Scalable Recognition of Human Activities for Pervasive Applications in Natural Environments

by

Selene Atenea Mota Toledo

S.M., Massachusetts Institute of Technology (2002)
B.S., Electronic Systems Engineering, ITESM, Cuernavaca, Mexico (1999)

Submitted to the Department of Architecture
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Architecture: Design and Computation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2014 [September 2014]

Signature redacted

Author ....
Department of Architecture
August 8, 2014

Signature redacted

Certified by.......................
Kent Larson
Principal Research Scientist
Departments of Architecture and Media Arts and Sciences
Thesis Supervisor

Signature redacted

Accepted by..........................
Takehiko Nagakura
Chairman, Department of Architecture Graduate Committee

Scalable Recognition of Human Activities for Pervasive Applications in Natural Environments

by

Selene Atenea Mota Toledo

The following people served as readers for this thesis:

Signature redacted

Thesis Supervisor ... .

Kent Larson

Principal Research Scientist

Departments of Architecture and Media Arts and Sciences

Thesis Reader ....

George Stiny

Professor of Design and Computation

Department of Architecture

Thesis Reader ....

Dr. Richard Fletcher

Assistant Professor of Psychiatry

University of Massachusetts Medical School

Scalable Recognition of Human Activities for Pervasive Applications in Natural Environments

by

Selene Atenea Mota Toledo

Submitted to the Department of Architecture
on August 8, 2014 , in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Architecture: Design and Computation

Abstract

Past approaches on the automatic recognition of human activities have achieved promising results by sensing patterns of physical motion via wireless accelerometers worn on the body and classifying them using supervised or semi-supervised machine learning algorithms. Despite their relative success, once moving beyond demonstrators, these approaches are limited by several problems. For instance, they dont adapt to changes caused by addition of new activities or variations in the environment; they dont accommodate the high variability produced by the disparity in how activities are performed across users; and they dont scale up to a large number of users or activities. The solution to these fundamental problems is critical for systems intended to be used in natural settings, particularly, for those that require long-term deployment at a large-scale.

This thesis addresses these problems by proposing an activity recognition framework that uses an incremental learning paradigm. The proposed framework allows learning new activities - or more examples of existing activities - in an incremental manner without requiring the entire model to be retrained. It effectively handles within-user variations and is able to reuse knowledge among activities and users. Specifically, accelerometer signals -generated by 3-axis wireless accelerometers worn on the body- are recognized using a machine-learning algorithm based on Support Vector Machine classifiers coupled with a majority of voting algorithm.

The algorithm was implemented and evaluated using datasets collected at experimental, semi-naturalistic, and naturalistic settings. Hence, compared to other state-of-the-art approaches, such as Hidden Markov Models or Decision Trees, the system significantly improves the between-class and between-subject recognition performance and requires significantly less data to achieve more than 90% within-class overall recognition rate. Based on this approach, a functional system was designed and implemented across a variety of application scenarios (from a social-exergame for children to a long-term data collection of physical activities in free-living settings). Lessons learned from these practical implementations are summarized and discussed.

Thesis Supervisor: Kent Larson
Title: Principal Research Scientist
Departments of Architecture and Media Arts and Sciences

Acknowledgments

Thanks to my advisor Kent Larson. Kent, thanks for supporting me all these years, for your wise guidance in important moments of this journey, for letting me explore and pursuing research ideas that I am passionate about, for letting me grow in the scientific community, for your kind words and valuable advise, and for teaching me the art of communicating complex ideas in a simple yet effective manner. Thanks for making the Changing Places Group an awesome place to work and be creative.

Thanks to my reader and thesis committee Dr. Richard Fletcher who has provided me with extremely valuable advice, constructive feedback, and encouraged me to be part of exciting research communities. Rich, you are not only exceptional researcher but also an incredible person. Thanks for reading this document.

Thanks to Prof. George Stiny. George, thank you so much for your invaluable support throughout this process. Specially, thanks for teaching me the importance of why I should never give up. As a *Design and Computation* student, I had learned to be independent, pursue out dreams with determination, be truly creative, and grow all aspects of my intellect and persona.

