MODELING OF SMARTPHONE BATTERY AND SENSORS, AND ENERGY CONSUMPTION PROFILES

The use of mobile devices, such as smartphones, is constrained by battery life. With the ever-increasing computing power and hardware development in mobile devices and slow growth in the energy densities of the battery technologies used in these devices, topics such as extension of battery lifetimes and estimation of energy delivery by the batteries have been one of the primary focuses in the mobile computing research field. Hence, modeling power consumption profiles belonging to new generation mobile devices with the knowledge of the battery behavior will be important for energy optimization and management in resource-constrained mobile computing systems. The battery behavior is not consistent with respect to energy required by the device since the energy drawn from the battery is not always equivalent to the energy consumed by the device itself. In addition, the impact of different usage patterns on devices needs to be investigated to link to the projected effect on the power consumption. By modeling the battery's non-linearities, and understanding the correlation between the usage patterns and the battery depletion leads one to discover suboptimal energy reduction strategies in order to help continual improvement existing within the concept of the context-aware mobile computing.

This chapter studies the battery modeled under the scope of the battery non-linearities with respect to variant discharge profiles. In addition, the energy consumption behavior of some smartphone sensors are analytically modeled including a real time application carried out for the accelerometer sensor to investigate the behavior in detail. Energy consumption profiles are created by assigning different pair of duty cycles and sampling frequencies during the sensory operations. Finally, a Markov-reward process is integrated in order to model the energy consumption profiles and represents the cost by each profile as an accumulated reward. The accumulated reward is also linked to the battery modeling to make a connection between the usage pattern on sensors and the battery behavior.

5.1 Battery Modeling

The energy stored in the batteries is expressed by the product of two quantities which are voltage (*Volt, V*) and capacity (*Ampere – Hour, Ah*). The battery lifetime mostly depends on the energy consumption rate, the discharge profile, e.g., the usage pattern, and the battery non-linearities. At the high energy consumption rate, the effective residual battery capacity degrades resulting in having a shorter battery lifetime. However, any precautionary change in the usage pattern could extend the battery lifetime. More importantly, the physical non-linearities in the batteries could recover the lost capacity while no or a very small amount of energy is consumed.

Battery modeling can help to predict the battery behaviors and then to extend the battery lifetimes. In this regard, many different approaches have been used to model the battery properties. The electro-chemical models [149–151] have been developed to describe chemical processes that take place in the batteries. These models are known as the most accurate models to predict the battery behaviors; however, these models are much complicated to be used in the applications belonging to contemporary mobile devices by including non-linear differential equations as a function of time and position of some battery properties, such as the voltage and current, the potentials in the electrolyte and electrode phases, and reaction rate and current density in the electrolyte. In addition, electrical-circuit models [152] were proposed in order to simulate batteries. These models seem simpler than the electro-chemical models in terms of computational complexity; however, it requires much effort to configure the relevant electrical-circuit model corresponding to a specific battery and also requires a vast quantity of experimental data to better model the battery behaviors. Moreover, analytical models [153] are described to model batteries at a higher level of abstraction instead of the electro-chemical and electrical-circuit models. The major properties of the battery characteristic are given by equations. These models mostly consider batteries in an ideal case, which would have a constant voltage running and a constant capacity profile for every load throughout the discharge. In an ideal case, $L = C/I$ where the lifetime (L) is calculated with a constant capacity load (C) over the running current (I). In constant, the voltage could drop during discharge and the effective capacity observed lower under a heavy usage pattern in reality. Therefore, an approximation is proposed by introducing Peukert's law in [2], which proposes $L = a/I^b$ where a and b are considered as approximately the battery capacity and integer value (e.g., ideally 1) respectively in order of showing non-linearity. However, this Peukert's law would provide the same battery lifetime for all load profiles. The model

would give good results for constant continuous loads, but not for variant or interrupt loads. In this regards, an extended Peukert's law, $L = a / (\frac{1}{L} \int_0^L I_t dt)^b$ is also proposed in the same study to give a new analytical battery model in terms of handling non-constant discharge profiles, i.e., variant loads. Furthermore, stochastic models [154, 155] are presented to model batteries in an abstract manner like the analytical models by discretizing the battery charge and creating probabilistic transition among discharge level caused by variations in workloads. A very detailed study in battery modeling can be found in [1, 4, 156]

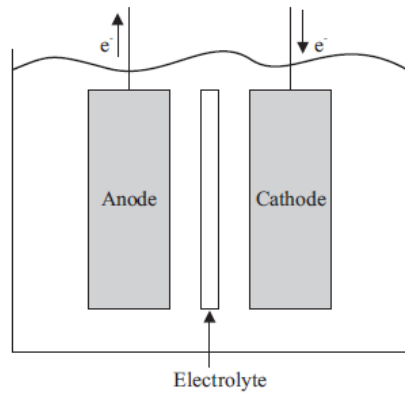
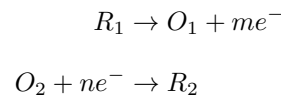


Figure 5.1: A battery cell

A battery consists of electro-chemical cells which drive electro-chemical reactions in order to convert chemically stored energy into electrical energy. A cell includes two electrodes, which are an anode and a cathode, and the electrolyte, which separates the two electrodes, as shown in Figure 5.1. The electrolyte may be liquid as in lead-acid batteries or solid as in lithium-ion batteries. During the discharge, an oxidation reaction occurs at the anode. By this reaction, a reductant (R_1) donates m electrons which become available to be released into the connected circuit. On the other hand, at the cathode, a reduction reaction occurs. By this reaction, m electrons are received by an oxidant (O_2):



Lithium-ion batteries are widely used in notebook computers and cellular phones due to their high energy density and light weightiness. For example, the relevant chemical reactions for a

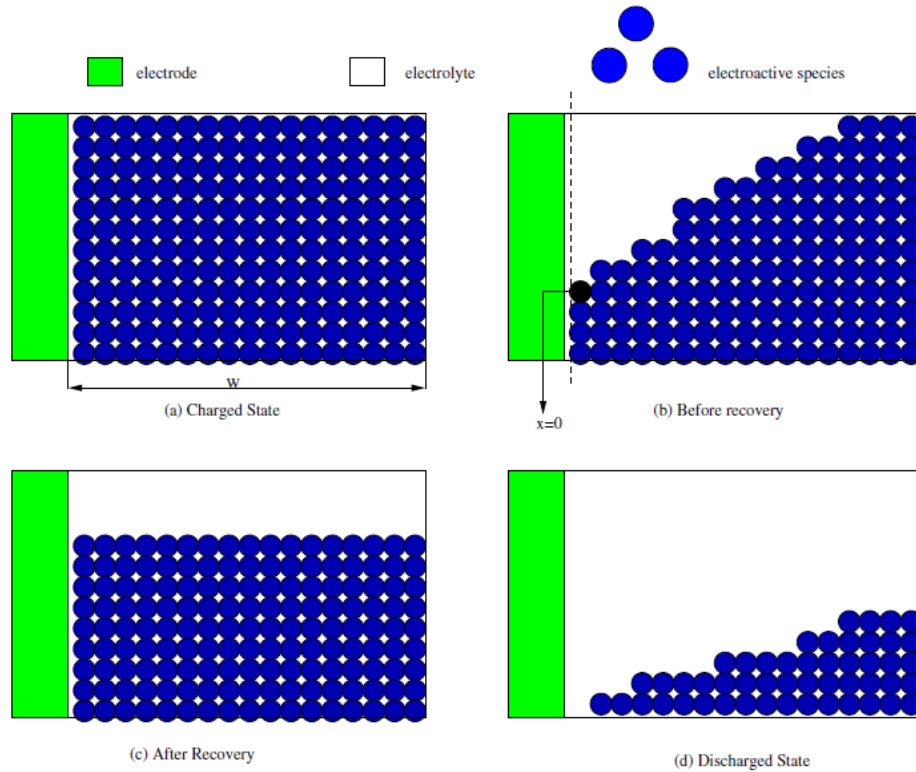


Figure 5.2: Recovery effect (Courtesy of [1–4])

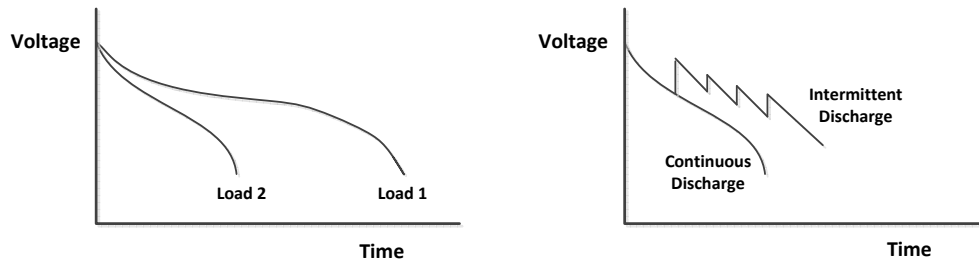
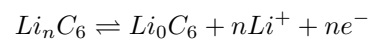
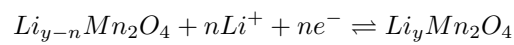


Figure 5.3: Rate capacity effect

lithium-ion battery with the positive electrode ($Li_yMn_2O_4$) and the negative electrode (Li_xC_6) is given in [153, 157] by



The theoretical capacity of the battery is a measure of the maximal charge. The ions at the anode diffuses into the cathode when a current is drawn from the battery. If the current drawn is too high, the speed of diffusion is slower than the rate of the ions reacted at the cathode. This results in reduction occurring at the outer surface of the cathode and access to the inner ions impossible; hence, a drop in the output voltage sourced by the battery is observed. As the intensity of the current drawn increases, the loss of capacity due to the non-uniform deviation in the ion concentration becomes significant, and thereby the cell voltage decreases, which is called *Rate Capacity Effect*, see Figure 5.3. At the lower current drawn or when the discharge process occurs in some intermediate or periodic time scale, the ions could have enough time to diffuse into the inner cathode and let the charge recovery process take place. Therefore, the reacted ions at the cathode become uniformly distributed. This non-linearity is called *Recovery Effect*, see Figure 5.2, which depends on the discharge profile and the time length while no load is applied [158, 159].

Let C and \acute{C} be considered as the theoretical and nominal capacity values of a fully loaded battery. In addition, let $x(t)$ denote the level of available charge and $v(t)$ denote the level of available theoretical capacity of the battery with the conditions of $t \in [0, T]$ and $(v(0), x(0)) = (C, \acute{C})$. According to the Nernst equation in electro-chemistry, the logarithm of electric charge determines the terminal voltage that exists between the pair of electrode terminals, the potential E is determined by $E = E_0 - K_e \ln C$ where C is the concentration of electro-active species at the an electrode, E_0 is the equilibrium potential, $K_e = RT_a/nF$ with R the ideal gas constant, T_a absolute temperature, $n = 1$ the valency of the lithium-ion battery reaction and F Faraday's constant. By Faradays first law, the mass of active material altered at an electrode is directly proportional to the quantity of electrical charge which is transferred at the electrode in the battery reaction. Thus, it identifies the capacity of the cell as a product of C . As a result, the terminal voltage E_t of the cell at time t is given by

$$E_t = E_0 + K_e \ln \tilde{x}(t), \quad 0 \leq t \leq t_0, \quad (5.1)$$

where the normalized and dimensionless $\tilde{x} = x(t)/C \in [0, 1]$ is a measure of capacity, then the battery is full where $\tilde{x}(t) = 1$ and the battery is empty where $\tilde{x} = 0$. Also, t_0 notifies the battery life at which E_t in (5.1) reaches to a cut-off voltage E_c and battery stops functioning. For lithium-ion battery, $E_0 = 3V$ and $E_c = 2V$ where $E_c \leq E_0$.

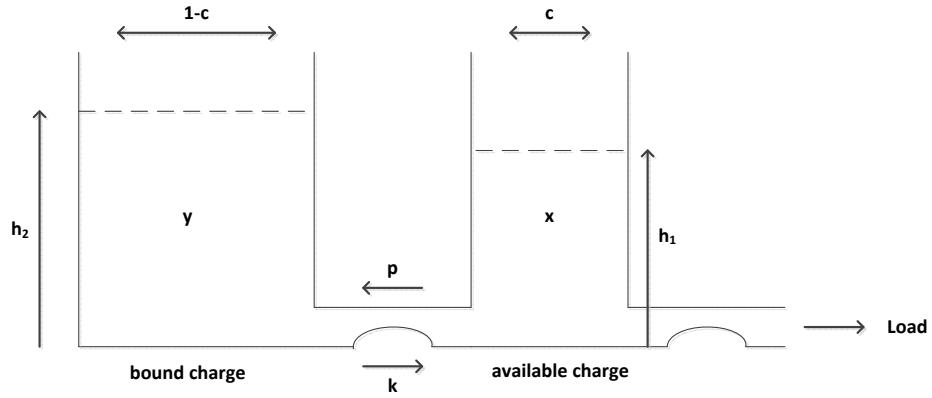


Figure 5.4: The two-well KiBaM

The Kinetic Battery Model (KiBaM) is the most powerful analytical battery model represented firstly in [160] and widely used in recent studies of [156, 161, 162]. The name kinetic is given to the model because of chemical processes taken place during the discharge. KiBaM considers the stored charge to be distributed over two wells, the available-charge well, and the bound-charge well as shown in Figure 5.4. The bound-charge well supplies electrons only to the available-charge well; whereas, the available-charge well supplies electrons directly to the connected load. The rate of charge flow between two wells depends on the conductance parameter k and the difference in heights of two wells h_1 and h_2 . The capacity ratio is denoted as parameter c and corresponds to the friction of total charge stored in the available-charge well. Recall that $x(t)$ denotes the available capacity of the battery and $v(t)$ denotes the total capacity where $x(0) = \hat{C}$ and $v(0) = C > \hat{C}$, so $y(t) = v(t) - x(t)$ becomes the bound charge. On the other hand, the KiBaM was primarily developed to model lead-acid batteries. Since these type of batteries have a flat discharge profile, there are some shortcomings of KiBaM to model lithium-ion batteries, which are widely used in today's mobile devices. However, KiBaM could be still examined under some issues such as the battery lifetime, the capacity rate and the recovery effect [156, 162]. To be able to extend the battery model for lithium-ion cells, a solid state diffusion must be added into KiBaM. In solid state, electro-activated species at the electrodes experience a drift motion diffusion in addition to random diffusion [163]. The drift motion can be added into flux as a negative charge, i.e., degradation on the conductance parameter k .

During the battery operation, the current drawn due to discharge is denoted by $i(t) \geq 0$ with the average discharge rate of $\hat{\lambda} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t i(s) ds$ if the limit exists. Then, the unit change in the charge stored in both wells is given by the differential equations:

$$\begin{aligned} \frac{dx(t)}{dt} &= -i(t) + k(1-p)(h_2 - h_1) \\ \frac{dy(t)}{dt} &= -k(1-p)(h_2 - h_1) \end{aligned} \quad (5.2)$$

with initial conditions of $x(0) = c.C = \dot{C}$, $y(0) = (1-c).C = C - \dot{C}$, $h_1 = x(t)/c$ and $h_2 = y(t)/(1-c)$. Also, $p \in [0, 1]$ denotes a fraction of the flow of charge which has released from and sent back to the bound-charge well. A very detailed study for further equation derivations can be found in [164].

When the battery supplies a continuous discharge, the available-charge well would be reduced rapidly, the difference of the height in both wells would grow since there is no time to move charge from the bounded-charge well into the available-charge well, and the battery would not last long. However, when the load is removed, a remedy charge flows from the bounded-charge well into the available-charge well until h_1 and h_2 are balanced. This gives the idea of why a recover process takes place when an intermittent charge is applied by variant loads.

By using (5.2), the total discharge process becomes independent from the charge flow gradient:

$$v(t) = x(t) + y(t) = C - \int_0^t i(s) ds, \quad t \geq 0 \quad (5.3)$$

By replacing $i(t)dt$ with some measure of $\Lambda(dt)$, then $v(t) = C - \Lambda(t)$. (5.2) turns into

$$\begin{aligned} dx(t) &= -\Lambda(dt) + k(1-p) \left(\frac{y(t)}{(1-c)} - \frac{x(t)}{c} \right) dt \\ dy(t) &= -k(1-p) \left(\frac{y(t)}{(1-c)} - \frac{x(t)}{c} \right) dt \end{aligned} \quad (5.4)$$

The equivalent differential equation solution for (5.4) then becomes

$$\begin{aligned} x(t) &= cv(t) - (1-c) \int_0^t e^{-k(1-p)(t-s)/c(1-c)} \Lambda(ds) \\ y(t) &= (1-c)v(t) + \int_0^t e^{-k(1-p)(t-s)/c(1-c)} \Lambda(ds) \end{aligned} \quad (5.5)$$

If a cell is subject to variant loads, the current drawn would vary with different discharge rates by examining (5.3) accordingly, thereby (5.5) has solutions as follows:

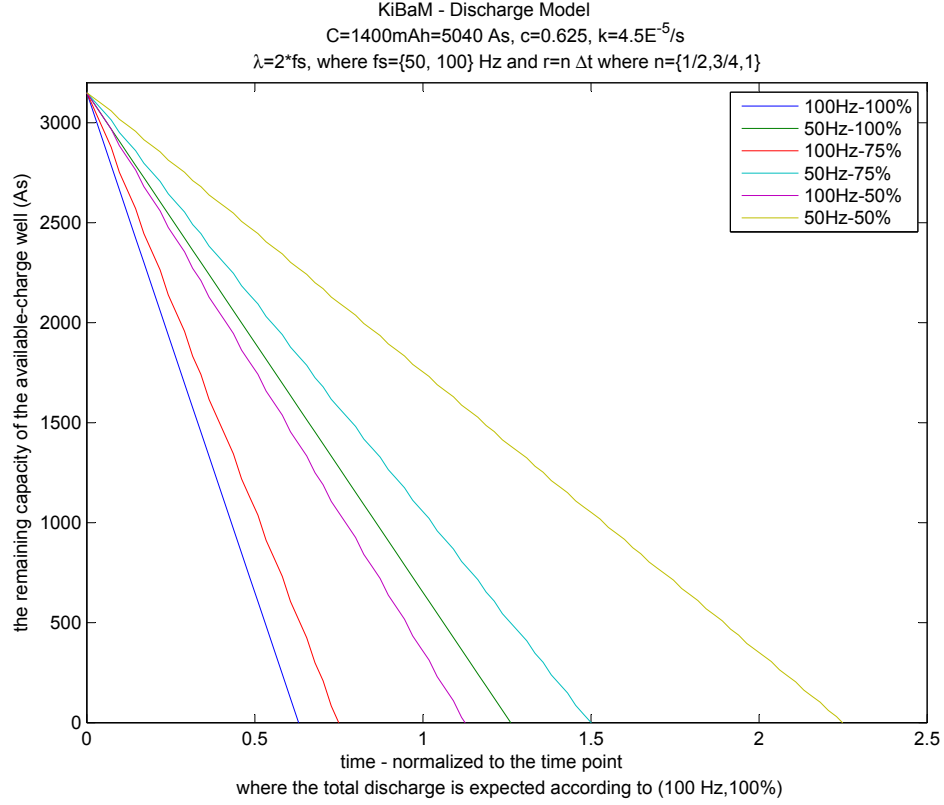


Figure 5.5: The KiBaM discharge model: an example

- *Case 1:* The constant discharge, $\Lambda(t) = \lambda t$, where $v_\lambda(t) = C - \lambda t$ on the bivariate dynamic system of $(v_\lambda(t), x_\lambda(t))$.

$$x_\lambda(t) = cv_\lambda(t) - C_\lambda(1 - e^{-k(1-p)t/c(1-c)}) \quad (5.6)$$

$$C_\lambda = \lambda c(1 - c)^2/k(1 - p)$$

- *Case 2:* The periodic regularly spaced pulsed discharge, $\Lambda(t) = \lambda r \sum_{j=1}^{t/r} \delta_{jr}$, where λr denotes the released charge during the two consecutive pulses in $r > 0$ time intervals on the system of $(v^{(r)}(t), x^{(r)}(t))$.

$$x^{(r)}(t) = cv^{(r)}(t) - (1 - c)\lambda r \sum_{j=1}^{t/r} e^{-k(1-p)(t-jr)/c(1-c)} \quad (5.7)$$

$$= cv^{(r)} - C_\lambda^{(r)}(1 - e^{-k(1-p)(C-v^{(r)})/\lambda c(1-c)})$$

with the evaluation of geometric sum, $C_{\lambda}^{(r)} = \lambda r(1 - c)/(1 - e^{-k(1-p)r/c(1-c)})$ gives the bursting points of the discharge profile due to the periodically drawn load.