Special thanks to Prof. Rosalind Picard who has been over many years a mentor and inspiration. Roz, you share so much wisdom and acknowledge with everyone. The motivation of measuring behavior to respond and understand human emotion has, without doubt, inspired my research and the research of many others over the years -inclusive at the time that Affective Computing was an obscure topic. Many of the application ideas that I have had over the years are inspired by your lectures. As you always said, the challenge is not how humans adapt to the technoly, but how the technoly adapts, improves, and enhances the very best in each of us.

I also want to thank Prof. Tommy Jaakkola, Prof. Jacob Crandall, and Prof. David Sontag (my TA in the machine learning class, now professor at NYU) who, since a very early stage of my PhD, encouraged me to pursue in class what has becomed the machine learning and game theory methods that I am using in this thesis.

Renee Caso and Cynthia Wilkes, thanks for your help and guidance during my studies at MIT. I have many things to thank you for –specially at the end of this process- and little space. Thanks you for all the amazing work you do at the department. You make MIT not only be a place to study, but also a place to be connected with and a community. A community that, for many of us, feels like home.

Thank you Emmanuel Munguia Tapia for proding me with guidance, inspiration and advice in my research. Your thesis work was a great source of inspiration and a solid starting point that made possible for me to speed up my learning on activity recognition and find early on the right path for pursuing effective solutions to the problem of large-scale activity classification. I probably have no enough words to prize the high quality of your PhD work. Many thanks for sharing all your finding and insights.

Thanks to Fahd Albinali for making the working hours fun. Fahd, I always enjoyed very much the late working hours of coding and experimenting. Thanks for teaching me the importance of laughing and having good humor when working intensely.

Thanks you to all the past and current members of Changing Places Group (before HouseN). To the past members: Shaun, Preveen, Jonathan, Ned, Bruno, Anh, Jennifer B, Manu, Juno, Fahd, and Topper. And the new members: Caleb, Mo, Ira, Ryan, Daniel, Wolfgang, Hasier, Michael, Yuki, Carlos, Layla, Carson, Ramiro, Sotirios, Jarmo, Aithne. To my awesome officemates over the years: Sandra, Jason, Tyron, Harrison, Pilar, Yi, and Clay. You always made the office to feel a friendly welcoming place in which I was happy to work and see you everyday. To some of you, thanks for tolerating me sleeping/living at the office days when there were project deadlines. You guys are amazing and I enjoyed everyday to have you around. I felt surrounded of lots of creativity and insightful views.

Thanks to the group UROPS, Cindy and Alex, who worked very hard in various aspects of the wockets project in an early stage and were always fun and collaborative. Cindy and Alex, along with Yi, Jason and Fahd, thought me how fun is to play ping-pong at the lab.

Thanks to all the past and current members of the affective computing group. Special thanks to Akane, Javier, Daniel, Rana and Elliot. You guys are like family to me. I always felt so welcomed whenever I worked in your area, discussed research ideas with you, help each other in our research whatever we could. I think you exceptional people and it has been a joy to spend time with you.

Thanks to my colleagues and friends at Media Lab: Gordon, Nan, Nanwei, Gershon, Sandra, Karthik, John, NovySan, Edwina, Valentine, and Lining Yao. I just love your originality, happiness, and creativity. You are the most amazing creative people I have met. Also, thanks to my long-time friends: Cati, Ash, Stefan, Raul, and Mic, Who have always been there whenever I have needed it, independently of where in the world I am.

I want to thank the Wilkings-Greenwald, the M4s and the Pingitore families for opening the doors to their loving homes and make me feel extremely welcome every time I visit them. Their loving and playful environment always made me feel welcomed, happy and energized.

I want to thank to my friends in Mexico who are like family to me, Karina and Adriana, who through out the years have been the best and most loving friends that someone could have. To Robert that, despite all the time and distance, has been always present whenever I have needed it. To my family, papa Fer, Anita and Xochilt Mota, and Ana Gomez. Your sweetness, kindness and creativity can light up the cloudiest day.