Figure 5.5 shows an example investigating KiBaM behavior under different load profiles and at fixed system parameters. The load profiles are characterized by mixture pairs of sampling frequency, f_s and the duty cycling on the load. Thereby, the load is defined by $\lambda = 2f_s$ where $f_s = \{50, 100\}$ Hz, and $r = n\Delta t$ where $n = \{1/2, 3/4, 1\}$. For instance, if $n = 1$, it means the load has a constant discharge profile, i.e., 100% duty cycling; whereas, if $n \in \{1/2, 3/4\}$, it means duty cycle values of $\{50\%, 75\%\}$ are applied on the load. In addition, the same power consumption rate per unit time is considered during discharge. On the other hand, the battery parameters for KiBaM are chosen as in $C = 1400$ mAh, $c = 0.625$, $p = 0.1$ and $k = 4.5E^{-5}$ (1/s). These parameters can differ from one battery to another. However, by analyzing the given example, it is intended to see how differently a battery discharges with respect to variant load profiles.

Figure 5.5 is normalized to the time point where the total discharge is expected according to the full depletion in C where $f_s = 100$ Hz and 100% for duty cycle. According to the results, obtaining the depletion time less than 1 means the battery seems as if it was depleted even if it still has charges in storage. In contrast, the battery life time is extended when the depletion time is greater than 1. As a conclusion, two hypothesis can be made from the example in Figure 5.5. First, the constant aggressive loading by (5.6) affects the discharge profile severely and yields to deplete the device battery faster even if the battery still has a sufficient stored charge. The second hypotheses by (5.7) is that battery recovery effect takes place when load gets lighter. The effect increases the available-charge well and prolongs the battery lifetime.

5.2 The Modeling of Energy Consumption in Sensors

Sensors on mobile devices can be classified into two categories depending on operation methods. The first category consists of sensors, such as accelerometer and microphone, which when turned on to operate can run as long as needed. They need an external command to finish their operations and to be turned off. When sampling operation is in progress, the relevant sampling period in the operation can be adaptively changed if power considerations are taken. On the other hand, the second category consists of sensors such as GPS, Wi-Fi and Bluetooth. When these sensors are turned on, they follow their own dedicated protocols to operate and be in idle position

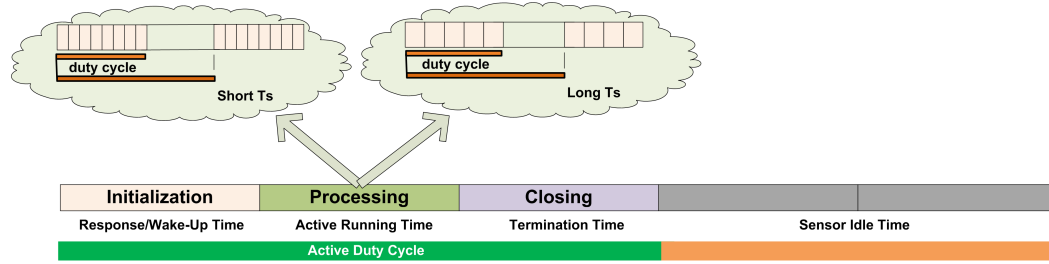


Figure 5.6: Sensory operations

after the required work is done. Therefore, the idea of setting different sampling periods might not be a option for this category. Exceptionally, although Bluetooth is selected in the second category while scanning nearby users around, it might also be considered as a member of the first category as well while its spectrum is being sensed and analyzed with different sampling periods.

The cost of energy consumed during any sensor operation does not only depend on instant energy consumption per operation, but also depends on the duration of operation itself. In Figure 5.6, sensor operation structure for both categories is illustrated. The structure starts with an initialization block which is in charge of waking the sensor up and then waiting for an acknowledge response informing that the sensor is ready. For example, the study in [12] states that for GPS operation, it is required at least 10 seconds to successfully synchronize with satellites. For other sensors, especially the first category sensors, a little time duration would be sufficient to power sensors up and then to set initial system requirements before sampling operations begin. The second block is called processing. This block is dedicated to provide the mentioned efficiency in energy consumption, which this chapter proposes. In this block, sensors start to capture user contextual information and continue to do the same operation repeatedly. The third block terminates sensor's active duty. After all these three blocks are operated, the sensor becomes shut down until a new duty is assigned.

The processing block is where the time length for duty cycle and sampling period are adjusted dynamically for a sensor. Energy consumption is reduced by carefully assigning duty cycles and sampling periods; however, these assignments cause a tradeoff between reduced energy consumption and accurate sensing. If successive sampling intervals are too long, there would not be a sufficient number of samplings to present real conditions, and eventually it would cause user contextual information to be sensed incorrectly. On the other hand, in case where the intervals

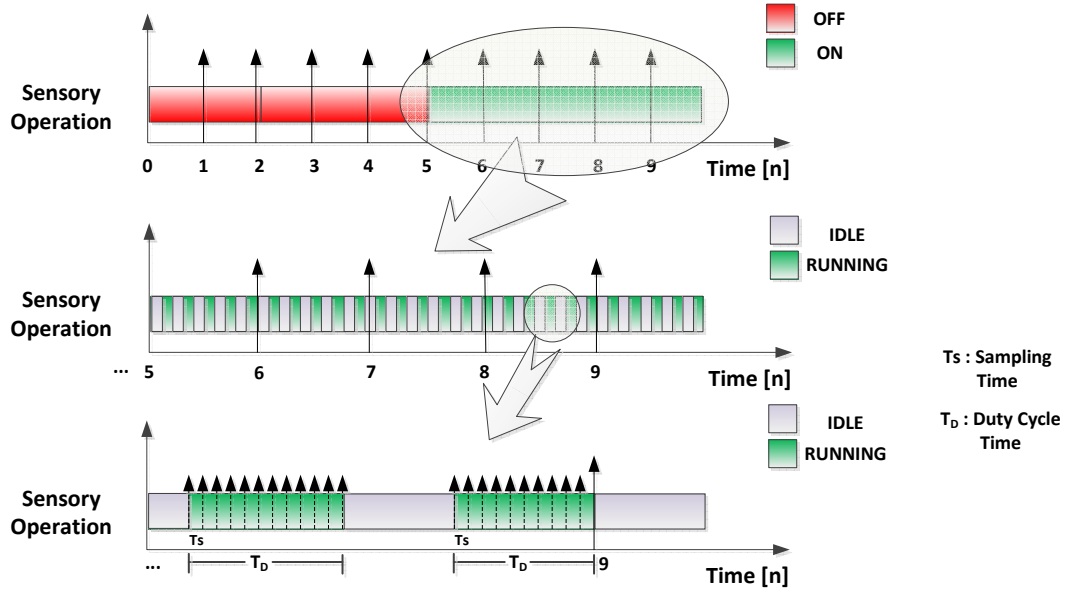


Figure 5.7: Duty cycling and sensor sampling

are too short, the energy saving would not be satisfied. The same approach can also be applied to sensor sleeping time intervals. A longer sleeping time interval would reduce energy consumption; nonetheless, the detection latency will be increased so that false detections could occur.

5.2.1 Preliminaries

Assume that a first category sensory operation is modeled by some certain time parameters, such as t_w , t_s , t_c and t_r are given for a wake-up/initialization time, which could be a termination time as well, a total time per sampling, a total time for operation cycle and a total time throughout sensor running respectively. In addition, Ω_s and Ω_i are given as constant default sensor properties for power consumption to make a sampling and to run idle respectively [15].

DC stands for duty cycle which indicates the active sensor operation (i.e., time interval in which actually sensor samplings are being made) within t_{cycle} . Thus, the number of sampling occurrences is found by $N_s = DC * t_c / T_s$ where $0 \leq DC \leq 1$ and T_s is sampling period (i.e., inversely proportional to sampling frequency, $1/f_s$) which defines a waiting time between two consecutive sensor samplings, see details in Figure 5.7.

Then, a total energy consumption throughout sensor running is approximately given by

$$\begin{aligned}\Theta_{total} &\approx \left[(DC * t_c) \left(\sum_{k=1}^N \left(\sum_{(k-1)T_s}^{(k-1)T_s+t_s} \Omega_s + \sum_{(k-1)T_s+t_s}^{kT_s} \Omega_i \right) \right) + ((1 - DC) * t_c) \Omega_i \right] \left(\frac{t_r - 2t_w}{t_c} \right) \\ &\approx \left[N_s t_s \Omega_s + (t_c - N_s t_s) \Omega_i \right] \left(\frac{t_r - 2t_w}{t_c} \right)\end{aligned}\quad (5.8)$$

while ignoring power consumption wasted during initialization and termination of the sensor.

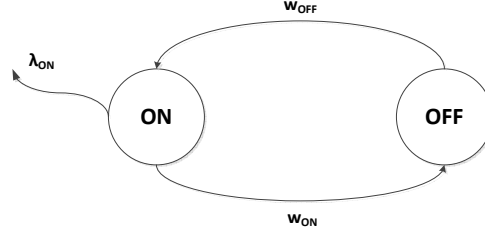


Figure 5.8: Interrupt Poisson Process (IPP)

5.2.2 The Modeling of Sensory Operation

Sensor operation can be modeled as 1-Burst process, also called ON-OFF sensory operation. This process is a special case of Markov-modulated Poisson processes. This special case is called Interrupted Poisson Process (IPP) and has two states: ON and OFF, as shown in Figure 5.8. According to the process, the traffic is Poisson $\{F(x) = 1 - e^{-\lambda x}\}$, but the traffic is generated at deterministic intervals in ON state with a constant rate λ_{on} and a length of mean time, i.e., transition time, $1/w_{on}$. In contrast, there is no traffic in OFF state, and the intensity of residency in this state is given by w_{off} .

The traffic in IPP can be seen as discrete-event simulator for samplings in sensory operations. Thus, the intensity rates become $\lambda_{on} = fs$, $w_{on} = (DC * tc)^{-1}$ and $w_{off} = ((1 - DC) * tc)^{-1}$.

Note that alternatively the burst periods can be modeled by Weibull distribution $\{F(x) = (1 - e^{-\lambda x^\alpha})\}$ where $\alpha > 1$ yields to have more deterministic variations. Then, sampling intervals, i.e., inter-arrival time in traffic, $T = 1/\lambda_{on}$ are generated. In addition, the silence periods can be derived from Pareto distribution, which is known as tail process, $\{F(x) = (1 - (k/x)^\alpha)\}$ where $\alpha > 1$ yields to have more deterministic variates. On the other hand, in case of no silence periods, Erlang-K model, which is stochastically sum of K times *i.i.d.* random variables with each having a Poisson

Table 5.1: The power consumption ratio in the sensor drain per each operation cycle: $t_c = 2s$, and the comparison applied based on (50%, 12.5 Hz)

$(DC(\%), f_s \text{ (Hz)})$	Ratio
(100, 100)	4.45
(50, 100)	2.58
(100, 50)	2.85
(50, 50)	1.80
(100, 25)	1.75
(50, 25)	1.24
(100, 12.5)	1.26
(50, 12.5)	1

distribution, can be considered by setting $\lambda_{on} = 2f_s K$ and the expected frequentness of consecutive on/off times becomes $K/(2f_s K)$.

5.3 A Case Study: A Real-Time Application by The Smartphone Accelerometer Sensor

The least power consuming sensor on today's smartphones is the accelerometer [147]. Therefore, the accelerometer sensor is used in the implementation of a real-time application.

The Blackberry RIM Storm II 9550 smartphone is chosen as the target device for a real-time context aware application. The Blackberry Java 7.1 SDK is used for programming and Eclipse is used as the software development tool. Storm II consists of 3-axis accelerometer named ADXL346 from Analog Devices. This sensor operates at 2.6 V as supply voltage and 1.8 V as drain voltage. Its current flow varies under different operation modes and data rate selections (refer to Table 1.). The drawn current values at specific sampling frequencies give an idea of how much power could be consumed in different accelerometer sensor operation modes. Turn-on and wake-up times are determined by data rate and a relevant formula is given approximately by $\tau + 1.1$ in milliseconds, where $\tau = 1/(\text{data rate})$ [148].

By using IPP sensory modeling together with the data obtained through Table 4.2, and (5.8) with parameter setting of $t_c = 2s$, the power consumption ratio in the sensor drain per each operation cycle under variant DC and f_s values are shown in Table 5.1. According to the results, the smartphone accelerometer sensor consumes 4.45 times more power per each operation cycle in aggressive sampling mode in comparison with the most non-aggressive sampling mode.

The accelerometer sensor needs to be modeled in order to examine power efficiency achieved under different sampling and duty cycling strategies. By using (5.2) and (5.8), the power consumption model within each t_c can be considered as follows:

$$\Theta_{t_c} = \sum_0^{DC*t_c} (L_s(x, y) - R_s(x, y)) + \sum_{DC*t_c}^{t_c} (L_i(x, y) - R_i(x, y)) + \sum_0^{t_c} L_b(x, y) \quad (5.9)$$

- (x, y) is the bivariate dynamic battery system parameters which define capacity values stored inside the bound-charge well and the available-charge well in KiBaM.
- L_s and L_i are denoted as the discharged load when sensor samplings are being made and the sensor is being idle respectively.
 - If $DC = 1$, then the sensor makes samplings continuously. This means the effect of L_i can be ignored due to the constant discharge profile. Thus, recall that the model follows (5.6).
 - If $DC \in [0, 1)$, then the sensor makes samplings in defined intervals. In this case, the effect of L_i cannot be ignored so that the model follows (5.7).
- R_s and R_i are *battery recovery effects* in KiBaM to define the remedy charge flowing from the bound-charge well to the available-charge well when sensor samplings are being made and the sensor is being idle respectively.
- L_b stands for the background load occurring instantly within smartphone operation. For instance, while the application is running, the device is in stand-by mode and only connected to a 3G network. There is no other system functionality being executed while the sensor operates.

The application runs until initially fully-loaded one-year-long used smartphone battery depletes. Only one constant pair of sampling frequency and duty cycle is applied as operation parameters to the accelerometer sensor at each application run. Sampling intervals are modeled as $I_i = n/f_s$ where n and f_s are an integer value and sampling frequency respectively. I_i defines a waiting time between two consecutive sensor samplings; $n = \{1, 2, 4, 8\}$ and $f_s = 100$ Hz are taken. On the other hand, values of $\{0.5, 0.75, 1\}$ are taken for DC . t_c is also taken as 1 sec.

Results are shown in Figure 5.9. Note that the Blackberry Java 7.1. SDK only helps to reveal remaining battery status. According to results, applying a more aggressive sampling methodology

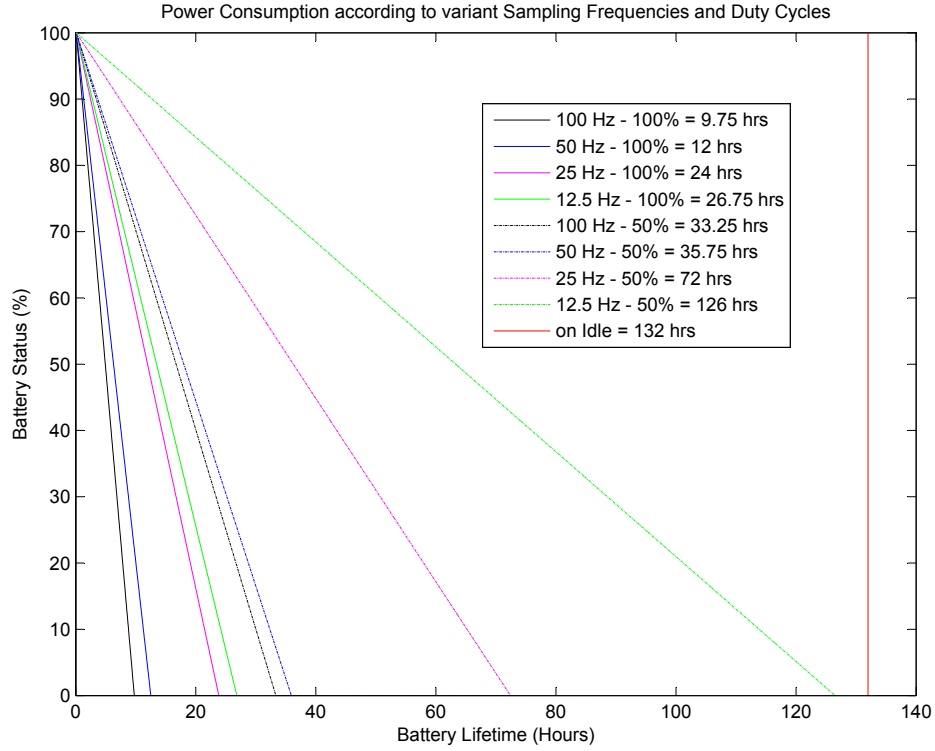


Figure 5.9: The battery depletion due to the accelerometer sensor

causes faster battery depletion. In addition, the lower value of DC makes the battery recover effect more significant and thus prolong the battery lifetime.

According to (5.8), energy consumption per time of ($n_{max}/f_{smax} = 8/100$) under different sampling frequency is modeled as

$$\Theta_{n*12.5Hz} = n * \Omega_{sample} + (n_{max} - n) * \Omega_{idle} \quad (5.10)$$

Then, with the help by (5.10), (5.9) turns into

$$\Theta_{t_c} = \frac{DC t_c}{n_{max}/f_{smax}} (\Theta_{n*12.5Hz} - r_s) + \frac{(1 - DC) t_c}{1/f_{smax}} (\Omega_{idle} - r_i) + l_b \quad (5.11)$$

where r_s , r_i and l_n are considered having a stable energy consumption rate per unit time.

Since the parameters r_s and r_i in (5.11) cannot be verified through smartphone programming, in order to compare energy consumption between the analytical model and the application, the following inequality needs to be approximately satisfied.

$$\frac{\Theta_{100Hz,DC=1}^{real}}{\Theta_{100Hz,DC=1}^{sim}} \approx \frac{\Theta_{50Hz,DC=1}^{real}}{\Theta_{50Hz,DC=1}^{sim}} \approx \frac{\Theta_{25Hz,DC=.5}^{real}}{\Theta_{25Hz,DC=1}^{sim}} \approx \frac{\Theta_{12.5Hz,DC=1}^{real}}{\Theta_{12.5Hz,DC=1}^{sim}}$$

Due to the effects of r_s and r_i are more significant when $DC \neq 1$. Hence,

$$\frac{\frac{1}{9.75} - \frac{1}{132}}{\Theta_{100Hz,DC=1}^{sim}} \approx \frac{\frac{1}{12} - \frac{1}{132}}{\Theta_{50Hz,DC=1}^{sim}} \approx \frac{\frac{1}{24} - \frac{1}{132}}{\Theta_{25Hz,DC=1}^{sim}} \approx \frac{\frac{1}{26.75} - \frac{1}{132}}{\Theta_{12.5Hz,DC=1}^{sim}}$$

The inequalities with a small error margin prove a valid connection between the analytical model and the application in terms of power efficiency achieved. Relevant error would stem from computational workload on the processor and non-linear functionality of the battery. With the help of these facts, the accelerometer sensor is analytically modeled with sufficient success. This information reveals that many sensors whose operation method is similar to accelerometer, such as microphone, gyroscope, pulse meter and so forth, can be modeled in a similar way.

5.4 Sensor Management

In this section, the battery discharge profiles are seen within the concept of Markov Reward process.

5.4.1 Discrete-Time Markov Reward Model

Suppose that an inhomogeneous semi-Markov chain is given as a statistical tool for handling sensor managements. The chain consists of a finite state space $S = \{1, \dots, N\}$, the state transition density matrix Q , and the state transition matrix P where $Q, P \in \mathbb{R}^{N \times N}$. In addition, a reward structure can be attached to this on-going chain, and can be thought of a random variable associated with the state occupancies and transitions. Thus, the reward can depend on the time of entrance in a state defined in this chain. $r(s, t)$ is given where the reward is accumulated from s to t , $t \geq s$.

This process considers "reward" being paid/received at each time that system passes for a given state. Rewards are defined in two different ways. The first, called permanence/rate reward, represents the reward given for the permanence in a state and is denoted by ψ . On the other hand, the second, called transition/impulse reward, represents the reward given for a state transition and is denoted by γ . When the reward increases at a given rate while hanging in a state it is referred to as rate reward accumulation; whereas, when the reward increases instantly at a state transition of the underlying process it is referred to as impulse reward accumulation. The reward processes can be

positive or negative and discounted or non-discounted. Therefore, a variable called interest/discount rate, denoted by ρ , is introduced into the reward processes within the definition of discount function, φ .