Thanks to my dear friends: Carla, Akane, Aithne, Heidi, Bernhard, and Cami. Thanks for your friendship, joy and happiness. Thanks for sharing innumerable fun moments. Also, thanks to my housemates: Rana, Nadia, Max, Katya, Andra, and Dave, who always had a warm and friendly smile for me whenever I came back home.

Specially, I want to thank from the bottom of my heart my sister Ari and my friends Iliana, Jason, Pilar and Jean. Thanks for helping me in all fronts during this process and gave me strength and an infinity amount of emotional support. Everyday, I feel so blessed to have you in my life.

Finally, I want to thank Scott Greenwald for giving me an incomparable and endless amount of love and support in every single step of this process. Thanks so much for making me feel that I was not alone from the beginning to the end.

Scott:

You made this thesis possible, thank you very much from the bottom of my heart. Thank you a lot for believing in me, even when I did not. I dedicate this thesis to you with love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The notion of systems adapting to a users activity is an essential part of many context aware systems, ambient intelligent environments and personal-health applications [3, 45, 132]. The advances on miniaturized sensing, wireless communication technologies and smartphones have started to make possible to collect daily fine-grained activity data over a large number of individuals [109, 31, 35]. As a result, human activity recognition research has gone from low-level detection of actions (body locomotion like walking) to high-level understanding of behavior in natural environments (complex actions like washing dishes). Figure 1-1 shows some examples.

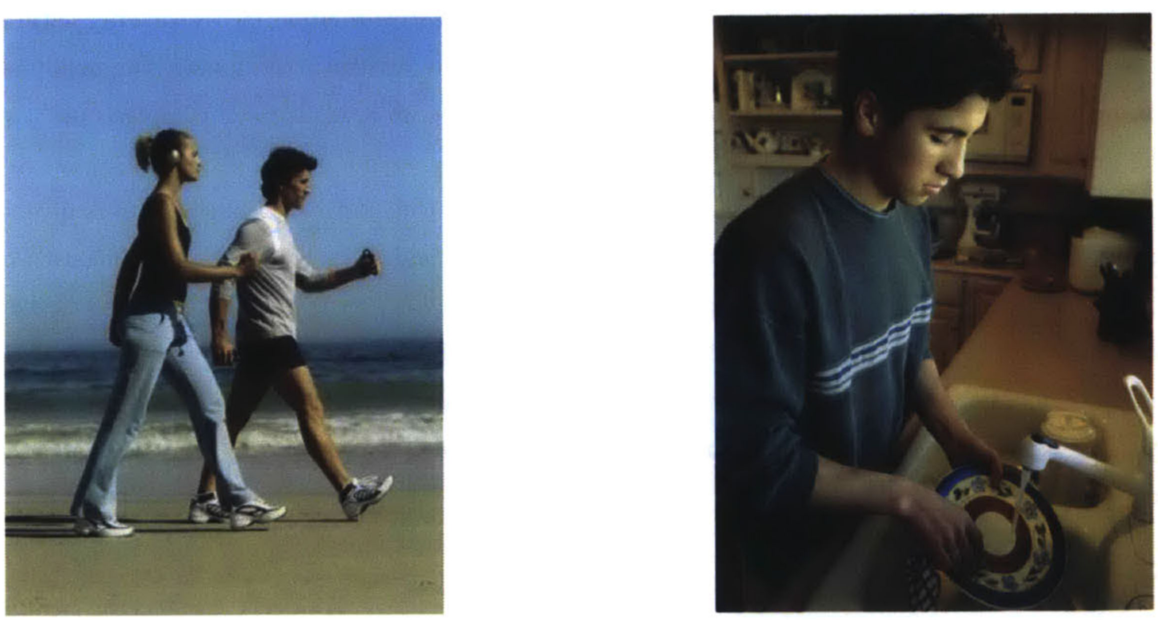


Figure 1-1: Activity recognition can range from body locomotion to complex actions. Left: Body locomotion. right: Complex actions.

These advances and the availability of vast amounts of data have open up new possibilities for developing innovative applications that explore novel forms of interaction and behavior understanding. In particular, they have made possible to start exploring activity recognition systems that can be deployed in the real-world over long-periods of time (such as in [116, 184], see examples of these activities in figure 1-2).

Figure 1-2: Examples of activites happening in the natural settings.

Currently several applications use environmental sensors and smart phones to provide sensing capabilities that take advantage of simple context information such as users location, coarse-grained body motion or simple environmental indicators (e.g., light or noise levels). However, after years of research, results have indicated that coarse-grained body motion and simple location are not enough to recognize a broad range of complex activities and situations that are intrinsic to natural settings. As a consequence, the recognition of natural occurring activities is not a trivial problem. Of course, this is not surprising given that humans are very complex creatures and such complexity increases when their behavior is studied in naturalistic and unconstrained settings.

In this space, wearable activity recognition offers an attractive solution for recognizing natural occurring activities in free-living settings. It allows systems to classify activities starting at a fine-grained level of abstraction and, then, uses them for building up towards more complex ones (see figure 1-3).

This solution has the advantage of being able to capture a wide-range of behaviors across diverse settings, while requiring little infrastructure. It also better addresses sensitive issues in terms of privacy, ethics, and obtrusiveness compared to conventional activity recognition approaches based on computer vision [138, 115]. However, the caveat is that wearable activity recognition has challenging requirements in terms of sensing hardware, algorithm, and user interaction. As can be seen next, such requirements are not only challenging, but also deeply interconnected with each other and the entire design of the system.

Sensing Hardware:

Most wearable sensors need to run continuously and be operated and programmed wirelessly. This imposes many hardware design challenges, in particular, for systems that aim to be used in real-

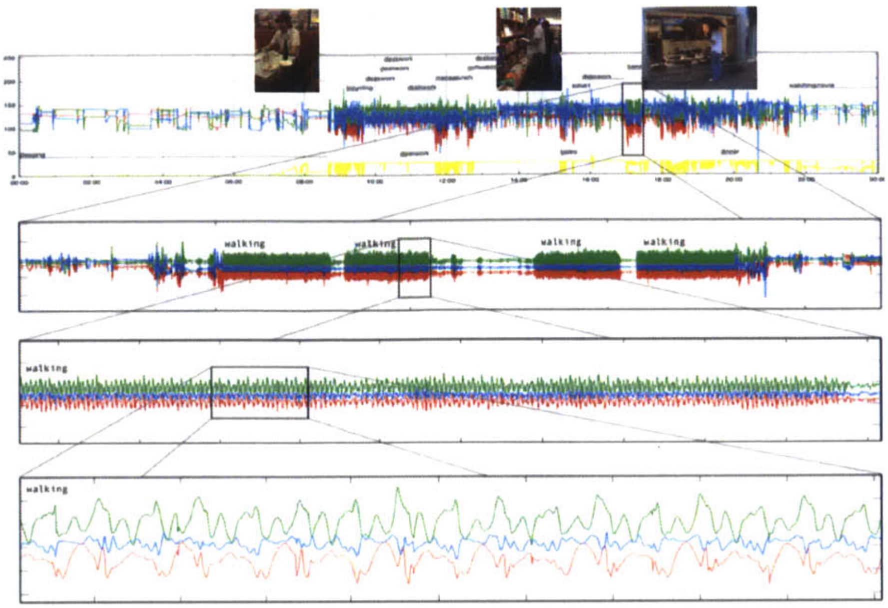


Figure 1-3: Example of wearable activity recognition systems can classify activities starting at a fine-grained level of abstraction and build up towards complex ones.

world settings. Practical issues include the acceptability and viability to wear the sensors on the body (specially for applications that require 24/7 behavior monitoring). Other issues include that the system needs to be small size, be easy to use, have long battery life and, in general, satisfy the needs of the real-world scenarios (see section 2.5.1 for a detailed discussion about these needs).