The general evolution equation for reward process for inhomogeneous semi- Markov chain is given by

$$V_i(s, t) = (1 - H_i(s, t)) \sum_{v=s+1}^t \psi_i(s, v - s) \varphi(s, v) + \sum_l \sum_{v=s}^t b_{il}(s, v) \varphi(s, v) (\gamma_{il}(v) + V_l(v, t)) \quad (5.12)$$

The LHS represents the expected present value of all received/paid discounted rewards from time s to t given that process enters into state i at time s . Whereas, the first element of RHS represents the rewards in case there is no state transition from the state i to any from time s to t . The second element represents the rewards after a possible state transition occurs.

(5.12) can be redefined for an immediate case where the total reward is an aggregation of rewards earned both at previous time and at current time as the following structure:

$$V_i^{(t)}(s) = V_i^{(t-1)}(s) + \varphi_i(s, t) \sum_l p_{il}^{(t-1)}(s) \psi_l(s, t) + \sum_j p_{lj}(t) \gamma_j(s, t) \quad (5.13)$$

which, i.e., collects the reward by either continuity in the same state or transition to another state.

The interest rate, ρ , is a constant value; however, it can change by implying forward rates, ρ_x . Each time that system is in state i it is paid a reward ψ_i which should be discounted for each epoch at time s . Thus, the paid/received rewards change in time.

$$\varphi(s, h) = \begin{cases} 1, & \text{if } s = h \\ \prod_{x=s+1}^h (1 + \rho_x)^{-1}, & \text{if } s < h \end{cases} \quad (5.14)$$

with the condition of $0 < \varphi(s, h) \leq 1$.

The permanence reward, $\psi_i(s, t)$, may vary in time or set fixed as $\psi_i.(t - s)$.

5.4.1.1 Battery Case

Assume that reward is seen as power consumption per unit time while a mobile device battery is discharging, and states are defined as the battery discharge profiles. Due to the present power consumption being independent from future power consumption in a mobile device discharge

model, the discount factor would be 0. The immediate reward is also ignored because it could be seen as a difference of permanence rewards during a state transition.

The total accumulated reward while residing in state i from time s to t follows the difference equation of (5.13) as in

$$\Delta_t(v_i(s, t)) = \psi_i(t, v_i(s, t)) \quad (5.15)$$

with the distribution of $F^V(t, v) = Pr\{V(t) \leq v\}$.

For modeling a battery, two different rewards could be considered. By recalling (5.2) along with (5.15), the first reward indicates the power consumption in each state i ; whereas, the second reward belongs to degradation in maximum battery capacity, i.e., the recovery effect which makes a remedy charge flow migrate from the bound-charge well into the available-charge well in KiBaM:

$$\begin{aligned} \Delta_t(v_{i,1}(s, t)) &= \psi_{i,1}(v_{i,1}(s, t), v_{i,2}(s, t)) \\ &= I_i - k(1 - p)(h_2 - h_1) \end{aligned} \quad (5.16)$$

$$\begin{aligned} \Delta_t(v_{i,2}(s, t)) &= \psi_{i,2}(v_{i,1}(s, t), v_{i,2}(s, t)) \\ &= k(1 - p)(h_2 - h_1) \end{aligned}$$

where $h_2 > h_1 > 0$ and $v_{i,1}(s, 0) = v_{i,2}(s, 0) = 0$.

The joint distribution of the accumulated rewards for a specific state i in (5.16) becomes

$$F_i^{(V_1, V_2)}(t, v_1, v_2) = Pr\{S(t) = i, V_1(t) \leq v_1, V_2(t) \leq v_2\} \quad (5.17)$$

with boundaries of $\min\{v_1, v_2\} = \{0, 0\}$ and $\max\{v_1, v_2\} = \{c.C + v_2, (1 - c).C\}$ since the battery always has a predefined capacity C , which is distributed by a fraction factor c over two wells in KiBaM.

Therefore, the battery gets empty by evaluating (5.17) when $V_1(t) \geq \dot{C} + V_2(t)$,

$$Pr\{V_1(t) \geq \dot{C} + V_2(t)\} = \sum_{v_1=0}^{\dot{C}+v_x} \sum_{v_2=0}^{v_x} \sum_{i \in S} F_i^{(V_1, V_2)}(t, v_1, v_2) \quad (5.18)$$

where $v_x \leq C - \dot{C}$. If $v_x \neq C - \dot{C}$ when the battery gets empty, it gives a clue that a constant high load was being applied to the battery and that made the recovery effect not to take place.

5.4.1.2 Sensor Utilization Case

Assume that a set of DC and a set of fs are given by $\{1, 0.75, 0.5\}$ and $\{100, 50, 25, 12.5\}$ Hz respectively. In addition, let the state space lie over two sub-spaces, which are sets of DC and fs , $S = \{S_{DC} \times S_{fs}\}$. Thus, the state space is defined by $S = \{S_{\{1,100\}}, S_{\{1,50\}}, \dots, S_{\{0.75,100\}}, \dots, S_{\{0.5,12.5\}}\}$

The corresponding reward for each state can be obtained by using (5.8) and Table 5.1. The power consumption at each fs is proportional to flowing drain current, I_{DD} under a stable drain voltage supply due to $P = VI$. Thus, the reward for staying in a state can be defined as

$$\psi_{\{DC=j,fs=k\},1} = \Theta_{\{DC=j,fs=k\}} / \Theta_{\{(DC=1,fs=100)\}} \quad (5.19)$$

where $0 < \psi_{\{j,k\},1} \leq 1$ since a pair of $(DC = 1, fs = 100)$ is the most aggressive sensor utilization method in terms of power consumption.

The ascending order of $\psi_{\{j,k\} \rightarrow \{i\},1}$, $\forall j \in DC, \forall k \in fs$ according to Table 5.1 defines state characterization: $S_i = \{S_{\{0.5,12.5\}} = 1, \dots, S_{\{1,100\}} = N\}$ where $N = \text{length}(DC) \times \text{length}(fs)$.

The second reward can be found by derivation from (5.2) and (5.6) if a state transits into another state at s

$$\psi_{j,2}(s, t) = \frac{c(1-c)}{k} (\psi_{i,1} - \psi_{j,1}) (1 - e^{-k(t-s)/c(1-c)}) \quad (5.20)$$

where $t \geq s$ and $\psi_{i,1} > \psi_{j,1}$ since the recover effect takes place whenever a lower load is being applied.

This chapter investigates how any change in sensory operations affects the power consumption profiles triggered by the sensor and depletion on the battery. Understanding the nonlinear battery behavior with respect to diverse operation methods in sensors, a tolerable power consumption balance could be achieved while employing context aware services in resource constrained mobile devices.

CHAPTER 6

CONTEXT-AWARE FRAMEWORK: A COMPLEX DESIGN

The ever-increasing technological advances in embedded systems, together with the proliferation of growing development and deployment in MEMS technologies, have enabled smartphones to be re-purposed in order to recognize daily occurring human based actions, activities and interactions which mobile device users encounter in their surrounding environment. Recognizing human related event patterns, called *user states* in this chapter, accurately enough could give a better understanding of human behaviors, and more significantly, could reveal a chance for assisting individuals in order to enhance the quality of human lives. Therefore, the inference of a vast variety of human activities in a computationally pervasive way within a diverse context acquired by a series of sensory observations has drawn much interest in the research area of ubiquitous sensing. However, the evolution of the ubiquitous sensing on the resource-constrained mobile devices have empowered Cyber-Physical Systems (CPS) [120] to emerge as a promising solution for the dynamic integration of highly complex and rich interactions among the modeling computational virtual world and the exploiting heterogeneous and unobtrusive physical world.

By utilizing smartphone sensors within the concept of ubiquitous sensing, it would bring about some drawbacks on the resource-constrained hardware platforms. First, continuously capturing user context through sensory data acquisitions, and then inferring desirable hidden information from the context would put a heavy workload on the smartphone processor and sensors. Thereby, these operations cause more power consumption than the device itself does while a regular run, and the device battery would deplete rapidly. To better understand this issue, an application example in [11] can be examined. Accordingly, the accelerometer sensor built-in HTC Touch Pro is employed at a fixed sampling frequency. When the phone receives data samples from the accelerometer, the overall power consumption on device increases by 370 mW; whereas the accelerometer should consume less than 1 mW while being active according to the data sheet. Even if the accelerometer itself wastes very little power to operate its functionality, the phone, together with its main processor

and other hardware components, causes much more power consumption while being in process to accomplish a contextual sensory data extraction. Another example can also be given by [12]. It is reported that today's mobile devices are not feasible to employ all sensors at the same time by providing an example of Nokia N95 mobile phone with a fully-charged battery. It is experimentally examined that the phone battery would get totally depleted within six hours if GPS is switched on permanently even it is not being actively used; whereas the same battery supports a conventional telephone conversation up to ten hours.

The best energy saving algorithm would consist of powering a minimum number of sensors to recognize any user activity [33, 165]. The algorithm assigns different duty cycles to different sensors so that at any time, depending on an application, there might be some sensors running; others might be shut down. For example, if user stays indoors, it is not necessary to power GPS sensor. In contrast, accelerometer and Wi-Fi might be turned on. Another approach would also consider the duty cycling approach at the low level sensory operations. With this approach, the wave form to power a sensor is tuned according to a desired power efficiency achievement. Besides adjustable duty cycles, adaptively changing sampling periods would be another approach [6]. By changing sampling periods as needed, the total number of sampling occurrences either increases or decreases. As a result, relevant power consumption will do accordingly. Above of all, an adaptive sensor management mechanism to assign a mixture pair of duty cycles and sampling periods, which is proposed in this chapter, would be a cure of consuming less power while running context-aware applications accurately enough.

In summary, to be able to run context extracting/infering applications, energy saving methods must be considered and applied for the mobile devices; however, this jeopardizes the accuracy, i.e., quality of service, provided by the applications, and thereby it creates a tradeoff between the power consumption caused and the service accuracy provided by the context-aware applications.

This chapter proposes a novel framework as shown in Fig. 6.1 in order to run a HAR based application while achieving a fine balance in the defined tradeoff. The framework consists of a context inference module including an observation analysis module and a statistical machine to acquire and infer the desired contexts through a sensor and to represent user activities, and also a sensor management system to prolong device battery lifetimes. The ultimate goal of this chapter is to recognize user activities accurately and to make sure power efficiency is maximized. Most importantly, this research intends to create and clarify an effective HMM-based framework included

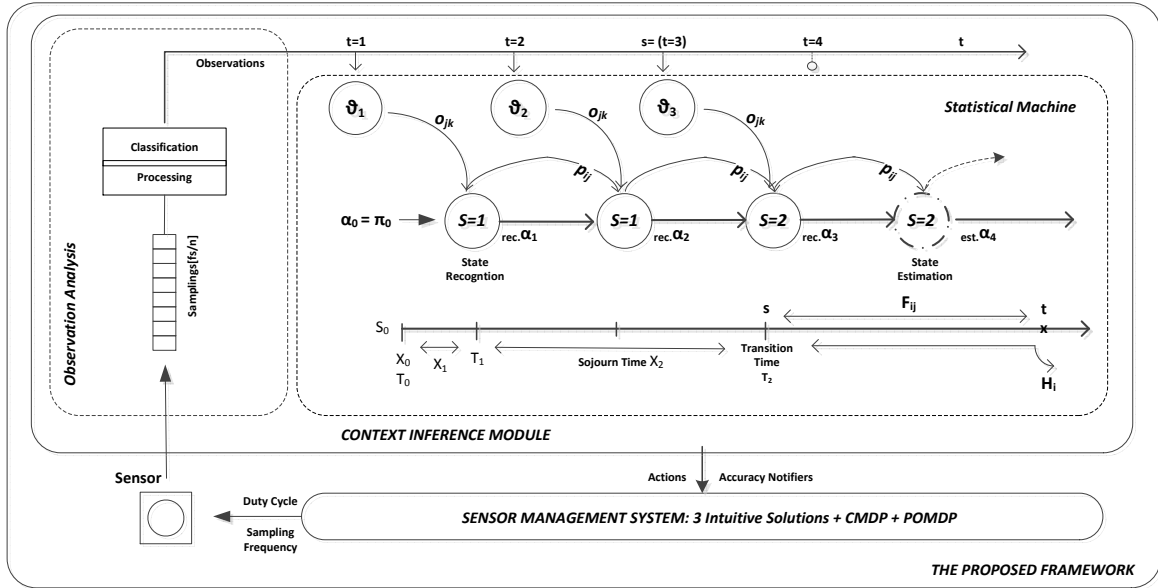


Figure 6.1: The operational work-flow of the proposed framework

with power saving methods at the low-level sensor operations in order to guide the development of future context-aware applications.

There are a few distinctive novelties shown in Figure 6.2 that this chapter exposes as follows:

- A light-weight and unsupervised classification algorithm is applied over sensory data to produce observations as the inputs of HMM-based statistical machine.
- User profiles are considered time-variant (inhomogeneity) in the statistical machine.
- Adaptability problem is defined for time-varying user profiles, and a relevant solution is given by introduction of the entropy production rate.
- The adaptive statistical machine regulates the power saver mechanism.
- The analytical modeling of the accelerometer sensor is provided.
- A power saver mechanism is provided by utilizing a mixture pair of duty cycling and adaptive sampling in order to prolong mobile device battery lifetimes.
- Five different sensory operation methods are provided: three intuitive solutions, Constrained Markov Decision Process (CMDP) and Partially Observable Markov Decision Process (POMDP) based sub-optimal solutions

- Missing observations occurred due to the power saver mechanism are found according to inhomogeneous semi-Markovian process.
- A real time smartphone application is implemented to show the power consumption analysis of the different sensor operations, and a well examined offline process is also simulated in presence of application results and the sensor model in order to show the effectiveness of the proposed framework and power saving methods.

The least power consuming sensor on today's smartphones is the accelerometer [147]. Therefore, the accelerometer sensor is considered for use in the implementation of HAR based applications. The Blackberry RIM Storm II 9550 smartphone is chosen target device. Storm II consists of 3-axis accelerometer named ADXL346 from Analog Devices [148]. While application is running, the target smartphone is only connected to a 3G network, and the background operations kept minimal. In addition, for the sake of simplicity, a two-user state, which are 'sitting' and 'standing', consisting

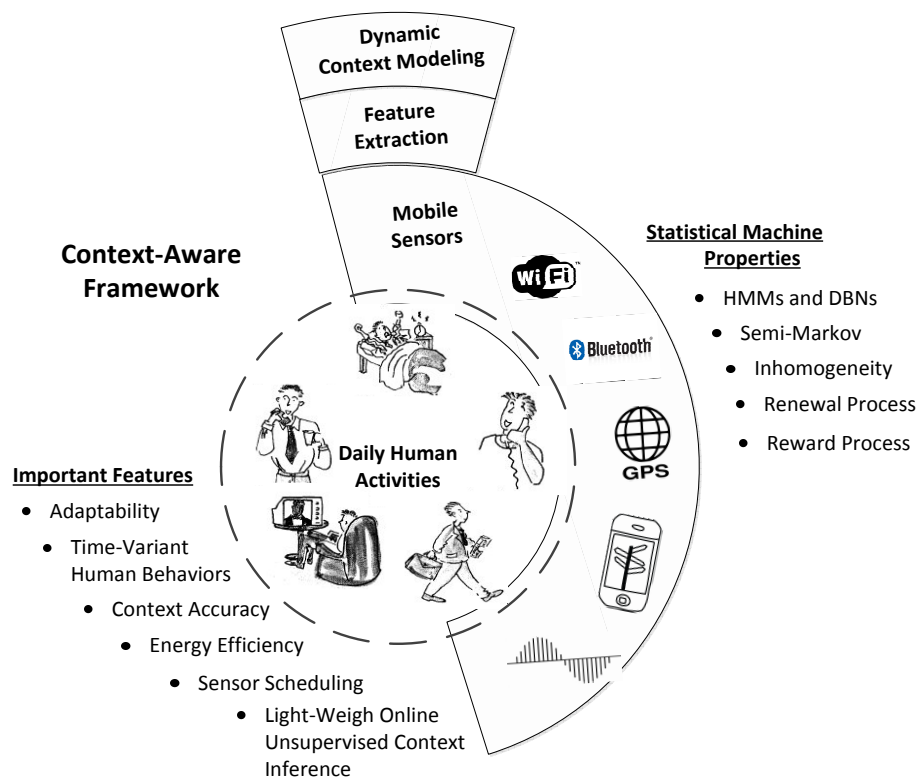


Figure 6.2: The properties of context-aware framework in mobile computing

Table 6.1: Summary of important symbols in chapter 6

Symbol	Definition (Section where the symbol is first used)
S_t	user state (6.2.1)
S_s^τ	Markov chain, or sequence of user states (6.2.1)
ϑ_t	observation (6.2.1)
ϑ_s^τ	sequence of observations (6.2.1)
o	observation emission matrix (6.2.1)
n, s, t, τ	time indexes throughout the chapter (6.2.1)
i, j, m	indexes for user states (6.2.1.1)
ξ	inhomogeneous Markov process (6.2.1.1)
q_{ij}	user state transition rate (6.2.1.1)
Q	user state transition density matrix (6.2.1.1)
p_{ij}	user state transition probability (6.2.1.1)
P	user state transition matrix (6.2.1.1)
π_i	initial user state probability (6.2.1.1)
F_{ij}	probability of waiting time in a state (6.2.1.2)
H_i	probability of leaving a user state (6.2.1.2)
d_i	a random time distribution (6.2.1.2)
\mathcal{F}_j	filtered probabilities (6.2.1.3)
\mathcal{P}_j	predicted probabilities (6.2.1.3)
\hat{S}_t	estimation of user state (6.2.1.3)
N	total number of user state transitions (6.2.1.4)
N_i	total number of passages in a fixed user state (6.2.1.4)
e_p	instantaneous entropy production rate (6.2.2)
ϕ	accuracy notifier (6.2.2)
a	actions (6.2.2)
t_{suff}	sufficient time to trigger an action (6.2.2)
S^R , or S^{R^2}	1D, or 2D state space for reward process (6.3.1)
r, w	indexes for states $\in S^R$ (6.3.1)
l, k	indexes for $l \in DC$ and $k \in f_s$ (6.3.1)
$\Theta_{t_{span}}$	total power consumption for a spanning time (6.3.1)
ψ_{S^R}	reward process attached to ongoing S^R (6.3.1)
\bar{V}	total received reward, i.e. power consumption (6.3.1)
u	optimal policies in CMDP and POMDP (6.3.4)
P^a	state transition matrix under actions (6.3.4)
I	identity matrix (6.3.4)
λ	belief vector (6.3.5)
R^a	rewards according to actions (6.3.5)

statistical machine is considered in the framework. However, more complex models can be applied as well by using same system approach.

In the following sections, Section 6.1 gives the relevant prior research. Section 6.2 is dedicated to the total explanation of the context inference module which encapsulates the analysis of the sensory data and the creation of the statistical machine in order to represent true user activities and behavior. Section 6.3 includes the analytical model of sensor utilization, and power saving solutions to balance the tradeoff. Section 6.4 is left for performance analysis of the proposed method. Finally, Section 6.5 is for conclusion and future work. In addition, the summary of important notations used throughout the chapter is listed in Table 6.1.

6.1 Prior Works

The pervasive context aware mobile computing, which captures and evaluates sensory contextual information in order to infer user relevant behaviors, has been becoming a well established research domain. Most studies rely on recognition of user activities and definition of common user

behaviors, especially with the realm of Human Activity Recognition (HAR) [12, 28] and location-based services [147, 166]. In addition, researchers have been aware of the existing trade-off. However, most works provide some partial answers to the tradeoff. It is difficult to be said that power saving considerations are significantly taken at the low-level physical sensory operations. Especially, there is not a framework construction that intends to apply *adaptively changing* duty cycles and sampling period on a sensory operation like this chapter intends to propose. In contrast, most works done so far emphasize either to set a minimum number of sensors when needed in a context-aware application or to maximize power efficiency by solely applying less complexity in computations or by changing transferring methods of data packets.

From the literature search, it is important to refer [12], which proposes a hierarchical sensor management system. The system improves the device battery life by powering a minimum number of sensors. Unfortunately, sensors have fixed duty cycles whenever they are utilized and they are not adjustable to respond different to user behaviors. The hierarchical sensor management system is also studied in [28], which achieves energy efficiency and less computational complexity by only performing continuous detection of context recognitions when changes occur during the context monitoring. Moreover, the advantage of a dynamic sensor selection scheme for accuracy-power tradeoff in user state recognition is demonstrated in [33, 165, 167, 168]. [6] also use different sampling period schemes, which are assigned according to the stream of context events, for querying sensor data in continuous sensing mobile systems to evaluate energy-accuracy tradeoffs. The same solution attempts but at this time for localization applications can be found in [147], which describes a system for saving energy consumption in sensing localization applications for mobile phones. Also, [166] studies energy efficiency in mobile device based localization, and the authors show that human can be profiled based on their mobility patterns and thus location can be predicted. The proposed system achieves good localization accuracy with a realistic energy budget. On the other hand, energy saving in wireless sensors is a well studied topic. Relevant studies can be found in [169, 170]. In addition, for the studies on energy-hunger sensors and for the solutions to achieving energy efficiency by employing them, it can be referred to [171].