Among the wide-variety of sensing options, miniaturized low-power accelerometers are the most widespread type of sensor used for wearable activity recognition. Their popularity is based on the fact that they are very effective at capturing actions involving repetitive motion or still positions like physical activities and postures (walking, running, standing, etc.), while they are low-cost, require minimal instrumentation, and are easy to deploy. As a consequence, when combined with mobile phone technologies, they currently offer the most suitable and scalable sensing solution for capturing natural occurring fine-grained behavior during extended periods of time. Figure 1-4 shows examples of common wearable devices embedding miniaturized accelerometers.

Algorithm:

Besides the sensing hardware component, wearable activity recognition requires an algorithm that recognizes the patterns of movement coming from the sensing devices. Current approaches have had promising results by sensing patterns of physical motion using 3-axis wireless accelerometers worn on the body and classifying them using supervised or semi-supervised machine learning algorithms (like decision trees or hidden markov models). However, when deployed in natural settings, these approaches are limited by numerous practical problems such as: high-degree of user and situational

Figure 1-4: examples of common wearable devices embedding miniaturized accelerometers.

dependence, need of having large amounts human labeled data -which is expensive to obtain when the system is used continuously (24/7) in natural settings-, lack of capability to accommodate the high-variability caused by the disparity in users behavior and noise coming from sensing devices.

As a matter of fact, the majority of existing machine learning methods cannot be directly used in real-world applications because they are often carefully handcrafted to very specific conditions in terms of activities, subjects and context. Indeed, most activity recognition systems are proof-of-concept systems which are mainly completed by carefully hardwiring the discontinuous prototype technologies which makes them dependent to ecological arrangement, sensor types and installation, as well as, specific activities and users. Existing state-of-the-art activity recognition systems suffer a great deal from lack of scalability and ability to exchange information between different parts of the system. These issues make it difficult to integrate them into end-user and real-world applications, reducing the possibility to be accessible to non-experts who want to use activity recognition as a tool to gain insights about fine-grained behavioral information.

On the other hand, as the scope of the activity recognition system broadens from carefully controlled experiments to large-scale user-generated naturalistic data, existing approaches deal poorly with user diversity in terms of age, behavioral patterns, daily routines, lifestyle, etc. (figure 1-5 shows examples of such user diversity). Performance degradation due to the difference between people can seriously affect the classication accuracy of activity recognition systems. Actually, according to Lane et. al. [98], this can be the case even when the system consists of only one activity and, as few as, fifty users. Although approaches based on user-customized models (such as [108, 165]) gen-

Figure 1-5: Activities happening in natural settings have high behavioral diversity.

erally deal better with the user diversity problem, this occurs at the cost of increased dependence on human annotation or feedback. Thus, to obtain high recognition accuracy requires providing large amounts of carefully labeled data-segments (containing start and end activity makers) for each user. This dependence brings to the table another set of issues related to ground truth annotation and the user-system interaction.

User Interaction:

As described in the previous section, two major problems among current activity recognition systems are: (1) high-dependency on user-specific training, and (2) the need for large amounts of carefully labeled data. Indeed, these problems are significantly more challenging when activities are collected in naturalistic settings than in controlled experimental settings.

Typically activity recognition experiments in controlled settings involve few subjects (often inclusively, the researchers who develop the system) and sets of scripted activities. In general, these conditions tend to oversimplify the problem and, similarly, the activities can unconsciously be performed in a way that favors the recognition system. Figure 1-6(A) shows an example of a typical experimental setting.

On the other hand, longitudinal capturing of activities in naturalistic settings enables more realistic collection of data. However, its drawback is that collecting ground-truth annotations for this purpose is not an easy task. For instance, previous research on activity recognition makes the assumption that labels are consistent. While this might be true in simple experimental settings, in naturalistic settings this might not be the case and annotation issues might start becoming a concern (e.g., it might cause annotator annoyance or interruption, assignment of different labels to similar activities, overlap of activity labels boundaries, etc.)[137]. Figure 1-6(B) shows an example of a typical setup in a natural setting. There are several methods to capture longitudinal annotations in naturalistic settings (see section 2.5.3 for a detailed discussion). But, most methods are error-prone and time-consuming, they might use invasive sensors that are not acceptable due to privacy reasons,

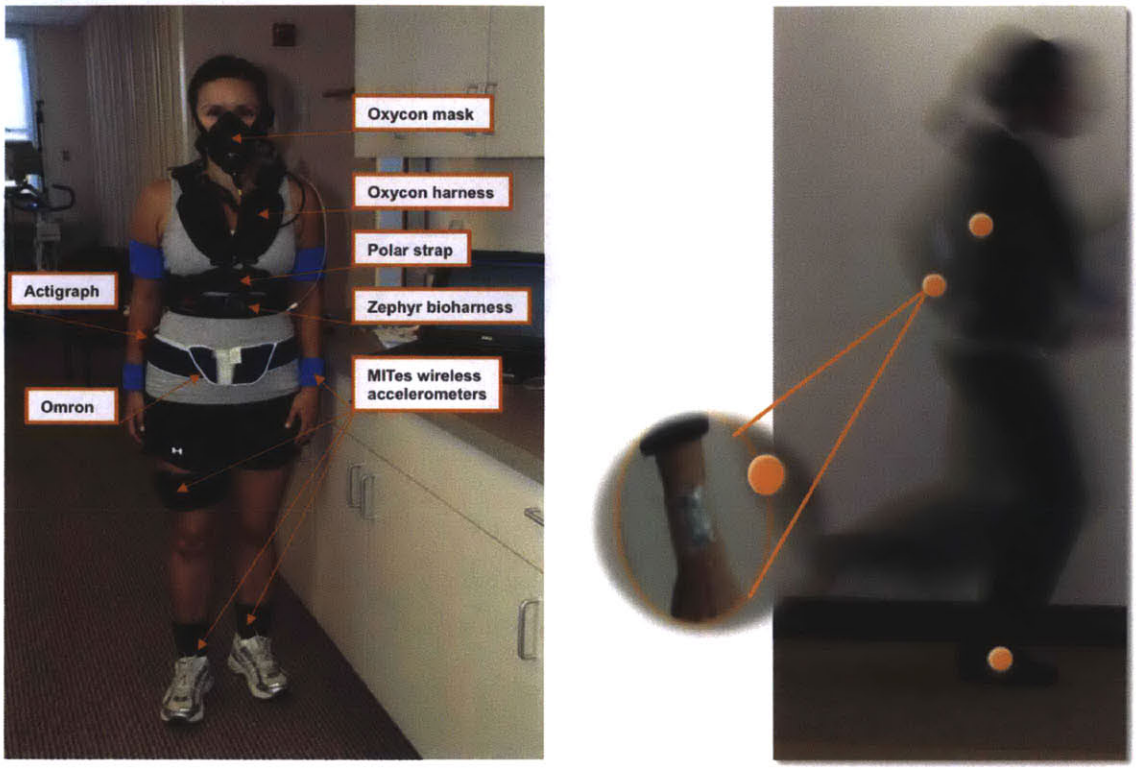


Figure 1-6: Example of typical experimental and naturalistic setups for activity recognition. (A) The left figure shows the set up used for the data collection reported in Tapia et al. [169]. (B) The right figure shows the set up used for activity recognition in natural settings using wockets, which is the sensing system used in this work.

or they might disrupt or annoy the user - who typically is whom annotates the data. And this might be especially problematic when detailed user annotations are needed.