6.2 The Context Inference Module

The context inference framework consists of two main modules as shown in Fig. 6.1, which are sensory data acquisition and analysis, and a statistical machine. The first module receives *raw*

sensory readings (i.e., extracted user contexts through mobile device based sensors) as *inputs*. These readings undergo a series of signal processing operations, and eventually end up with a classification algorithm in order to provide desirable inferences about user relevant information for context-aware applications. A required classification algorithm differs according to the inference of the interested context through a specific sensor. The probabilistic outcomes of the classification algorithms source the inputs of the second module.

The second module chooses a Discrete Time Inhomogeneous Hidden Semi-Markov Model (DT-IHS-MM) as the desired statical machine. Using Hidden Markov Models (HMMs) in order to infer mobile device based human-centric sensory context have already been applied in Human Activity Recognition (HAR) [140]. However, this chapter intends to expand the properties of the statistical machine so as to obtain a better realization in context-awareness. First, the concept of Markov Renewal Process is adopted to describe the functionalities of user behavior modeling. Second, the inhomogeneity is introduced in order to characterize time-variant user behaviors so that the framework could adapt itself to dynamically changing user behaviors. Third, the semi-Markovian feature is added in order to specify aperiodically received discrete time observations through sensory readings. Fourth, the estimation theory is included in case of missing sensory inputs. Finally, the entropy rate is provided in order to demonstrate the accuracy of inferences made by the framework since there is not an absolute solution to actually calculate the accuracy of a real-time running HAR based context-aware application. *The convergence of the entropy rate* is considered as *output* of the framework, which will be used later by a sensor management system in Section 6.3.

6.2.1 Inhomogeneous Hidden Semi-Markov Model: A Statistical Machine

Classification algorithms produce observations (i.e., *visible states*), ϑ_t , of DT-IHS-MM. Amongst observations, there is only one observation expected to provide the most likely differentiation in the selection of instant user behavior. This observation is marked as instant observation, ϑ_T , which also indicates the most recent element of observation sequence, ϑ_1^T , of DT-IHS-MM. On the other hand, user states, *sitting* and *standing*, are defined as *hidden states*, S , of DT-IHS-MM since they are not directly observable but only reachable over visible states. Therefore, each observation has cross probabilities to point a user state. These cross probabilities build an observation emission matrix, o , which basically defines decision probabilities to pick any user states from available observations.

In addition to that, the transition probabilities among user states might not be stationary since a general user behavior changes in time. Thus, it is expected from a user state either to transit into another user state or to remain in the same with a different probability. These occurrences build a time-variant user state transition matrix, p .

6.2.1.1 Basic Definitions and Inhomogeneity

Let an inhomogeneous Markov process exist as $\xi = \{\xi(t), t \geq 0\}$ with a user state space of $S = \{1, 2, \dots, M\}$ and let $Q(t) = q_{ij}(t)$ where $\{i, j\} \in S$ and $t \geq 0$ be a transition density matrix of ξ . If Q satisfies both $0 \leq q_{ij}(t) \leq \infty$ and $q_i(t) = -q_{ii}(t) = \sum_{i \neq j} q_{ij}(t)$, then Q is called a conservative inhomogeneous transition density matrix function on S .

$q_{ij}(t)$ represents jump or transition rates from user state i to user state j at time t . Whenever $i = j$, it means that the current user state remains unchanged, or i.e., a dummy transition occurs.

Moreover, suppose that a user state transition probability matrix $P(s, t) = \{p_{ij}(s, t) = Pr(S(t) = j | S(s) = i)\}$ where $t \geq s \geq 0$ together with Q satisfies both forward and backward Kolmogorov's equations [172], which assume to have $\lim_{t \downarrow s} \partial p_{ij}(s, t) / \partial t = q_{ij}(s)$, then S becomes an *inhomogeneous Markov chain* with the transition density of Q . The chain can revisit a user state at different system times, and also not every user state needs to be visited. Hence, there is no requirement that user state transition probabilities must be symmetric ($p_{ij} \neq p_{ji}$) or a specific state might remain in the same in succession of time ($p_{ii} = 0$).

Furthermore, let an initial user state $\pi(t) = \{\pi_i(t) = Pr(S(t) = i)\}$ satisfy the Fokker-Planck equation [173]: $d\pi(t)/dt = \pi(t)Q(t)$.

6.2.1.2 The Working Process

Let $\xi = \{\xi_n, n \in \mathbb{N}\}$ be redefined as an inhomogeneous irreducible discrete Markov process with a user state space of S . The process evolves from S_0 as initial user state and stays in there for a non-negative length of time X_1 until going into another user state S_1 . Then, it stays in the new user state for X_2 before entering into S_2 , and so on. As indicated in [174–176], this process is a two-dimensional or bivariate stochastic process in discrete time called positive ($S - X$) process: $(S - X) = ((S_n, X_n), n \geq 0)$ with initial of $X_0 = 0$ where X_n is called the successive sojourn times.

X_n is the time spent in state S_{n-1} which defines inter-arrival times. There is also another time variable T_n introduced for the definition of times at which state transitions occur. This random

time sequence is called renewal sequence, and it is given by $X_n = T_n - T_{n-1}$, $n \geq 1$ with the initial statuses of $\{X_0, T_0\} = \{0, 0\}$.

The Markov renewal process is now redefined over two-dimensional process of $(S - T) = ((S_n, T_n), n \geq 0)$ by

$$Q_{ij}(s, t) = Pr(S_{n+1} = j, T_{n+1} \leq t | S_n = i, T_n = s), \quad (6.1)$$

where T_n represents n-th renewal time at which a user state transition happens.

The probability of waiting time, also called conditional distributions of sojourn times, for each user state i in the presence of (6.1) and information about the successively followed user state is given by

$$F_{ij}(s, t) = Pr(T_n \leq t | S_{n-1} = i, S_n = j, T_{n-1} = s),$$

$$= \begin{cases} Q_{ij}(s, t)/p_{ij}(s), & p_{ij} \geq 0, \\ 1, & p_{ij} = 0. \end{cases} \quad (6.2)$$

In addition, with the help of (1) and (6.2), the probability of the process leaving the user state i , also called sojourn times distributions in a given user state, from time s to t is introduced by

$$H_i(s, t) = Pr(T_n \leq t | S_{n-1} = i, T_{n-1} = s),$$

$$= \sum_j p_{ij}(s) F_{ij}(s, t) = \sum_{j \neq i}^U Q_{ij}(s, t). \quad (6.3)$$

If $F(s, t) = F(t - s)$, $s \leq t$, then the kernel Q only depends on $t - s$, which it yields to have $Q(t - s) = p * F(t - s)$ being called an inhomogeneous semi-Markov process. The semi-Markov process [177, 178] indicates that the sojourn time in each state might have a random distribution, $d_i(t)$ ¹, see Figure 6.3, which can depend on the next user state to be visited. Thereby, this yields to

¹The proposed solutions in Section 6.3 regulate the sampling times in the sensory operations, and change the time distribution accordingly.

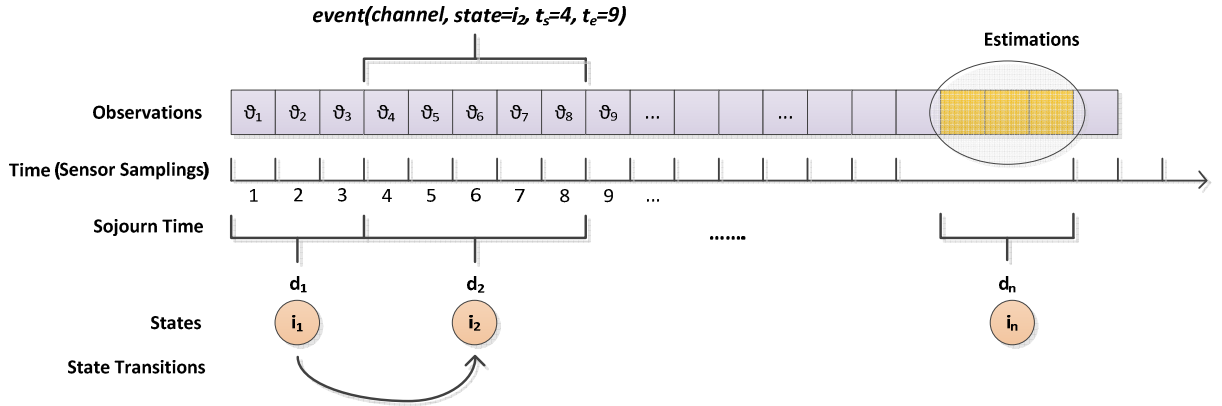


Figure 6.3: Semi-Markovian feature by sensor samplings

find the probability of a user state transition being occurred at time t :

$$b_{ij}(s, t) = \begin{cases} Q_{ij}(s, t) = 0, & t \leq s \\ Q_{ij}(s, t) - Q_{ij}(s, t - 1), & t > s. \end{cases} \quad (6.4)$$

Also, for each waiting time, a user state is occupied. Therefore, the transition probabilities are defined with (6.3) and (6.4) by

$$p_{ij}(s, t) = Pr(S_t = j | S_s = i) = \delta_{ij}(1 - H_i(s, t)) + \sum_{m \in M} \sum_{\tau=1}^t b_{im}(s, \tau) p_{mj}(\tau, t), \quad (6.5)$$

where δ_{ij} represents the Kronecker symbol.

The first element of RHS where $d_i(t) = 1$ if $i = j$ notifies the probability of remaining in user state i at time t without any change in context from time s ; and, the second represents the probability of a user state transition from state i in some way to user state j and staying in this new user state at time t .

6.2.1.3 User State Representation Engine

User state representation engine infers an instant user behavior in light of prior knowledge of a human behavior pattern, i.e., system parameters defined in Section 6.2.1.1, and the availability of sensory observation at a decision time. If sensory observation exists, the applied process is called recognition method; otherwise, estimation method, see Fig. 6.4.

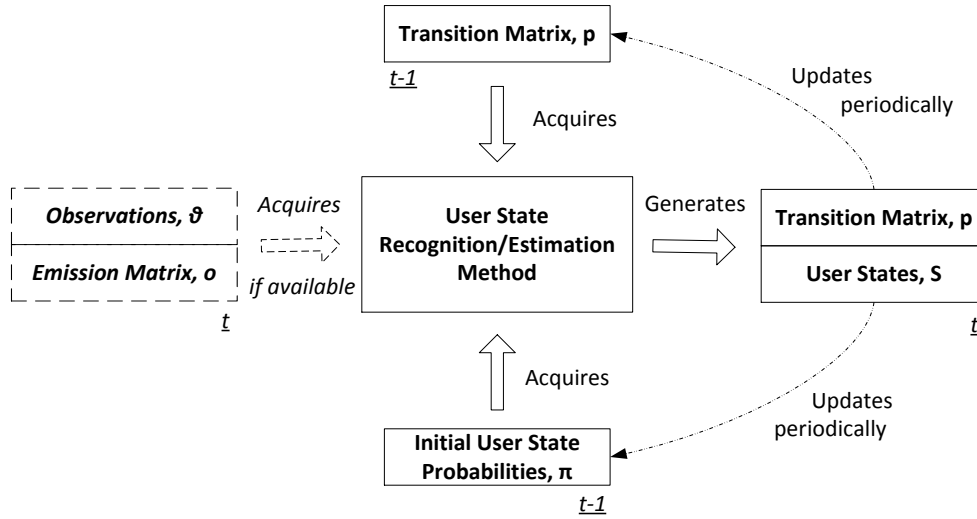


Figure 6.4: User state representation engine: recognition and estimation models

Let ϑ_t denote an observation at time t which is associated with user state S_t , and let $o_i(\vartheta_t)$ be the probability of observing ϑ_t from given $S_t = i$. Thus, $o_i(\vartheta_s^t) = \prod_{t'=s}^t o_i(\vartheta_{t'})$ represents a sequence of the emitted observations from time s to t , $s \leq t$. In addition, note that since the process flows in a discrete time and follows the first order Markovian feature, the current user state S_t depends solely on the most recent user state S_{t-1} .

The inference of a hidden user state j at time t given the last known hidden user state i at time s , $s \leq t$ is presented by

$$Pr(S_t = j | S_s = i, \vartheta_s^t), \quad (6.6)$$

where $\tau = t - s$. (6.6) is termed as *predicted* \mathcal{P} , *filtered* \mathcal{F} , and *smoothed* \mathcal{S} , probabilities of S_t , depending on observation sequence of ϑ_s^{t-1} , ϑ_s^t or ϑ_s^T respectively where $T > t$.

$$\alpha_j(s, t) = \sum_{t'=s}^t \sum_i \left(\alpha_i(s, t-t') p_{ij}(s, t-t') d_j(t') \prod_{t''=1}^{t'} o_j(\vartheta_{t-t'+t''} = z) \right), \quad (6.7)$$

The recognition method uses the filtered probabilities of S_t where it is derived from (6.6) as in $\mathcal{F}_j(s, t) = Pr(S_t = j | S_s = i, \vartheta_s^t)$ in presence of sufficient number of available observations. The probability of an instant user state recognition is found by the forward algorithm [140], which is proposed to find the most likely one-step ahead user state in a hidden chain. The forward algorithm relies on updating a probability weight α inductively, which decides the probability of current

occurrence of a user state, S_t , generated from the one-step-previous occurrence, S_{t-1} . However, this method works well for traditional HMMs not for Semi-Markovian featured models due to the random sojourn time distribution between two consecutive user states in the hidden chain. In this manner, an extended forward algorithm has been proposed in [179, 180] by (6.7) with the condition of $\alpha_i(s, t) = \pi_i$ if $t \leq 0$.

Since the recognition process of user states evolves in real time by context-aware applications, the forward algorithm is employed to assign a proper user state in order to specify current user activity whenever a new observation is made. On the other hand, to make sure that user state recognitions are made true, the backward algorithm, whose corresponding weight is denoted by β , is employed [140]. By this algorithm, the accuracy of previous user state recognitions is validated, i.e., *smoothing*. However, applying this algorithm seems redundant due to the fact that it consumes additional computational power on the mobile device batteries, and also the context-aware applications run in real-time, thereby there is no point of discovering what happened in the past again. Hence, the filtered probability becomes

$$\mathcal{F}_j(s, t) = \alpha_j(s, t)\beta_j(s, t') = \alpha_j(s, t), \quad t = t' = T. \quad (6.8)$$

Then, the instant user state recognition is found using (6.8) in case where observations are available by

$$S_t = \arg \max_{1 \leq j \leq M} [\mathcal{F}_j(s, t)]. \quad (6.9)$$

Note that computational complexity while calculating system parameters causes a crucial underflow problem. When time goes by during evolution of ξ , $\alpha_i(0, t) \xrightarrow{t} 0$ starts to head to zero at an exponential rate since p_{ij} includes elements being lower than 1. Therefore, $\mathcal{F}_j(s, t)$ needs to be scaled [140] by a factor of $\prod_{t'=s}^t \sum_j \mathcal{F}_j(s, t')$.

In addition to using the filtered probabilities to recognize user states, the predicted probabilities are used to estimate user state in case of no observation received. When power saving methods are taken into consideration as studied in Section 6.2.2, there will be some time intervals during sensory operations in which no sensor readings are obtained. As a result, the framework cannot receive a relevant observation. In that case, the inference of instant user state is based on the estimation method not on the recognition method.

The predicted probabilities are found by

$$\begin{aligned}
\mathcal{P}_j(s, t) &= Pr(S_t = j \mid S_s = i, \vartheta_1^{t-1}, \vartheta_t), \\
&= Pr(S_t = j, \vartheta_t = z \mid S_s = i, \vartheta_1^{t-1}), \\
&= \sum_j \mathcal{F}_i(s, t-1) p_{ij}(s, t-1).
\end{aligned} \tag{6.10}$$

Alternatively, (6.10) can be found by assigning the most likely visionary observation instead by accepting there is a missing observation:

$$\mathcal{P}_{j,z}(s, t) = \sum_j \mathcal{F}_i(s, t-1) p_{ij}(s, t-1) o_j(\vartheta_t = z). \tag{6.11}$$

Then, the most likely observation is selected according to assigning each possible observation as a final node to observation sequence while calculating (6.11) by

$$\hat{\vartheta}_t = \arg \max_z \sum_j \mathcal{P}_{j,z}(s, t). \tag{6.12}$$

Finally, the instant user state estimation is found using (6.11) together with (6.12) by

$$\hat{S}_t = \arg \max_{1 \leq j \leq M} [\mathcal{P}_{j,z}(s, t)]. \tag{6.13}$$

6.2.1.4 Time-Variant User State Transition Matrix

The most important feature of context-aware applications is being capable of adapting themselves to distinctive user behaviors. User relevant context differs in time and the corresponding user state also does. For instance, one user might remain in the same user state for a long time; whereas others might be more active by changing their user states frequently. Therefore, it cannot be expected from the user state transition matrix to remain stationary under such conditions.

- *Default Settings:* User state transitions can be represented as simple random walk on a graph [141, 142]. On this graph, a vertex, v , represents a user state, and an edge represents a user state transition. Thus, ξ always starts evolving by a default transition matrix, which is

$$p_{ij}^{default} = \frac{1}{d(v_i)}, \quad v_i \sim v_j \tag{6.14}$$

where $d(v_i)$ is the number of vertices v_j adjacent to v_i . For example, if $d(v_i)$ is 0, $p_{ii}^{default} = 1$.

- *Update*: A random variable $N(t) > n - 1 \leftrightarrow T_n \leq t$ is represented as the total number of jumps or transitions of the $(S - T)$ process during $(s=0, t]$. Therefore, $N(t)$ is also called the discrete-time counting process of the number of the jumps. Jumps or transitions may include any transition towards user state itself (i.e., virtual transitions).

By having the counting process, there are some counting parameters that can be calculated where $0 < \tau \leq t$ as follows:

- The number of visits to user state i during $(0, t]$: $N_i(t) = \sum_{n=0}^{N(t)-1} 1_{\{S_n=i\}}$
- The number of transitions from user state i to user state j during $(0, t]$: $N_{ij}(t) = \sum_{n=1}^{N(t)} 1_{\{S_{n-1}=i, S_n=j\}}$
- The number of transitions from user state i to user state j during $(0, t]$ with the sojourn time, τ , in state i : $N_{ij}(\tau, t) = \sum_{n=1}^{N(t)} 1_{\{S_{n-1}=i, S_n=j, X_n=\tau\}}$

The empirical estimations of the user state transition matrix, p_{ij} , the conditional distributions of the sojourn times, f_{ij} , and the discrete time semi-Markov kernel, q_{ij} , are given in [181, 182] by

$$\begin{aligned}\hat{p}_{ij}(t) &= N_{ij}(t)/N_i(t), \\ \hat{f}_{ij}(\tau, t) &= N_{ij}(\tau, t)/N_{ij}(t), \\ \hat{q}_{ij}(\tau, t) &= N_{ij}(\tau, t)/N_i(t).\end{aligned}\tag{6.15}$$

Given empirical estimations in (6.15) approach non-parametric maximum likelihood estimations with having good asymptotic properties if they maximize the likelihood function of

$$\mathcal{L}(t) = \prod_{n=1}^{N(t)} p_{ij} f_{ij}(X_n) \left(1 - \sum_j \sum_{\tau=n}^{B(t)} q_{ij}(\tau)\right).\tag{6.16}$$

where $B(t) = t - X_{N(t)}$ is called age process showing the sojourn time in the last visited state $S_{N(t)}$.

With the evaluation of (6.16), the corresponding transition density kernel turns into

$$Q_{ij}(s, \tau_0, \tau) \stackrel{update}{=} \frac{Q_{ij}(s, \tau) - Q_{ij}(s, \tau_0)}{Q_{ij}(s, \tau_0)},\tag{6.17}$$

where τ_0 is the elapsed time since the first entrance into user state i .

$$p_{ij}(s, \tau_0, \tau) \stackrel{\text{update}}{=} \delta_{ij}(1 - H_i^*(s, \tau_0, \tau)) + \sum_{m \in M} \sum_{v=\tau_0}^{\tau} b_{im}^*(s, \tau_0, v) p_{mj}(s + v, \tau - v). \quad (6.18)$$

Finally, beginning from the default status in (6.14), the evolving inhomogeneous state transition probability (6.5) is updated by (6.17) together with (6.3) and (6.4) as in (6.18).