1.2 Scope of Research

Recently, several models have been proposed to alleviate the issues and limitations of current activity recognition systems. For example, the learning-upon-use paradigm (described in Lukowicz et. al. [109]) proposes a practical approach in which simple action detectors can be trained reasonably well in a user-depended way. This means that an activity recognition system running on a smart phone application can be pre-trained and delivered ready to use. Precisely the idea is that, while the user might have given an application that initially doesnt fully work, the activity recognition system could learn as the user interacts with the system. Of course, depending on the training and feedback/annotation method, this can be a complex and tedious process. In contrast to the learning-upon-use paradigm, a number of studies [97, 98] had suggested models that require minimal user intervention by crowdsourcing the labeling of training examples and updating the user activity classification model in a networked fashion.

Even though the crowdsourcing paradigm is relative novel in the field of wearable activity recognition, it is not within the field of computer vision - in where its advantages and disadvantages are well known [175]. Such knowledge confirms the findings indicating that existing activity recognition methods based on crowdsourcing dont scale well with an increasing number of users. Hence for large groups of users (particularly among cases in where the behavioral differences get more prominent and, for instance, harder to generalize), the crowdsourcing annotation becomes an impractical, redundant and non-scalable task. Besides, data recorded from wearable sensors (like accelerometers or gyroscopes) is generally not intuitive and more difficult to interpret than data produced via other sensors such as RFID tags, environmental sensors, microphones, or cameras [189]. Indeed, annotating activities based on accelerometer data generally requires an experienced annotator (generally the researcher) to identify the activity from the data stream. In addition, crowdsourcing methods could be costly, since the data is labeled in a traditional fashion without end-user involvement via a third observer -the crowdsourcing online worker- who has to be trained to label complex data.

These two paradigms represent two opposite sides of the spectrum that characterizes the user-system interaction. For instance, one side, the learning-upon-use paradigm is low-cost and scalable. However, the intense labeling of examples could cause user-burden because of unwanted interruptions and information overload. On the other side, the crowdsourcing paradigm has the advantage of keeping the burden for the end-user low, but it has to deal with errors generated by annotators/online-workers. The number of errors is generally high due to the difficulty to interpret and generalize behavioral data. In particular, if the data is based on accelerometer signals, crowd-sourced annotations often suffer from low-quality and inconsistency [113, 129]. Moreover, the crowdsourcing approach is subjected to additional privacy concerns since it explicitly exposes users behavioral data to third parties (who are not part of the research staff or the system creation team and who, often, have few or not background checks). As well, this approach makes it difficult to tag personal preferences or information that can be valuable for the user.

Alternatively to these paradigms, this thesis focuses in a solution lying at the middle of the spectrum. It takes advantage of the learning-upon-use paradigm -by incrementally acquiring knowledge from examples provided by the users- whereas minimizes the number of examples -by maximizing the information provided to the classifier taking advantage of data similarities across activities and users (see chapter 3 for details).

Specifically, this thesis argues that the performance of current activity recognition systems is highly conditional to the systems capability to learn efficiently with few examples and quickly adapt to new activities and situations. Thus, this work focuses on the problems of adaptability and scalability faced by wearable activity recognition systems deployed in free-living settings. It systematically analyzes limitations of state-of-the-art approaches to identify current challenges and critical trade-offs encountered in real-time settings. Further, it proposes a functional system and an algorithm framework designed with basis on lessons learned from their functional implementation and their use in real-world applications (see chapter 7.2 for more details). Indeed, it is argued that wearable activity recognition for the real-world needs to satisfy the following requirements:

1. Learn incrementally and upon-use the specifics of the user and the environment using as few instances as possible.

2. Handle and adapt to the high variability produced by changes in users behavior, environmental conditions, and systems operational difficulties.

3. Adapt the system interaction with the user to maximize user-adoption while minimizing users cognitive load, interruptions and, in general, negativity bias. In other words, design the system and algorithm by having the user-in-the-loop.

4. Be scalable and computationally tractable in order to be used with other systems at different or higher-levels of abstraction. This can be achieved by creating robust activity primitives that can be used and reused - exploiting common acknowledge - within the context of a hierarchical activity recognition framework.