6.2.1.5 Observation Emission Matrix

In Chapter 3, a light-weight online classification method to detect the user centric postural actions, such as sitting, standing, walking and running, by smartphones is studied. By applying this work into the current study, the differentiation between user state ‘sitting’ and user state ‘standing’ is made. Then, the observation emission matrix is constructed by

$$o_{jz} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \text{prob. of} \\ \text{sitting} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \text{prob. of} \\ \text{standing} \end{bmatrix} \quad (6.19)$$

6.2.2 The Output of the Context Inference Framework

In Chapter 4, the convergence of entropy rate belonging to HMM based HAR framework can distinguish a specific user activity profile is pointed out. In addition to that, the linkage between the entropy rate and the accuracy of the recognitions is enlightened in this chapter.

Supposing $\pi_i(0) > 0$ where $\forall i \in S$, the Markov process ξ_n evolves in bidirectional way over the distributions of $P_{[n, n+\acute{n}]}$ and $P_{[n, n+\acute{n}]}^-$ where $\forall n \in \mathbb{Z}^+$ and $\forall \acute{n} \in \mathbb{N}$, and the user state transition matrix also obeys the condition of $p_{ij} > 0 \leftrightarrow p_{ji} > 0$, then ξ_n satisfies

$$\lim_{t \downarrow s} \frac{\pi_i(s) p_{ij}(s, t)}{\pi_j(s) p_{ji}(s, t)} = 1, \quad (6.20)$$

which indicates that the inhomogeneous Markov process has instantaneous reversibility at time s , and hence it yields to have $\pi(s)Q(s) = 0$.

Having the reversibility feature defined by (6.20), the instantaneous entropy production e_p^n of ξ at time n is given by

$$e_p^n = \mathcal{H}(P_{[n,n+1]}, \bar{P}_{[n,n+1]}) = \frac{1}{2} \sum_{i,j \in S} [\pi_i^n p_{ij}^n - \pi_j^n p_{ji}^n] \log \frac{\pi_i^n p_{ij}^n}{\pi_j^n p_{ji}^n}. \quad (6.21)$$

where $\mathcal{H}(P_{[n,n+1]}, \bar{P}_{[n,n+1]})$ is the relative entropy of the distribution of (ξ_n, ξ_{n+1}) , $P_{[n,n+1]}$, with respect to the distribution of (ξ_{n+1}, ξ_n) , $\bar{P}_{[n,n+1]}$.

By using (6.21), Fig. 4.10 shows the convergence of entropy rate under sensory operation parameters such as fixed duty cycle $DC = 1$ and variant sampling frequencies $f_s = \{100, 50, 25, 12.5\}$ Hz. Aggressive sampling method, which takes 100 Hz as f_s , draws an actual track of the entropy rate. Circles over blue line indicate a difference in user behavior. Since user states, such as sitting and standing, are recognized, the frequentness of transition from one user state to another cannot be much due to the human nature. Therefore, user state transition matrices over time are desired as

$$p_{ij} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.85 & 0.15 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.75 & 0.25 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.6 & 0.4 \\ 0.1 & 0.9 \end{bmatrix}, \begin{bmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{bmatrix}.$$

According to the results, the entropy rate converges late while samplings are collected at f_s being less than 100 Hz. This indicates the reason why accuracy rate decreases as well. The most significant result can be obtained from Fig. 4.10 is the entropy rate cannot sometimes converge into any point, where the plot lines belonging to $f_s = \{12.5, 25, 50\}$ Hz stop. When the frequentness of user state transitions increases, sampling frequency may not be fast enough to capture the activeness of a user profile. Therefore, the system cannot find any proper user state transition matrix to define instant user activity profile.

Let $e_p(s, t)$ denote a sequence of entropy rates from (6.21) in time range of s up to t . In addition, assume that a simple threshold is defined by $\varepsilon_{e_p} = [\mu_{e_p} - \sigma_{e_p}, \mu_{e_p} + \sigma_{e_p}]$ where μ_{e_p} and σ_{e_p} are mean and standard deviation of $e_p(s, t)$ respectively. Thereby, the first output delivered to the sensor management system by the statistical machine as shown in Fig. 6.1 is named as accuracy notifier, which is defined by

$$\phi(s, t) = \frac{1}{(t - s + 1)} \sum_{n=s}^t 1_{(e_p^n \in \varepsilon_{e_p})}. \quad (6.22)$$

Moreover, τ_{ii} denotes the return time, i.e., elapsed total sojourn time, to user state i entering at s .

$$\tau_{ii} = \begin{cases} \min\{n = t - s, n \geq 1, S_t = i \mid S_s = i\}, \\ \infty, \quad S_t \neq i, t \geq 1, \end{cases} \quad (6.23)$$

represents the amount of time until the process returns to the same user state i given the fact that it started from user state i . Note that it may never return back to the same state i .

By considering that a time variable t_{suff} is assigned during application run to indicate a sufficient time interval in which user state i would not change, the second output is defined then using (6.22) and (6.23) by

$$a(t) = \begin{cases} 1, & \phi(s, t) \geq \phi, \tau_{ii} > t_{suff}, \\ 2, & \phi(s, t) \geq \phi, \tau_{ii} \leq t_{suff}, \\ 3, & \phi(s, t) < \phi, \end{cases} \quad (6.24)$$

where $\phi \in [0.5, 1]$ and $a(t)$ denotes the actions for sensory management introduced in Section 6.3.2.

6.3 Sensor Management System

In this section, the effect of variant sensory load profiles on the mobile device battery depletion is studied. Then, these battery discharge profiles are examined within the concept of Markov Reward process. In addition, there are five novel solutions provided in this section for balancing the tradeoff existing between the accuracy in the user state recognitions and the power consumption required by the recognition process.

6.3.1 Sensor Utilization

The smartphone accelerometer sensor is utilized in order to examine the power efficiency achieved under different sampling and duty cycling strategies. Assume that a set of DC and a set of f_s are given by $\{1, 0.75, 0.5\}$ and $\{100, 50, 25, 12.5\}$ Hz respectively. In addition, let a state space lie over two sub-spaces, which are sets of DC and f_s , as in $S^{R^2} = \{\text{DC} \times f_s\}$. Thus, the state space is defined as in

$$S_r^R = S_{\{l,k\}}^{R^2} = \{S_{\{1,100\} \rightarrow 1}^{R^2}, S_{\{1,50\} \rightarrow 2}^{R^2}, \dots, S_{\{0.5,12.5\} \rightarrow W}^{R^2}\}, \quad (6.25)$$

where $S^{R^2} \rightarrow S^R : \{DC = l, f_s = k\} \rightarrow r, \forall l \in DC, \forall k \in f_s$ and $W = \text{length}(DC) \times \text{length}(f_s)$.

The state space S^R , or S^{R^2} , is considered to represent different sensory operation methods supported by the accelerometer sensor in a sensor management system.

To be able to see the effect of S^R on the battery depletion, an application is implemented on the target device. The application runs from point where the smartphone battery is fully-loaded until it totally depletes. Only one constant pair of sampling frequency and duty cycle, i.e. a state in S^R , is applied as sensory operation parameters to the accelerometer at each application run. A total time for sensory operation cycle, denoted by t_c , is also defined as 1 second. For instance, where $f_s = 100$ Hz, $DC = 100\%$ and $t_c = 1$ sec are taken, the total number of samplings per second becomes 100.

The application results are shown in Fig. 5.9. Note that Blackberry Java 7.1. SDK only reveals remaining battery status. According to results, the more aggressive sampling methodology is applied, the faster the battery depletes. In addition, the lower value of DC makes the battery recovery effect more significant, and thus it prolongs the battery lifetimes. However, the battery non-linearities [156, 162] are not intended to study in this chapter.

In Chapter 4, the analytical model of the accelerometer sensor was studied, and the sampling frequency dependent power consumption in sensory operations was found as in

$$\Theta_{\omega*12.5Hz} = \omega * \Omega_{sample} + (\omega_{max} - n) * \Omega_{idle}, \quad (6.26)$$

where $\omega = \{1, 2, 4, 8\}$, and Ω_{sample} and Ω_{idle} are defined as power consumption occurred during the operations where sensor makes samplings or runs idle respectively, refer [148] for a detailed information.

After the application results shown in Fig. 5.9, (6.26) can be extended approximately during a time span of t_c as in

$$\Theta_{t_c} = \frac{(DC)t_c}{\omega_{max}/f_{smax}} \Theta_{\omega*12.5Hz} + \frac{(1-DC)t_c}{1/f_{smax}} \Omega_{idle}. \quad (6.27)$$

By using (6.27) and the application results obtained for Fig. 5.9, Fig. 6.5 shows power consumption rate of each sensory operation methods by the accelerometer, where the aggressive sampling method, $DC = 100\%$ and $f_s = 100$ Hz, is taken as normalizing factor.

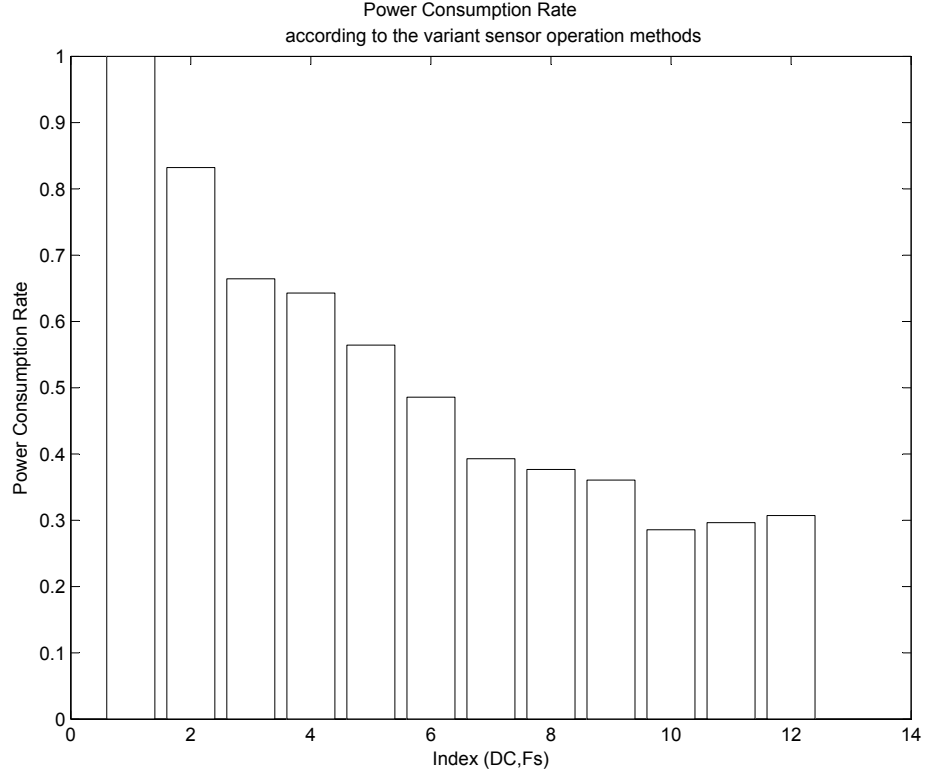


Figure 6.5: Power consumption rate according to variant sensory operation methods

Assume that a semi-Markov chain represents the evolution of changing sensory operation methods, S^R , for a desired sensor management system. The chain consists of a finite state space $S^R = \{1, \dots, W\}$, the state transition density matrix $q^R \in Q^R$, and the state transition matrix $p^R \in P^R$ where $Q^R, P^R \in \mathbb{R}^{W \times W}$. In addition, a reward structure can be attached to this ongoing chain, and it can be thought as a random variable associated with the state occupancies and transitions. Moreover, assume that the reward, denoted by ψ , is seen as power consumption per unit time while a mobile device battery is discharging, and S^R is redefined as the battery discharge profiles/states. Thereby, the total reward, i.e., total power consumption, depends on the total visiting time in a state r where $r \in S^R$. Then, it can be said that the reward ψ_r belonging to state r is proportional to the corresponding power consumption defined by (6.27).

Finally, the general evolution of a semi-Markov reward process to describe power consumption caused by sensory operations in the sensor management system is given by

$$V_w(s, t) = V_r(s, t - 1) + \sum_{w \in W} p_{rw}^R(s, t - 1) \psi_w(s, t). \quad (6.28)$$

The LHS, V , represents the expected present value of all received rewards from time s to t given that process enters into state i at time s . Whereas, the first element of RHS represents the aggregation of rewards earned both at previous time; and the second element of RHS is the reward obtained from either continuity in the same state or transition to another state.

6.3.2 The Trade-off Analysis: The Description of Action Set

There are five different solutions proposed in order to respond the defined trade-off between the sensing accuracy and the power consumption. The proposed solutions aim at reducing power consumption by intervening sensory operations. Therefore, the context inference framework always receives the manipulated sensory samplings, then tries to recognize user states accurately according to (6.9) and (6.13). After the recognition process is done, it releases $a(t)$ as in (6.24), which defines actions to be taken on sensory operations. These actions force the proposed solutions to adjust a pair of duty cycle or sampling frequency dynamically while sensory sampling operations are actively operated. As a result, a feedback system is integrated into a cyber-physical sensor management system which balances the increase in power efficiency with the decrease in user state recognition accuracy.

Actions are defined as commands $\{1, 2, 3\}$ for sensor management, which are to *decrease*, *preserve* and *increase* power consumption respectively. If the entropy rate is not stable, which means user profile changes frequently, and thereby the corresponding entropy rate does not converge a specific value. Action #3 needs to be taken in this case to increase the power consumption in sensory operations by making more aggressive samplings. In contrast, if the entropy rate converges and hangs in a specific margin, then action #2 preserves the same set-up for the applied sensory operations. More significantly, if the same user profile has been observed at least for a sufficient time t_{suff} , then action #1 is taken to reduce power consumption by estimating that user profile is expected to stay on hold.

6.3.3 Intuitive Solutions

Intuitive solutions either reduces the power consumption by decreasing DC or/and f_s or improves the accuracy in user state recognition by increasing them. Relevant adjustments are regulated by the action set $a(t)$. There are three different intuitive solutions proposed as follows:

6.3.3.1 Method I

This method tries to change DC in the first place rather than to change f_s . Let the pairs of DC and f_s lie over a space S^{R^2} which is defined in a matrix of $\{DC, f_s\} \rightarrow \{l, k\}, l \in DC, k \in f_s$. Method I proposes how to wander over the defined space according to actions by

$$S^{R^2}(l, k) = \begin{cases} S^{R^2}(l-1, k), & a = 1, l \neq l_{min} \\ S^{R^2}(l, k-1), & a = 1, l = l_{min}, k \neq k_{min} \\ S^{R^2}(l+1, k), & a = 3, l \neq l_{max} \\ S^{R^2}(l, k+1), & a = 3, l = l_{max}, k \neq k_{max} \\ S^{R^2}(l, k), & otherwise \end{cases} \quad (6.29)$$

Whenever an adjustment is needed in the sensory operation, the current index of S^{R^*} goes up or down by moving through different duty cycles until it gets to the boundaries, then it goes left or right by moving through different sampling frequencies.

6.3.3.2 Method II

This method, in contrast to Method I, makes the adjustments in f_s in the first place. Then, the relevant state transitions over S^{R^2} becomes

$$S^{R^2}(l, k) = \begin{cases} S^{R^2}(l, k-1), & a = 1, k \neq k_{min} \\ S^{R^2}(l-1, k), & a = 1, k = k_{min}, l \neq l_{min} \\ S^{R^2}(l, k+1), & a = 3, k \neq k_{max} \\ S^{R^2}(l+1, k), & a = 3, k = k_{max}, l \neq l_{max} \\ S^{R^2}(l, k), & otherwise \end{cases} \quad (6.30)$$

6.3.3.3 Method III

State transitions are executed according to the ascending order of power consumption rates shown in Fig. 6.5. The definition of (6.25) is then re-characterized as in $S^R = \text{ascend}(S^R)$. Hence,

both DC and f_s could be changed simultaneously.

$$S^R(r) = \begin{cases} S^R(r-1), & a = 1, i \neq r_{min} \\ S^R(r+1), & a = 3, i \neq r_{max} \\ S^R(r), & otherwise \end{cases} \quad (6.31)$$

In summary, the intuitive solutions (6.29), (6.30) and (6.31) regulate p^R , and hence affect the evolution of (6.28).

6.3.4 Constrained Markov Decision Process (CMDP)

Constrained Markov Decision Process (CMDP) is applied into sensor management system by setting a Markov-optimal policy u . This policy controls sensory sampling operations by deciding which pair of DC and f_s to be assigned in the sampling process, and it randomizes the decisions over given actions.

The CMDP parameter set is provided as follows:

- *Decision Epochs, O* : Decisions are the outputs obtained from the context inference framework (see Section 6.2.2).
- *State Space, S^R* : The system state space is given by (6.25).
- *Action Space, A* : The system action space is given by (6.24).
- *State Transition Probability, P_{rw}^a* : This probability matrix defines transition probabilities among states $\{r \rightarrow w\}$ while action a is taken.

$$P_{rw}^a = \begin{cases} \frac{1}{r-1}, & a = 1, w < r \\ 1, & a = 2, w = r \\ \frac{1}{W-r}, & a = 3, w > r \\ 0, & otherwise. \end{cases} \quad (6.32)$$

Remark that all transitions from a specific state is set an equal probability according to the rule of actions.

- *Accuracy Cost, $c(r,a)$* : The accuracy cost is the retrieved error rate in user state recognitions while the context inference framework is running, which is defined by ϕ in (6.22).

$$c(r, a) = 1 - \phi_r^a. \quad (6.33)$$

On the other hand, the default settings for the accuracy cost is ruled by the rate of missing sampling points under different system states where $\{(S = r)\} \rightarrow \{(DC = l) \times (f_s = k)\}$ and $\forall a \in A$:

$$c(r, a)_{default} = c(\{l, k\}, a) = 1 - l + l \frac{k}{k_{max}}. \quad (6.34)$$

Remark that the default settings are the maximum error rates indeed.

- *Power Consumption, $d(r,a)$* : Power consumption rate is the reward process ψ_r :

$$d(r, a) = \psi_r, \quad \forall a \in A. \quad (6.35)$$

The policy aims to maximize the accuracy in user state recognitions subjected to the power constraints. Therefore, a CMDP distinguishes from a regular MDP in the added power consumption function d , which is related to the constraints V_y where $y \in [1, Y]$.

$\rho(r, a)$ is denoted in CMDP as the occupation measure by specifying the probability of a relevant state-action pair in the decision process which satisfies given constraints, whose probability distribution is given by

$$f(\gamma, u, r, a) = \sum_{t=1}^{\infty} Pr_{\gamma}^u(S_t^R = r, A_t = a), \quad (6.36)$$

where γ and u are defined any initial distribution and any stationary policy.

Having (6.32), (6.33), (6.34) and (6.35), the constrained optimization problem is given by the following requirements:

$$\begin{aligned} \min_{\rho} \left\{ \sum_r \sum_a \rho(r, a) c(r, a) \right\} \text{ subject to} \\ \left\{ \begin{array}{l} \sum_r \sum_a \rho(r, a) (\delta_w(r) - P_{rw}^a) = 0, \\ \sum_r \sum_a \rho(r, a) = 1, \\ \rho(r, a) \geq 0, \end{array} \right. \end{aligned} \quad (6.37)$$

where $\forall r, w \in S^R, \forall a \in A, \delta_w(r) = \{1, r = w; 0, \text{ otherwise}\}$.

Let u be the optimal policy that satisfies for all i, a :

$$u_r(a) = \frac{\rho(r, a)}{\sum_a \rho(r, a)}, \quad \forall r \in S, \forall a \in A. \quad (6.38)$$

whenever the denominator is non-zero. It is concluded from (A.1) in Appendix A that ρ equals to $\gamma(I - P(u))^{-1}$ like defined in (6.37), and hence to (6.36), where I is the identity matrix.

In addition, the following constraints are added into (6.37):

$$\sum_i \sum_a \rho(r, a) d^y(r, a) \leq V_y, \quad y = 1, \dots, Y. \quad (6.39)$$

where $V_y(t) = (1 \pm \nu)V_y(t - 1)$ is given for the constraint according to which action is taken, such as $\{a = 1 : -\nu\}$ and $\{a = 3 : +\nu\}$ where $0 < \nu < 1$, and $\{a = 2 : \nu = 0\}$.

Finally, the constrained optimization problem is defined from (6.37) and (6.39) as $\{\min c \text{ subject to } d^y \leq V_y\}$, whose solution is described in [183, 184], and solved based on linear programming as follows: Find the minimum $C^* \in C(\gamma, u)$ for any $u, \rho \in f(\gamma, u)$, $C(\gamma, u) = C(\rho(u))$ and each $D^y(\gamma, u) = D^y(\rho(u))$, see (A.2) and (A.3) in Appendix A for the definition of $C(\gamma, u)$ and $D(\gamma, u)$ respectively.

Under the policy u from (6.38), the expected average accuracy and power consumption cost is defined by

$$\mathbb{E}^u[C] = \frac{1}{n} \sum_{n'=1}^n \mathbb{E}^u c_{n'}(i, a), \quad (6.40a)$$

$$\mathbb{E}^u[V] = \frac{1}{n} \sum_{\forall y} \sum_{n'=1}^n \mathbb{E}^u d_{n'}^y(i, a), \quad (6.40b)$$

where n' and n are instant and total decision epoch times respectively.

6.3.5 Partially Observable Markov Decision Process (POMDP)

Partially Observable Markov Decision Process (POMDP) also describes an optimal solution to respond to the defined tradeoff. The parameter set by POMDP has some similarities like the one provided by CMDP. The same states S^R , actions a , state transitions P_{rw}^a are used in this model as well. A POMDP relies on an agent which takes some action $a \in A$, and hence makes the system move

from state r to a new state w . Due to the uncertainty in an action, the state transition is modeled by P_{rw}^a . In addition, the agent makes an observation $x \in O$ to gather information for the decision on the new system state selection, thereby, the state-observation relationship is probabilistically modeled by Z_{wx}^a . In each observation epoch, the agent takes action a in state r , then receives a reward $R(r, a)$. The POMDP parameter set is given as follows:

- Decision Epochs O , State Space S^R , Action Space A and State Transition Probability P_{rw}^a are given the same like in Section 6.3.4.
- *Observation Emission Probability*, Z_{wx}^a : The observation is the accuracy rate provided by the context inference framework (see Section 6.2.2).

$$Z_{wx}^a(t) = \frac{1}{|Z|} \begin{cases} \phi(t), & r = w, \\ (1 - \phi(t)), & r \neq w, \end{cases} \quad (6.41)$$

where $S_{(t-1)}^R = r, S_t^R = w, \forall a \in A, |Z| = \phi(t) + (W - 1)(1 - \phi(t))$, and $x = 1$ since there is only one observation, which is the accuracy rate.

- *Reward Function*, $R_r^a(t)$: The reward process (i.e., power consumption) ψ_r is defined in Section 6.3.1:

$$R_r^a(t) = \psi_r, \quad \forall a \in A. \quad (6.42)$$

- *Belief Vector*, $\lambda_r^a(t)$: Since the internal state of the underlying POMDP is not directly observable, the knowledge of the internal state based on the history of all past decisions and observations could be provided by a belief vector $\lambda_r^a(t) \in \Lambda$, which gives the conditional probability of being in state r under action a prior to any state transition.

The belief vector is updated whenever a new knowledge coming in after incorporating the action and observation obtained at time t within the history set of $\mathcal{H}(t) = \{a(\tau), O(\tau)\}, \tau \in [1, t]$. The updated belief vector is obtained using (6.41) by the Bayes rule:

$$\lambda_w^a(t+1) = \mathcal{T}(\lambda(t) | a, O) = \frac{Z_{wx}^a \sum_i P_{rw}^a \lambda_r^a(t)}{\sum_x Z_{wx}^a \sum_r P_{rx}^a \lambda_r^a(t)}. \quad (6.43)$$

The goal defined by OPMDP is to develop an opportunistic sensor sampling strategy which seeks for a favorable trade-off balance between the accuracy in sensing and energy efficiency. Hence,

a sensing policy $u : \Lambda \rightarrow A$ is defined to map a belief vector λ_r to an action a . The policy u is presented by a sequence of functions $\{u = [\eta_1, \eta_2, \dots, \infty]\}$ where η_t maps a belief vector $\lambda_r(t) \in \Lambda$ to an action $a \in A$ at time t over infinite horizon of POMDP.

From the time at the current belief vector is $\lambda(t)$, a value function $V_t(\lambda(t))$ is denoted to represent the minimum expected remaining reward which can be earned under the assigned policies. This reward is obtained through immediate and future rewards. The optimal policy strikes a balance between earning immediate reward and obtaining a lean toward future decisions on the system.

The optimal strategy aims at minimizing the expected total reward, and it is defined together with (6.42) and (6.43) as in

$$u = \arg \min_u \mathbb{E}_u \left[\sum_{t=1}^T R_r^a(t) \mid \lambda(1) \right]. \quad (6.44)$$

Hence, the value function for total reward aggregation is given with the help by (6.44) as in

$$V_t(\lambda(t)) = \min_a \mathbb{E} [R^a(\lambda(t)) + \varphi V_{t+1}(\mathcal{T}(\lambda(t) \mid a, S_r^a(t)))], \quad (6.45)$$

where $R^a(\lambda) = \sum_r \lambda_r R_r^a$ and $\varphi \in [0, 1]$ is a discount factor.

Due to the impact of the current action on the future rewards, the uncountable belief state because of the process lies over infinite horizontal space, and the non-stationarity of the optimal policy, finding an optimal strategy for a POMDP is often computationally prohibitive.

6.3.5.1 Myopic Strategy and Sufficient Statistics

Since the finding of an optimal strategy is computationally restricted, it is crucial to exploit the available POMDP and develop suboptimal strategies in order to reduce the complexity. Therefore, it is needed to show the *a posterior* of distribution of the belief vector under sufficient statistics. The belief vector (6.43) is then updated based on the chosen action under the following sufficient statistics:

$$\lambda_r(t+1) = \begin{cases} (\lambda(t)_r^T P_{rw}^a)^T, & a' = \{1, 3\}, r \neq w, \\ \lambda_i(t), & a' = 2, r = w, \\ 0, & \text{otherwise,} \end{cases} \quad (6.46)$$

where T is matrix transpose.

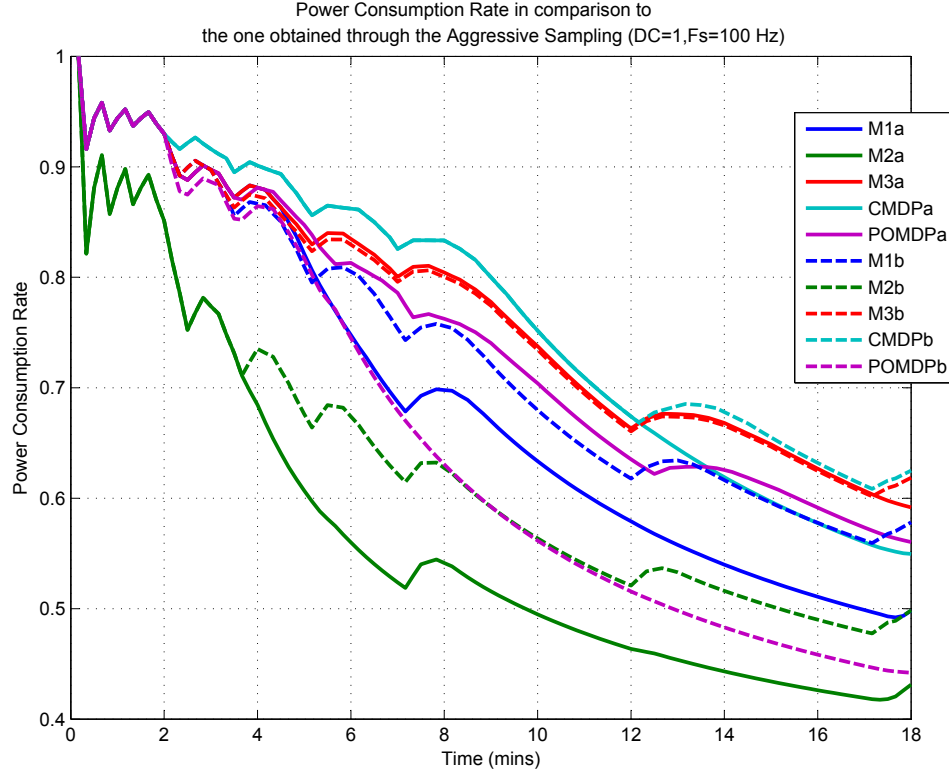


Figure 6.6: Power consumption rate in response to user profile

In addition, a myopic policy is introduced to ignore the impact of the current action on the future rewards by focusing only minimizing the immediate reward since power consumption caused by instant sensory operation settings does not rely on future diversity in the sensory operation methods. Thereby, the myopic policy makes $\varphi = 0$ in (6.45), and hence it turns (6.44) into:

$$S_*^R = \arg \min_r \mathbb{E}[R_r^a(\lambda_r^a(t))], \quad (6.47)$$

subjects to $\max(Z_{w,O=x}^a(t)) > \varepsilon,$

where S_*^R is the chosen optimal state, and ε denotes the minimum probability of accuracy allowed by the process so the accuracy is constrained above this probability.

6.4 Performance Analysis

A case study in a HAR model is examined in order to investigate the defined tradeoff by proposed sensor management methods. A same user activity profile is examined for each tradeoff

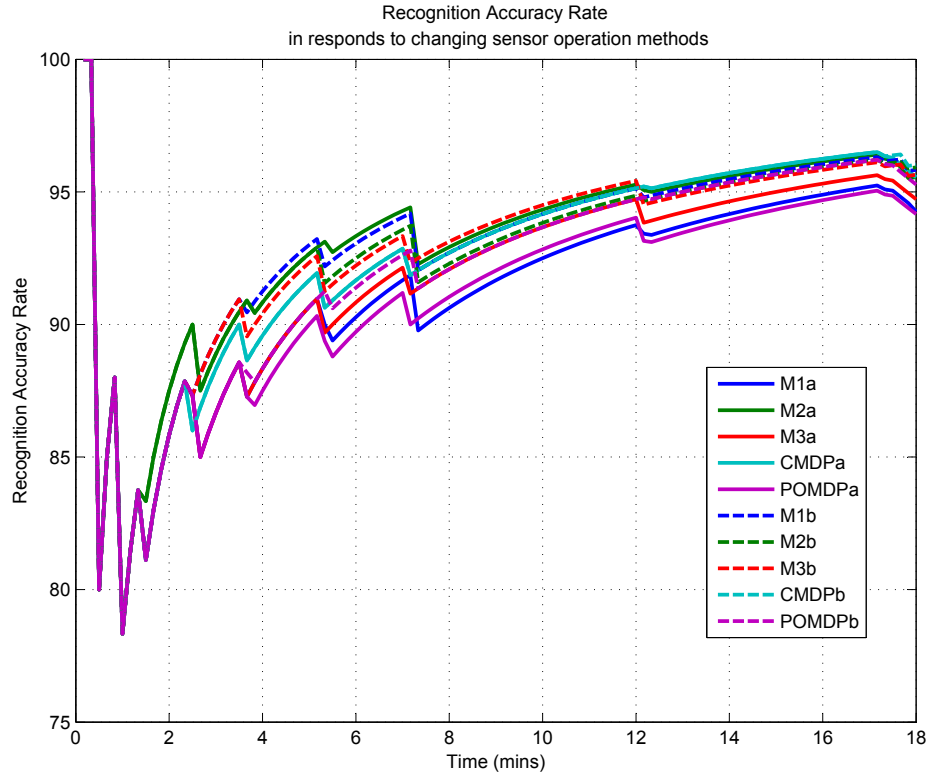


Figure 6.7: Recognition accuracy rate in response to user profile

analysis. Accordingly, the HAR based user activity profile begins with a random activity pattern of user states ‘sitting’ and ‘standing’ for a minute, then each state transit into the another one in the end of the following sojourn times of $\{5, 10, 30, 60, 100, 300\}$ seconds. The initial one minute interval is considered for adaptation process carried out by the context inference framework in order to set required adjustments in the system parameters towards to the activity pattern. Note that the system parameters have been already set to the default system settings initially by (6.14). From that point, the sensory operation parameters, which are duty cycle and sampling frequency, are updated with a 10 second of period. In addition, t_{suff} is set to 20 seconds. Recall that as long as the continuing settlement time in any state gets longer than t_{suff} , the sensor management system decreases the power consumption, which jeopardizes the recognition accuracy of the activity pattern. The default sensory operation parameters are set to the aggressive sampling method, which is equal to the pair of $\{100\%, 100 \text{ Hz}\}$ for $\{DC, f_s\}$.

The context inference framework recognizes a user state with a period of one second under the aggressive sampling method. The underlying Markovian chain in the framework has a finite

horizon length of 60, which means a minute long recent history of user states. Every one minute, system parameters are updated according to (6.18). Except for the aggressive sampling method in sensory operations, the context inference framework may not have a sensory observation at any time. For instance, in a case where the pair of {50%, 50 Hz} is selected for the sensory operation parameters, the decision period to recognize a user state is extended to 4 seconds, which results in having 3 empty decision points to estimate missing user state recognitions.

The tradeoff analysis is carried out for each sensor management method. The tradeoff solutions by each method are shown in Fig. 6.6 for the analysis of power consumption rate according to (6.28), (6.40b) and (6.45), and Fig. 6.7 for the analysis of recognition accuracy according to (6.22), (6.40a) and (6.47). Also, the tradeoff solutions by each method in both figures are noted by the suffixes 'a' and 'b' to demonstrate without/with some constraints added. The suffix 'a' indicates the actual sensor management methods without any additional constraints. However, the suffix 'b' sets extra rules on the methods. First, a 10% tolerance value is added into Method I, II and III to constrain the recognition accuracy rate, which help the prevention of drastic recognition errors. If the constraint is exceeded, the sensory operation parameters are forced to set the default settings, i.e, the aggressive sampling. Second, for CMDP, there is another constraint set on the power consumption rate to control the tendency of the decision process to take an immoderate decrease in power consumption. According to the regulation, the sensory operation must stay in $\pm 25\%$ of the present power consumption level. Finally, for POMDP, the update process of the belief vector is reconfigured by adding the feature of $\lambda_i(t+1) = 1$ where $a = 2$.

In light of the explanations above, the following discussions can be made through both figures:

- Recognition accuracy rate decreases significantly during the initial progress of context inference framework due to the adaptation to the user by the framework is not adequate, and the user exhibits a variant activity profile by changing user states regularly. Hence, the power consumption rate shows a zig-zag line in Fig. 6.7 by taking an increase to compensate the worsening recognition accuracy rate, and then by taking a decrease to benefit from the adaptation feature if possible.

- When the time passes by, since the framework gets a better adaptation to the user activity profile, the recognition accuracy rate increases even though power consumption rate decreases.
- Whenever a state transition occurs after a long visiting time in a user state or the number of estimation in user state recognitions increases after a slower sampling policy is attained or the entropy rate cannot converge into any point, the recognition accuracy rate may show a slight decrease; however, the framework will fix the accuracy rate if a drastic decrease is observed.
- Amongst intuitive solutions, a comparison can be made by $M3 > M1 > M2$ in terms of the power consumption rate, and by $M1 > M3 > M2$ in terms of the recognition accuracy rate. Results show that the sampling in slower frequencies consume higher power than the sampling in lower duty cycles; however, in terms of the recognition accuracy, it shows the opposite assumption. It is because that sampling in slower frequencies still obtain information about user activity where the sampling in lower duty cycles cannot do. On the other hand, M3 has the highest power consumption since it switches sensor operation modestly while achieving a fine accuracy in user state recognitions.
- $MXb > MXa$ where $X=I, II, III$ and $CMDPb > CMDPa$ are met in terms of the power consumption rate due to the aggressive sampling is forced to apply in case where severe errors occur in user state recognitions.
- $POMDPb$ makes a clear conclusion about the belief vector rather than $POMDPa$ does when a sufficient visiting time elapsed on a specific user state. Hence, the power consumption decreases since the conclusion notifies the continuity of the same user state.
- $MIIIa$ responds in a similar way what $CMDPa$ and $POMDPa$ have while trying to reach their optimal policies.

In general, the tradeoff solutions achieve overall 50% enhancement in power consumption caused by the physical sensor work with respect to overall 10% decrease in accuracy rate for user state recognitions thanks to the context inference framework.

To sum up, this chapter presents a novel comprehensive framework within the futuristic concept of context-awareness in mobile sensing. A statistical machine based context inference model

together with an intelligent sensor management system is created in order to recognize human-centric activities by aiming at achieving a tolerable power consumption caused by the total process. The study takes the smartphone accelerometer sensor into the scope to show the effectiveness of the proposed approaches as well as leaving the door open for future improvements in the functionality of other smartphone sensors.

While creating the statistical machine, some features are taken into the consideration, such as time-varying user activity profile, system adaptability to the changing profile, nonuniform time distribution of sensory sampling process due to the power saving precautions, and the estimation process where missing sensory observations exist. On the other hand, while creating the sensor management system, the analytical modeling of power consumption caused by the accelerometer is examined so that by the collaboration with the statistical machine, a better balance is achieved for the defined tradeoff throughout the chapter. For the tradeoff analysis, some intuitive and optimal sensory operation solutions are provided in order to increase efficiency in the battery power consumption; whereas the statistical machine tries to maintain the accuracy rate provided by the framework.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 The List of Contributions

- Chapter 2: This chapter surveys the context-awareness in ubiquitous sensing. It provides a comprehensive introduction to context definition, context presentation, context modeling and context inference. It also points out the importance of a context-aware middleware design and the challenges that are faced while designing process and system integration. Finally, the chapter summarizes context aware applications according to the interested context, and gives current trends in this area.
- Chapter 3: A sensory data has its own signal characteristics. Instead of applying computationally harsh signal processing techniques toward any sensory data, a sufficient number of recognition algorithms must be considered to find out the best context-exploiting methods. In this sense, this chapter proposes an online solution which intends to perfectly exploit acceleration signals within the fast Decision Tree (DT) classifiers without setting any predefined/fixed thresholds over any specific acceleration spaces in order to differentiate user activities, such as sitting, standing, walking, running, and transition among activities. The proposed classification method provides the following properties:
 1. *unsupervised learning*: no priori information, no fixed thresholds, no initial training data classes;
 2. *adaptive*: robust solution to a changing orientation of the device;
 3. *light-weight*: efficient tree-based classification by applying sufficient signal processing toolbox: no redundant computational workload;
 4. *online*: instant context inference;
 5. *assisting*: working standalone and/or assisting other classification algorithms by creating training data classes or input matrices;

6. *updating*: computational efficient update/add/delete process on training data classes.

The chapter also proposes a context monitoring system which controls required processes for context analysis in a such way that redundant computational repetitions of recognizing the same contextual information in a sequence of time can be avoided.

- Chapter 4: In this chapter, a Hidden Markov Model (HMM) based framework is proposed in order to represent user states as outcome of either a recognition or/and estimation model for context-aware applications. The framework also applies a sensor power management method, such as adaptive sampling, in order to prolong the device battery lifetime. Therefore, user states are either recognized as an inference of actual sensor readings or as an estimation by HMM. Adaptive sampling changes operational behavior of a sensor by setting different sampling frequencies. With this method, sleeping time for a sensor can be extended by entering its idle operating mode frequently. While the sensor is in idle mode, the user might either keep or change his/her current activity. In this sense, a statistical model is required to track time-variant user activity profiles in order to predict the best likely user state that fits into instant user behavior.

There are a few distinctive points provided by this chapter:

1. The framework accepts that user behavior changes in time; thereby, it assumes that HMM parameters are inhomogeneous (i.e., time-variant).
2. Every user differs from each other and each one responds uniquely while producing an activity pattern. Thus, a desired system cannot have predefinitions for each activity profile. User activity profile also changes in time which forces the system to make proper adjustments accordingly in order to make accurate user state recognitions. Thereby, adaptability problem is introduced for inhomogeneous user behaviors, and a relevant solution is given by the uncertainty of entropy rate. It is shown that the convergence of entropy rate could be used to notify a specific user behavior.
3. Effective online pattern recognition algorithms for context extraction through sensory data are adopted from Chapter 3 in order to infer relevant activities within the provided context and to conduct the observation analysis of HMM.

4. By changing sampling frequencies in sensory operations, the trade-off analysis is examined and the framework is validated. The connection between the context inference under different sampling frequencies and its effect on the convergence of entropy rate is discovered.
 5. In case of occurrences of missing observations, a desired system must estimate missing observations to make sure that accuracy provided by the system decisions are maintained. Missing observations are found by Naive Bayesian approach within HMMs.
- Chapter 5: This chapter studies the battery modeling under the scope of the battery non-linearities with respect to variant discharge profiles. In addition, the energy consumption behavior of smartphone sensors are analytically modeled, especially a real time application is carried out for the accelerometer sensor to investigate how any change in sensory operations affects the power consumption profiles triggered by the sensor and depletion on the battery. Also, energy consumption profiles are created by assigning different pair of duty cycles and sampling frequencies during the sensory operations. Finally, a Markov-reward process is integrated in order to model the energy consumption profiles and represents the cost by each profile as an accumulated reward. The accumulated reward, i.e., total energy consumption, is also linked to the battery modeling to make a connection between the usage pattern on sensors and the battery behavior. With understanding of the non-linear battery behavior with respect to diverse operation methods in sensors, a tolerable power consumption balance could be achieved while employing context aware services in resource constrained mobile devices.

There are a few distinctive points provided by this chapter:

1. Studying mobile device based battery modeling, which adopts Kinetic Battery Model (KiBaM), under the scope of battery non-linearities with respect to variant load profiles.
2. Modeling energy consumption behavior of accelerometers analytically as an example.
3. A Markov reward process is integrated in order to create energy consumption profiles, which consist of different pair of duty cycles and sampling frequencies

in sensor operations, and represent the total energy cost by each profile as an accumulated reward in the process.

4. Presenting the linkage between different usage patterns on the accelerometer sensor and the battery.

5. It is intended that modeling the battery nonlinearities together with the understanding the effects of different usage patterns in sensor operations on the power consumption and the battery discharge may lead to discover optimal energy reduction strategies in order to extend the battery lifetimes and help a continual improvement in context-aware mobile services.

- Chapter 6: In this chapter, a novel comprehensive framework is presented within the futuristic concept of context-awareness in mobile sensing. A statistical machine based context inference model together with an intelligent sensor management system is created in order to recognize human-centric activities by aiming at achieving a tolerable power consumption caused by the total process. The study takes the smartphone accelerometer sensor into the scope to show the effectiveness of the proposed approaches as well as leaving the door open for future improvements in the functionality of other smartphone sensors. The ultimate goal of this chapter is to recognize user activities accurately and to make sure power efficiency is maximized. Most importantly, this research intends to create and clarify an effective HMM-based framework included with power saving methods at the low-level sensor operations in order to guide the development of future context-aware applications.

While creating the statistical machine, some features are taken into the consideration, such as time-varying user activity profile, system adaptability to the changing profile, nonuniform time distribution of sensory sampling process due to the power saving precautions, and the estimation process where missing sensory observations exist. On the other hand, while creating the sensor management system, the analytical modeling of power consumption caused by the accelerometer is examined so that by the collaboration with the statistical machine, a better balance is achieved for the defined tradeoff throughout the chapter. For the tradeoff analysis, some intuitive and optimal sensory operation solutions are provided in order to increase efficiency in the battery power con-

sumption; whereas the statistical machine tries to maintain the accuracy rate provided by the framework.

7.2 Research Highlights

- Light-weight online unsupervised posture detection by smartphone accelerometer as an example of physical context inference
 - Standalone mode
 - Assisting mode for widely used supervised classification methods by providing training data classes/system matrices
 - * Helping them to be used for online processing by providing computational light-weight class/training sample update
- Studying smartphone battery non-linearities for variant loads by changing sampling periods and duty cycles
- Studying accelerometer sensor in terms of power consumption model with respect to variant sampling strategies
- Presenting the linkage between different usage patterns on the accelerometer sensor and the battery discharge
- Creating and clarifying an effective HMM based framework included with context inference methods to guide the development of real-time operating user-oriented future context-aware applications
 - No prior/fixed user behavior definition/context inference
 - Adaptability to time-invariant user behaviors
 - Adaptive and sub-optimal sampling policies
 - Robust inference solution with respect to unexpected/changing sampling intervals
 - Handling missing observations due to sensor saving methods
 - Providing a fine balance for the trade-off

7.3 Future Works

This dissertation provides solutions on emerging problems on context-awareness using one sensor model; whereas, multiple sensor utilization may provide a similar context. In such cases, the fusion of sensors need to be applied while constructing context aware frameworks like designed in Chapter 4 and 6. In addition, a sensor management system needs to be created to dynamically select a sufficient number of sensors while inferring a context. To achieve power efficiency with respect battery non-linearities and sensors' behaviors examined in Chapter 5, the system has to put sensors into an order according to their power consumption level and application relevance. Depending on which user activity profile is active, a different set of sensors is selected. Moreover, like introduced in Chapter 3, the definition of user states can be extended by detecting more postural actions. Especially, light-weight online unsupervised based detection algorithms have to be discovered for any mobile device based sensor to be able to enhance the working of context-aware applications in resource-limited mobile computing environments.

REFERENCES

- [1] R. Rao, S. Vrudhula, and D. N. Rakhmatov, "Battery modeling for energy-aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, Dec. 2003.
- [2] D. N. Rakhmatov and S. B. K. Vrudhula, "An analytical high-level battery model for use in energy management of portable electronic sys." in *Int'l Conf. on Computer-aided design (ICCAD)*. IEEE Press, 2001, pp. 488–493.
- [3] D. Rakhmatov, S. Vrudhula, and D. A. Wallach, "A model for battery lifetime analysis for organizing applications on a pocket computer," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 11, no. 6, pp. 1019–1030, Dec. 2003.
- [4] D. Rakhmatov and S. Vrudhula, "Energy management for battery-powered embedded sys." *ACM Trans. on Embedded Computing Sys.*, vol. 2, pp. 277–324, 2003.
- [5] A. Wood, J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru, "Context-aware wireless sensor networks for assisted living and residential monitoring," *IEEE Network*, vol. 22, no. 4, pp. 26–33, july-aug. 2008.
- [6] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas, "Emotionsense: a mobile phones based adaptive platform for experimental social psychology research," ser. Ubicomp '10. ACM, pp. 281–290.
- [7] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [8] J. Zheng, D. Simplot-Ryl, C. Bisdikian, and H. Mouftah, "The internet of things," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 30–31, november 2011.
- [9] M. Kranz, A. Maller, N. Hammerla, S. Diewald, T. Platz, P. Olivier, and L. Roalter, "The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices," *Pervasive and Mobile Computing*, 2012.
- [10] A. Madan, M. Cebrian, S. Moturu, K. Farrahi, and A. Pentland, "Sensing the health state of a community," *IEEE Pervasive Computing*, vol. 11, no. 4, pp. 36–45, oct.-dec. 2012.
- [11] B. Priyantha, D. Lymberopoulos, and J. Liu, "Littlerock: Enabling energy-efficient continuous sensing on mobile phones," *Pervasive Computing, IEEE*, vol. 10, no. 2, pp. 12–15, april-june 2011.
- [12] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *MobiSys '09*, pp. 179–192.
- [13] O. Yurur, C. H. Liu, and W. Moreno, "Light-weight online unsupervised posture detection by smartphone accelerometer," *submitted to IEEE Transactions on Emerging Topics in Computing*, 2013.

- [14] —, “Unsupervised posture detection by smartphone accelerometer,” *EIT Electronics Lett.*, vol. 49(8), 2013.
- [15] O. Yurur and W. Moreno, “Energy efficient sensor management strategies in mobile sensing,” in *ISTEC General Assembly*, may 2011.
- [16] O. Yurur, M. Labrador, and W. Moreno, “Adaptive and energy efficient context representation framework in mobile sensing,” *accepted to appear in IEEE Trans. on Mobile Comput.*, 2013.
- [17] O. Yurur, C. H. Liu, and W. Moreno, “Analytical modeling of energy consumption for context-aware smartphone sensing,” *submitted to J. Electrochem. Soc.*, 2013.
- [18] —, “Adaptive sampling and duty cycling for smartphone accelerometer,” in *submitted to IEEE MASS*, 2013.
- [19] O. Yurur, C. H. Liu, X. Liu, K. K. Leung, and W. Moreno, “Energy efficient and context-aware sensor employment in mobile sensing,” *submitted to IEEE Trans. on Mobile Comput.*, 2013.
- [20] M. Sama, D. S. Rosenblum, Z. Wang, and S. Elbaum, “Model-based fault detection in context-aware adaptive applications,” in *SIGSOFT Int’l Symposium on Foundations of software Eng.* ACM, 2008, pp. 261–271.
- [21] R. Tan, J. Gu, Z. Zhong, and P. Chen, “Metadata management of context resources in context-aware middleware system,” *Web Information Sys. and Mining*, pp. 350–357, 2012.
- [22] J. Gu, “Middleware for physical and logical context awareness,” *Signal Processing, Image Processing and Pattern Recognition*, pp. 366–378, 2011.
- [23] C. Bettini, O. Brdiczka, K. Henriksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, “A survey of context modelling and reasoning techniques,” *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, 2010.
- [24] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context-aware sys.” *Int’l Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.
- [25] T. Gu, H. Pung, and D. Zhang, “Toward an osgi-based infrastructure for context-aware applications,” *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 66 – 74, oct.-dec. 2004.
- [26] J. Yang, H. Lu, Z. Liu, and P. P. Boda, “Physical activity recognition with mobile phones: Challenges, methods, and applications,” *Multimedia Interaction and Intelligent User Interfaces*, pp. 185–213, 2010.
- [27] X. Zhu, “Semi-supervised learning literature survey,” 2005.
- [28] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song, “Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments,” in *MobiSys ’08*. ACM, pp. 267–280.
- [29] M. Quwaider and S. Biswas, “Body posture identification using hidden markov model with a wearable sensor network,” pp. 19:1–19:8.
- [30] M. Ermes, J. Parkka, and L. Cluitmans, “Advancing from offline to online activity recognition with wearable sensors,” in *Int’l Conf. of the IEEE Eng. in Medicine and Biology Society (EMBS)*, aug. 2008, pp. 4451 –4454.

- [31] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity," in *Int'l Conf. of the IEEE Eng. in Medicine and Biology Society (EMBS)*, aug. 2008, pp. 5250–5253.
- [32] E. Miluzzo, N. D. Lane, H. Lu, and A. T. Campbell, "Research in the app store era: Experiences from the cenceme app deployment on the iphone."
- [33] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Trister, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *European Conf. on Wireless Sensor Networks*. Springer-Verlag, 2008, pp. 17–33.
- [34] Y.-P. Chen, J.-Y. Yang, S.-N. Liou, G.-Y. Lee, and J.-S. Wang, "Online classifier construction algorithm for human activity detection using a tri-axial accelerometer," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 849–860, 2008.
- [35] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Int'l Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, april 2006, pp. 4 pp. –116.
- [36] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *IEEE Int'l Conf. on Robotics and Automation (ICRA)*, may 2009, pp. 2154–2159.
- [37] M. Berchtold, M. Budde, D. Gordon, H. Schmidtke, and M. Beigl, "Actiserv: Activity recognition service for mobile phones," in *Int'l Symposium on Wearable Computers (ISWC)*, oct. 2010, pp. 1–8.
- [38] T. Kao, C. Lin, and J. Wang, "Development of a portable activity detector for daily activity recognition," in *IEEE Int'l Symposium on Industrial Electronics (ISIE)*, 2009, pp. 115–120.
- [39] M. Berchtold, M. Budde, D. Gordon, H. Schmidtke, and M. Beigl, "Actiserv: Activity recognition service for mobile phones," *Wearable Computers (ISWC)*, pp. 1–9, 2010.
- [40] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell, "Darwin phones: the evolution of sensing and inference on mobile phones," ser. *MobiSys '10*. ACM, pp. 5–20.
- [41] D. Peebles, H. Lu, N. D. Lane, T. Choudhury, and A. T. Campbell, "Community-guided learning: Exploiting mobile sensor users to model human behavior." in *AAAI*. AAAI Press, 2010.
- [42] P. Siirtola and J. Rning, "Recognizing human activities user-independently on smartphones based on accelerometer data." *IJIMAI*, no. 5, pp. 38–45.
- [43] L. T. Vinh, S. Lee, H. X. Le, H. Q. Ngo, H. I. Kim, M. Han, and Y.-K. Lee, "Semi-markov conditional random fields for accelerometer-based activity recognition," *Applied Intelligence*, vol. 35, no. 2, pp. 226–241, Oct. 2011.
- [44] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Proc. of Pervasive*, 2006, pp. 1–16.
- [45] Z.-Y. He and L.-W. Jin, "Activity recognition from acceleration data using ar model representation and svm," in *Int'l Conf. on Mach. Learning and Cybernetics*, vol. 4, july 2008, pp. 2245–2250.

- [46] D. Riboni and C. Bettini, “Cosar: hybrid reasoning for context-aware activity recognition,” *Personal Ubiquitous Comput.*, vol. 15, no. 3, pp. 271–289, Mar. 2011.
- [47] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, “Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations,” in *Int’l Conf. on Ubiquitous intelligence and computing (UIC)*. Springer-Verlag, 2010, pp. 548–562.
- [48] Z. He and L. Jin, “Activity recognition from acceleration data based on discrete cosine transform and svm,” in *IEEE Int’l Conf. on Sys., Man and Cybernetics (SMC)*, oct. 2009, pp. 5041–5044.
- [49] Y. Hanai, J. Nishimura, and T. Kuroda, “Haar-like filtering for human activity recognition using 3d accelerometer,” in *DSP/SPE’ 09*, jan. 2009, pp. 675–678.
- [50] D. Minnen, T. Westeyn, D. Ashbrook, P. Presti, and T. Starner, “Recognizing soldier activities in the field,” in *Int’l Workshop on Wearable and Implantable Body Sensor Networks (BSN)*. Springer, 2007, pp. 236–241.
- [51] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, “Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor,” in *IEEE Int’l Symposium on Wearable Computers (ISWC)*. IEEE Computer Society, 2007, pp. 1–4.
- [52] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
- [53] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data.” Springer, 2004, pp. 1–17.
- [54] D. Bandyopadhyay and J. Sen, “Internet of things: Applications and challenges in technology and standardization,” *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, 2011.
- [55] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, “A middleware infrastructure for active spaces,” *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 74–83, 2002.
- [56] J.-y. Hong, E.-h. Suh, and S.-J. Kim, “Context-aware sys.: A literature review and classification,” *Expert Syst. Appl.*, vol. 36, pp. 8509–8522, May 2009.
- [57] K. Kakousis, N. Paspallis, and G. A. Papadopoulos, “A survey of software adaptation in mobile and ubiquitous computing,” *Enterprise Information Sys.*, vol. 4, no. 4, pp. 355–389, 2010.
- [58] K. E. Kjær, “A survey of context-aware middleware,” in *Conf. on IASTED Int’l Multi-Conf.: Software Eng.* ACTA Press, 2007, pp. 148–155.
- [59] C. Mascolo, L. Capra, and W. Emmerich, “Mobile computing middleware,” *Advanced lectures on networking*, pp. 506–510, 2002.
- [60] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. L. Wong, “Sensay: A context-aware mobile phone,” ser. ISWC ’03, pp. 248–.
- [61] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, “The jigsaw continuous sensing engine for mobile phone applications,” ser. SenSys ’10. ACM, 2010, pp. 71–84.
- [62] Z. Zhuang, K.-H. Kim, and J. P. Singh, “Improving energy efficiency of location sensing on smartphones,” ser. MobiSys ’10. ACM, 2010, pp. 315–330.

- [63] E. Jovanov and A. Milenkovic, "Body area networks for ubiquitous healthcare applications: Opportunities and challenges." *Journal of Medical Sys.*, p. 110, 2011.
- [64] P. Yan, I. Lin, M. Roy, E. Seto, C. Wang, and R. Bajcsy, "Wave and calfit; towards social interaction in mobile body sensor networks," in *Wireless Internet Conf. (WICON)*, march 2010.
- [65] E. M. Berke, T. Choudhury, S. Ali, and M. Rabbi, "Objective measurement of sociability and activity: Mobile sensing in the community," *The Annals of Family Medicine*, vol. 9, no. 4, pp. 344–350, 2011.
- [66] K. G. Stanley and N. D. Osgood, "The potential of sensor-based monitoring as a tool for health care, health promotion, and research," *The Annals of Family Medicine*, vol. 9, no. 4, pp. 296–298, 2011.
- [67] J. Blum and E. Magill, "M-psychiatry: Sensor networks for psychiatric health monitoring," in *Postgraduate Symposium The Convergence of Telecommunications, Networking and Broadcasting*, june 2008.
- [68] O. Lara and M. Labrador, "A mobile platform for real-time human activity recognition," in *IEEE Consumer Communications and Networking Conf. (CCNC)*, jan. 2012, pp. 667–671.
- [69] J. Choi and R. Gutierrez-Osuna, "Using heart rate monitors to detect mental stress," in *Int'l Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, june 2009.
- [70] S. Dhar and U. Varshney, "Challenges and business models for mobile location-based services and advertising," *Commun. ACM*, vol. 54, no. 5, pp. 121–128, May 2011.
- [71] H. Yoon, Y. Zheng, X. Xie, and W. Woo, "Social itinerary recommendation from user-generated digital trails," *Personal and Ubiquitous Computing*, vol. 16, pp. 469–484, 2012.
- [72] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Towards mobile intelligence: Learning from gps history data for collaborative recommendation," *Artif. Intell.*, vol. 184–185, pp. 17–37, June 2012.
- [73] L. Liao, D. Fox, and H. Kautz, "Extracting places and activities from gps traces using hierarchical conditional random fields," *Int. J. Rob. Res.*, vol. 26, no. 1, pp. 119–134, Jan. 2007.
- [74] X. J. Ban and M. Gruteser, "Towards fine-grained urban traffic knowledge extraction using mobile sensing," ser. UrbComp '12. ACM, pp. 111–117.
- [75] J. Krumm and E. Horvitz, "Predestination: inferring destinations from partial trajectories," ser. UbiComp '06. Springer-Verlag, pp. 243–260.
- [76] M. Kim and D. Kotz, "Extracting a mobility model from real user traces," in *IEEE INFOCOM*, 2006.
- [77] J. Herrera, D. Work, R. Herring, X. Ban, Q. Jacobson, and A. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment," *Transportation Research Part C*, vol. 18, no. 4, pp. 568–583, August 2010.
- [78] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," ser. SenSys '09. ACM, pp. 85–98.
- [79] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," ser. SenSys '08, pp. 323–336.

- [80] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, and M. Srivastava, “Biketastic: sensing and mapping for better biking,” in *SIGCHI '10*. ACM, 2010, pp. 1817–1820.
- [81] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, “The bikenet mobile sensing system for cyclist experience mapping,” ser. *SenSys '07*. ACM, pp. 87–101.
- [82] N. Husted and S. Myers, “Mobile location tracking in metro areas: malnets and others,” ser. *CCS '10*, 2010, pp. 85–96.
- [83] G.-S. Ahn, M. Musolesi, H. Lu, R. Olfati-Saber, and A. T. Campbell, “Metrotrack: predictive tracking of mobile events using mobile phones,” in *IEEE Int'l Conf. on Distributed Computing in Sensor Sys. (DCOSS)*. Springer-Verlag, 2010, pp. 230–243.
- [84] K. Shilton, “Four billion little brothers?: privacy, mobile phones, and ubiquitous data collection,” *Commun. ACM*, vol. 52, no. 11, pp. 48–53, Nov. 2009.
- [85] H. linh Truong and S. Dustdar, “A survey on context-aware web service sys.” 2009.
- [86] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, “Cenceme: injecting sensing presence into social networking applications,” in *European Conf. on Smart sensing and context (EuroSSC)*. Springer-Verlag, 2007, pp. 1–28.
- [87] A. Beach, M. Gartrell, X. Xing, R. Han, Q. Lv, S. Mishra, and K. Seada, “Fusing mobile, sensor, and social data to fully enable context-aware computing,” ser. *HotMobile '10*. ACM, pp. 60–65.
- [88] S. B. Mokhtar, L. McNamara, and L. Capra, “A middleware service for pervasive social networking,” in *Int'l Workshop on Middleware for Pervasive Mobile and Embedded Computing*. ACM, 2009, p. 2.
- [89] R. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *IEEE Communications Mag.*, vol. 49, no. 11, pp. 32–39, november 2011.
- [90] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, “Crowdsourcing with smartphones,” *IEEE Internet Computing*, vol. 16, no. 5, pp. 36–44, sept.-oct. 2012.
- [91] S. Mukhopadhyay and O. Postolache, *Pervasive and Mobile Sensing and Computing for Healthcare: Technological and Social Issues*. Springer Verlag, 2012, vol. 2.
- [92] Y.-L. Chen, H.-H. Chiang, C.-W. Yu, C.-Y. Chiang, C.-M. Liu, and J.-H. Wang, “An intelligent knowledge-based and customizable home care system framework with ubiquitous patient monitoring and alerting techniques,” *Sensors*, vol. 12, no. 8, pp. 11 154–11 186, 2012.
- [93] M. Skubic, “A ubiquitous sensing environment to detect functional changes in assisted living apartments: The tiger place experience,” *Alzheimer's & dementia : the journal of the Alzheimer's Association*, vol. 6, no. 4, pp. 1552–5260, 2010.
- [94] M. Fahim, I. Fatima, S. Lee, and Y.-K. Lee, “Daily life activity tracking application for smart homes using android smartphone,” in *Int'l Conf. on Advanced Communication Technology (ICACT)*, feb. 2012, pp. 241–245.
- [95] S. Hu, H. Wei, Y. Chen, and J. Tan, “A real-time cardiac arrhythmia classification system with wearable sensor networks,” *Sensors*, vol. 12, no. 9, pp. 12 844–12 869, 2012.

- [96] C. Liolios, C. Doukas, G. Fourlas, and I. Maglogiannis, “An overview of body sensor networks in enabling pervasive healthcare and assistive environments,” in *Int’l Conf. on Pervasive Technologies Related to Assistive Environments (PETRA)*. New York, NY, USA: ACM, 2010, pp. 43:1–43:10.
- [97] A. Rehman, M. Mustafa, I. Israr, and M. Yaqoob, “Survey of wearable sensors with comparative study of noise reduction ecg filters,” *Int. J. Com. Net. Tech*, vol. 1, no. 1, pp. 45–66, 2013.
- [98] O. Lara and M. Labrador, “Centinela: A human activity recognition system based on acceleration and vital sign data,” *Pervasive and Mobile Computing*, vol. 8, no. 5, pp. 717 – 729, 2012.
- [99] A. Pantelopoulos and N. Bourbakis, “A survey on wearable sensor-based sys. for health monitoring and prognosis,” *IEEE Trans. on Sys., Man, and Cybernetics*, vol. 40, no. 1, pp. 1 –12, jan. 2010.
- [100] C.-W. Lin, Y.-T. Yang, J.-S. Wang, and Y.-C. Yang, “A wearable sensor module with a neural-network-based activity classification algorithm for daily energy expenditure estimation,” *Information Technology in Biomedicine, IEEE Trans. on*, vol. 16, no. 5, pp. 991 –998, sept. 2012.
- [101] D.-W. Jang, B. Sun, S.-Y. Cho, S. Sohn, and K.-R. Han, “Implementation of ubiquitous health care system for active measure of emergencies,” in *Int’l Conf. on Advanced Language Processing and Web Information Technology (ALPIT)*, aug. 2007, pp. 420 –425.
- [102] R. Picard, E. Vyzas, and J. Healey, “Toward mach. emotional intelligence: analysis of affective physiological state,” *Pattern Analysis and Mach. Intelligence, IEEE Trans. on*, vol. 23, no. 10, pp. 1175 –1191, oct 2001.
- [103] E. Jovanov, A. O’Donnell Lords, D. Raskovic, P. Cox, R. Adhami, and F. Andrasik, “Stress monitoring using a distributed wireless intelligent sensor system,” *Eng. in Medicine and Biology Mag., IEEE*, vol. 22, no. 3, pp. 49 – 55, may-june 2003.
- [104] D. Craig, “Cognitive prosthetics in alzheimer’s disease: A trial of a novel cell phoned- based reminding system,” *Alzheimer’s & dementia : the journal of the Alzheimer’s Association*, vol. 6, no. 4, pp. 1552–5260, 2010.
- [105] S. Consolvo, D. McDonald, T. Toscos, M. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, *et al.*, “Activity sensing in the wild: a field trial of ubifit garden,” in *SIGCHI*. ACM, 2008, pp. 1797–1806.
- [106] A. Purohit, Z. Sun, F. Mokaya, and P. Zhang, “Sensorfly: Controlled-mobile sensing platform for indoor emergency response applications,” in *Int’l Conf. on Information Processing in Sensor Networks (IPSN)*, april 2011, pp. 223 –234.
- [107] C. Gomez-Otero, R. Martinez, and J. Caffarel, “Climapp: A novel approach of an intelligent hvac control system,” in *Iberian Conf. on Information Sys. and Technologies (CISTI)*, june 2012, pp. 1 –6.
- [108] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, “Peir: the personal environmental impact report, as a platform for participatory sensing sys. research,” in *MobiSys ’09*. ACM.
- [109] J. Froehlich, T. Dillahunt, P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. Landay, “Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits,” in *Int’l Conf. on Human factors in computing Sys.* ACM, 2009, pp. 1043–1052.

- [110] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, “Cartel: a distributed mobile sensor computing system,” in *Int’l Conf. on Embedded networked sensor Sys.* ACM, 2006, pp. 125–138.
- [111] R. Honicky, E. Brewer, E. Paulos, and R. White, “N-smarts: networked suite of mobile atmospheric real-time sensors,” in *ACM SIGCOMM workshop on Networked Sys. for developing regions.* ACM, 2008, pp. 25–30.
- [112] R. Murty, A. Gosain, M. Tierney, A. Brody, A. Fahad, J. Bers, and M. Welsh, “Citysense: A vision for an urban-scale wireless networking testbed,” in *IEEE Int’l Conf. on Technologies for Homeland Security*, 2008.
- [113] “Environmental sensor networks: A revolution in the earth system science?” *Earth-Science Reviews*, vol. 78, no. 34, pp. 177 – 191, 2006.
- [114] R. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. Abdelzaher, “Greengps: A participatory sensing fuel-efficient maps application,” in *Int’l Conf. on Mobile Sys., applications, and services.* ACM, 2010, pp. 151–164.
- [115] M. Azizyan, I. Constandache, and R. Roy Choudhury, “Surroundsense: mobile phone localization via ambience fingerprinting,” ser. *MobiCom ’09*, pp. 261–272.
- [116] N. Maisonneuve, M. Stevens, M. E. Niessen, P. Hanappe, and L. Steels, “Citizen noise pollution monitoring,” in *Int’l Conf. on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government.* Digital Government Society of North America, 2009, pp. 96–103.
- [117] D. Chu, N. D. Lane, T. T.-T. Lai, C. Pang, X. Meng, Q. Guo, F. Li, and F. Zhao, “Balancing energy, latency and accuracy for mobile sensor data classification,” ser. *SenSys ’11.* ACM, pp. 54–67.
- [118] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, “The weka data mining software: an update,” *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
- [119] D. Salber, A. K. Dey, and G. D. Abowd, “The context toolkit: aiding the development of context-enabled applications,” in *SIGCHI.* ACM, 1999, pp. 434–441.
- [120] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, “From wireless sensor networks towards cyber physical systems,” *Perv. and Mobile Comput.*, vol. 7, no. 4, pp. 397–413, 2011.
- [121] H. Bruce, G. Raffa, L. LeGrand, J. Huang, B. Keany, and R. Edgecombe, “An extensible sensor based inferencing framework for context aware applications,” *Int’l Conf. on Computer and Information Technology*, pp. 2878–2883, 2010.
- [122] B. S. Jensen, J. E. Larsen, K. Jensen, J. Larsen, and L. K. Hansen, “Estimating human predictability from mobile sensor data,” aug 2010.
- [123] N. Ravi, N. D, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in *Conf. on Innovative Applications of Artificial Intelligence (IAAI).* AAAI Press, 2005, pp. 1541–1546.
- [124] D. Figo, P. C. Diniz, D. R. Ferreira, and J. a. M. Cardoso, “Preprocessing techniques for context recognition from accelerometer data,” *Personal Ubiquitous Comput.*, vol. 14, no. 7, pp. 645–662, Oct. 2010.

- [125] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 48–53, jan.-march 2010.
- [126] Q. Shi, L. Wang, L. Cheng, and A. Smola, "Discriminative human action segmentation and recognition using semi-markov model," in *CVPR*, 2008.
- [127] S. Zhong and J. Ghosh, "Hmms and coupled hmms for multi-channel eeg classification," in *Int'l Joint Conf. on Neural Networks (IJCNN)*, vol. 2, 2002, pp. 1154–1159.
- [128] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Trans. Inf. Tech. in Biomed.*, vol. 10, no. 1, pp. 119–128, jan. 2006.
- [129] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-Efficient Continuous Activity Recognition on Mobile Phones: An Activity-Adaptive Approach," in *Int'l Symp. on Wearable Comput.*, 2012.
- [130] J. L. Bentley, D. F. Stanat, and E. W. Jr., "The complexity of finding fixed-radius near neighbors," *Information Processing Letters*, vol. 6, no. 6, pp. 209–212, 1977.
- [131] F. Lefvre, C. Montaci, and M.-J. Caraty, "On the influence of the delta coefficients in a hmm-based speech recognition system," in *Int'l Speech Sci. and Tech. Conf.* ISCA, Dec. 1998.
- [132] O. Thomas, P. Sunehag, G. Dror, S. Yun, S. Kim, M. Robards, A. Smola, D. Green, and P. Saunders, "Wearable sensor activity analysis using semi-markov models with a grammar," *Pervas. Mobile. Comput.*, vol. 6, no. 3, pp. 342–350, June 2010.
- [133] M. Li, V. Rozgic and, G. Thatte, S. Lee, A. Emken, M. Annavaram, U. Mitra, D. Spruijt-Metz, and S. Narayanan, "Multimodal physical activity recognition by fusing temporal and cepstral information," *IEEE Trans. on Neural Sys. and Rehabilitation Eng.*, vol. 18, pp. 369–380, aug. 2010.
- [134] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, Mar. 2011.
- [135] Y. Wang, B. Krishnamachari, Q. Zhao, and M. Annavaram, "Markov-optimal sensing policy for user state estimation in mobile devices," in *IPSN '10*. ACM, pp. 268–278.
- [136] G. Raffa, J. Lee, L. Nachman, and J. Song, "Don't slow me down: Bringing energy efficiency to continuous gesture recognition," in *Int'l Symposium on Wearable Computers (ISWC)*, oct 2010.
- [137] O. Thomas, P. Sunehag, G. Dror, S. Yun, S. Kim, M. Robards, A. Smola, D. Green, and P. Saunders, "Wearable sensor activity analysis using semi-markov models with a grammar," *Pervasive Mob. Comput.*, vol. 6, pp. 342–350, June 2010.
- [138] A. Krause and C. Guestrin, "Optimal nonmyopic value of information in graphical models: efficient algorithms and theoretical limits," 2005.
- [139] A. Krause, M. Ihmig, E. Rankin, D. Leong, S. Gupta, D. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta, "Trading off prediction accuracy and power consumption for context-aware wearable computing," in *IEEE Int'l Symp. Wear. Comput.*, 2005, pp. 20–26.
- [140] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *IEEE*, vol. 77, no. 2, pp. 257–286, feb 1989.
- [141] F. S. Schnatter, *Finite mixture and Markov switching models*. Springer Verlag, 2006.

- [142] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag New York, Inc., 2005.
- [143] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [144] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains,” *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.
- [145] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, July 2006.
- [146] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2000.
- [147] F. Ben Abdesslem, A. Phillips, and T. Henderson, “Less is more: energy-efficient mobile sensing with senseless,” in *MobiHeld '09*. ACM, 2009, pp. 61–62.
- [148] ADXL346 3-axis 2 G Ultra Low Power Digital Accelerometer by Analog Devices. [Online]. Available: <http://www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl346/products/product.html>
- [149] K. Lahiri, A. Raghunathan, S. Dey, and D. Panigrahi, “Battery-driven system design: A new frontier in low power design,” 2002.
- [150] T. F. Fuller, M. Doyle, and J. Newman, “Relaxation phenomena in lithium-ion-insertion cells,” *Journal of The Electrochemical Society*, vol. 141, no. 4, pp. 982–990, 1994.
- [151] M. Doyle, T. F. Fuller, and J. Newman, “Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell,” *Journal of The Electrochemical Society*, vol. 140, no. 6, pp. 1526–1533, 1993.
- [152] S. C. Hageman, “Simple pspice models let you simulate common battery types,” *Electronic Design News*, vol. 38, pp. 117–129, 1993.
- [153] P. Rong and M. Pedram, “An analytical model for predicting the remaining battery capacity of lithium-ion batteries,” in *Conf. on Design, Automation and Test in Europe*. IEEE Computer Society, 2003, pp. 11148–.
- [154] C. F. Chiasserini, R. R. Rao, and S. Member, “Energy efficient battery management,” in *Infocom '00*, 2001, pp. 396–403.
- [155] D. Panigrahi, S. Dey, R. Rao, K. Lahiri, C. Chiasserini, and A. Raghunathan, “Battery life estimation of mobile embedded sys.” in *Int'l Conf. on VLSI Design (VLSID)*. IEEE Computer Society, 2001, pp. 57–.
- [156] M. R. Jongerden and B. R. H. M. Haverkort, “Battery modeling, Technical Report,” January 2008.
- [157] Lithium-ion batteries. [Online]. Available: http://en.wikipedia.org/wiki/Lithium-ion_battery
- [158] M. Doyle and J. Newman, “Analysis of capacityrate data for lithium batteries using simplified models of the discharge process,” *Journal of Applied Electrochemistry*, vol. 27, pp. 846–856, 1997.
- [159] M. Computing, L. Martin, and T. L. Martin, “Balancing batteries, power, and performance: System issues in cpu speed-setting for mobile computing,” 1999.

- [160] J. F. Manwell and J. G. McGowan, "Lead acid battery storage model for hybrid energy sys." *Solar Energy*, vol. 50, no. 5, pp. 399 – 405, 1993.
- [161] L. Cloth, M. R. Jongerden, and B. R. Haverkort, "Computing battery lifetime distributions," in *Int'l Conf. on Dependable Sys. and Networks*. IEEE Computer Society, 2007, pp. 780–789.
- [162] V. Rao, G. Singhal, A. Kumar, and N. Navet, "Battery model for embedded sys." in *Int'l Conf. on VLSI Design*. IEEE Computer Society, 2005, pp. 105–110.
- [163] D. Danilov, R. A. Niessen, and P. H. Notten, "Modeling all-solid-state li-ion batteries," *Journal of The Electrochemical Society*, vol. 158, no. 3, pp. A215–A222, 2011.
- [164] I. Kaj and V. Konan, "Analytical and stochastic modelling of battery cell dynamics," in *Analytical and Stochastic Modeling Techniques and Applications*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7314, pp. 240–254.
- [165] N. Jaggi, S. Madakasira, S. R. Mereddy, and R. Pendse, "Adaptive algorithms for sensor activation in renewable energy based sensor sys." *Ad Hoc Networks*, 2011.
- [166] I. Constandache, S. Gaonkar, M. Sayler, R. Choudhury, and L. Cox, "Enloc: Energy-efficient localization for mobile phones," in *IEEE INFOCOM' 09*, April 2009, pp. 2716 –2720.
- [167] J. F. M. Bernal, L. Ardito, M. Morisio, and P. Falcarin, "Towards an efficient context-aware system: Problems and suggestions to reduce energy consumption in mobile devices," *Int'l Conf. on Mobile Business / Global Mobility Roundtable*, pp. 510–514, 2010.
- [168] J. Seo, S. Lee, and G. Lee, "An experience sampling system for context-aware mobile application development," in *Design, User Experience, and Usability. Theory, Methods, Tools and Practice*. Springer Berlin / Heidelberg, 2011, vol. 6769, pp. 648–657.
- [169] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, pp. 537–568, 2009.
- [170] C. Lin, N. Xiong, J. H. Park, and T.-h. Kim, "Dynamic power management in new architecture of wireless sensor networks," *Int. J. Commun. Syst.*, vol. 22, pp. 671–693, June 2009.
- [171] C. Alippi, G. Anastasi, M. D. Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors." *IEEE Trans. Instrumentation and Measurement*, pp. 335–344, 2010.
- [172] R. Pawula, "Generalizations and extensions of the fokker- planck-kolmogorov equations," *IEEE Trans. on Information Theory*, vol. 13, no. 1, pp. 33 –41, January 1967.
- [173] H. Risken and T. Frank, *The Fokker-Planck Equation*, 2nd ed. Springer, 1989.
- [174] H. Tijms, *A First Course in Stochastic Models*. Wiley, 2003.
- [175] A. Monteiro, G. V. Smirnov, and A. Lucas, "Non-parametric estimation for non-homogeneous semi-markov processes: an application to credit risk," 2006.
- [176] V. S. Barbu and N. Limnios, "Reliability of semi-markov sys. in discrete time: Modeling and estimation," in *Handbook of Performability Eng.*, K. B. Misra, Ed. Springer London, 2008, pp. 369–380.
- [177] R. Pyke, "Markov renewal processes: Definitions and preliminary properties," *The Annals of Mathematical Statistics*, vol. 32, no. 4, pp. 1231–1242, 1961.

- [178] G. D'amico, J. Janssen, and R. Manca, "Duration dependent semi-markov models," *Applied Mathematical Sciences*, vol. 5, no. 42, pp. 2097–2108, 2011.
- [179] S. E. Levinson, "Continuously variable duration hidden markov models for automatic speech recognition," *Comput. Speech Lang.*, vol. 1, no. 1, pp. 29–45, Mar. 1986.
- [180] S.-Z. Yu and H. Kobayashi, "A hidden semi-markov model with missing data and multiple observation sequences for mobility tracking," *Signal Processing*, vol. 83, no. 2, pp. 235 – 250, 2003.
- [181] G. Ciuperca and V. Girardin, "Estimation of the entropy rate of a countable Markov chain," *Communications in Statistics-Theory and Methods*, vol. 36, no. 13-16, pp. 2543–2557, 2007.
- [182] S. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, 2nd ed. New York, NY, USA: Cambridge University Press, 2009.
- [183] E. Altman, *Constrained Markov Decision Processes: Stochastic Modeling*, ser. Stochastic Modeling Series. Taylor & Francis Group, 1999.
- [184] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. John Wiley & Sons, Inc., 1994.

APPENDICES

Appendix A : Some Derivations for CMDP

Where γ and u are defined any initial distribution and any stationary policy, $r, w \in S^R$, $a \in A$, the occupation measure is derived from

$$\begin{aligned}
 \rho(w) &= \gamma(r) + \sum_r \sum_{a(r)} \rho(w, a) P_{rw}^a \\
 &= \gamma(w) + \sum_r \rho(r) \sum_{a(r)} \frac{\rho(w, a)}{\rho(w)} P_{rw}^a \\
 &= \gamma(w) + \sum_r \rho(r) \sum_{a(r)} u_w(a) P_{rw}^a \\
 &= \gamma(w) + \sum_r \rho(r) P_{rw}(u)
 \end{aligned} \tag{A.1}$$

yields to have $\rho = \gamma(I - P_{rw}(u))^{-1}$.

The expected cost is expressed as in

$$\begin{aligned}
 C(\gamma, u) &= E_\gamma^u \left\{ \sum_{t=1}^{\infty} c(S_t^R = r, A_t = a) \right\} \\
 &= \sum_{t=1}^{\infty} E_\gamma^u c(S_t^R = r, A_t = a) \\
 &= \sum_{t=1}^{\infty} \sum_r \sum_{a_i} Pr(S_t^R = r, A_t = a) c(r, a) \\
 &= \sum_r \sum_{a_r} \sum_{t=1}^{\infty} Pr(S_t^R = r, A_t = a) c(r, a) \\
 &= \sum_r \sum_{a_r} f(\gamma, u, r, a) c(r, a)
 \end{aligned} \tag{A.2}$$

In the similar way, for the constraints,

$$D^y(\gamma, u) = \sum_r \sum_{a_r} f(\gamma, u, r, a) d^y(r, a) \tag{A.3}$$

Appendix B : Copyright Permission for Chapter 4

7/5/13

Rightslink® by Copyright Clearance Center



RightsLink®

Home

Account Info

Help



Title: Adaptive and Energy Efficient Context Representation Framework in Mobile Sensing
Author: Yurur, O.; Labrador, M.; Moreno, W.
Publication: Mobile Computing, IEEE Transactions on
Publisher: IEEE
Date: 0
Copyright © 1969, IEEE

Logged in as:
Ozgur Yurur

LOGOUT

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Copyright © 2013 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement.](#) Comments? We would like to hear from you. E-mail us at customercare@copyright.com

<https://s100.copyright.com/AppDispatchServlet>

1/1

ABOUT THE AUTHOR

Özgür Yürür received double major degrees from the department of electronics engineering and the department of computer engineering at Gebze Institute of Technology, Kocaeli, Turkey in 2008, and M.S.E.E. degree from the department of electrical engineering at University of South Florida (USF), Tampa, FL, USA in 2010. He is currently pursuing the Ph.D. degree in Electrical Engineering at USF. Mr. Yurur conducts his research in the field of mobile sensing. His research area covers ubiquitous sensing, mobile computing, machine learning, and energy efficient optimal sensing policies in wireless networks. The main focus of his research is on developing and implementing accurate, energy efficient, predictive, robust, and optimal context-aware algorithms and framework design on sensor enabled mobile devices.