In summary, these fundamental functionalities are the corner stones for moving traditional approaches towards the next-generation of wearable activity recognition systems. We envision that they will facilitate the development of scalable, easy-to-use, and easy-to-deploy systems, which can be used effectively in real-world applications and with large-scale networked subsystems that interact with communities of users.

1.3 Proposed Approach

Building on the idea of incremental and transfer learning, this thesis provides a framework and presents a functional system that addresses the scalability and adaptability limitations of current approaches. In addition, due to its computational efficiency, the model is capable of running in real-time on a common mobile phone.

Specifically, accelerometer signals are generated using low-power wireless accelerometers called Wockets, and activities are recognized using a machine-learning algorithm based on a hierarchy of Support Vector Machine (SVMs) classifiers coupled with a voting mechanism. Specifically, the system uses a SVM to represent a particular activity. Each SVM is treated as a building block that fits into an output classification layer. This output layer tracks over time the SVM blocks classification results and computes the predicted activity using a weighted majority of voting algorithm. Figure 1-7 shows the wockets system.

The design guidelines of the entire system were formulated based on the end-to-end implementation, testing and user-centered validation of the hardware, software (signal processing and algorithm)

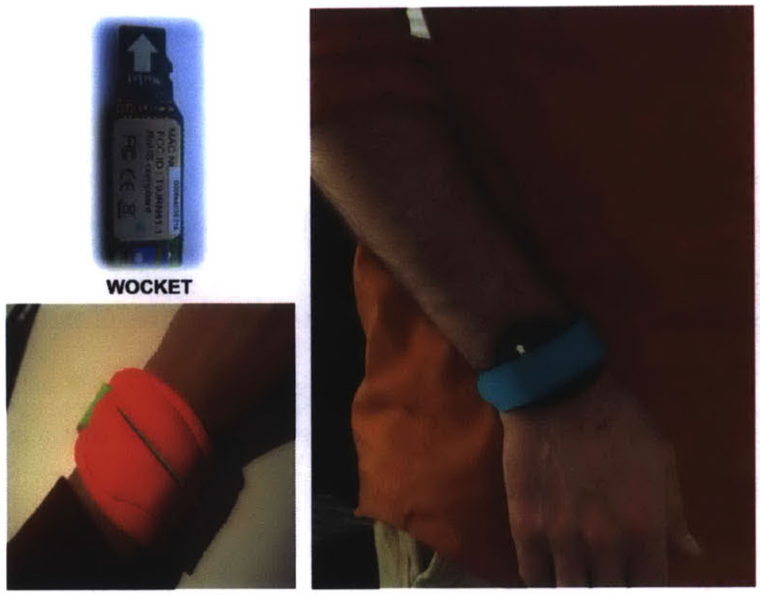


Figure 1-7: Example of an user wearing a miniaturized low-power accelerometer (wocket).

components. Particularly, inspired by the "learn-to-learn" paradigm, the framework presented in this thesis ultimately informs the design of scalable real-world wearable systems that allow end-users (jointly with researchers, doctors, designers, etc.) to work together in a feedback loop where activity data is collected and analyzed as a continuous design cycle and new activities are added, and in turn new data collected. Thus, researchers could monitor the activity data in real time, spot problems on the fly and, change what the system recognizes at anytime according to their needs - instead of waiting to analyze the results at the end of the study.

1.4 Contributions

This thesis proposes a novel scheme for activity recognition based on wearable accelerometers to handle differences across users, settings and sensing devices in a scalable and adaptable way. It provides a framework that facilitates the incorporation of user-feedback into the system by building learning-upon-use applications aimed to be deployed at large scale in naturalistic settings.

To bring down the cost of computing, we maintain groups of similar activities by modeling activities as a set of basic primitives and treating the user-behavior, settings and sensor differences as noise. In addition to the features extracted directly from the acceleration signals, we also use the similarity measures of features to train the activity statistical model. Specifically, to achieve comparable accuracy while keeping the user-burden low, we exploit the knowledge provided within groups of users. This means that an activity label provided by a particular user could be used to recognize the same activity for another user within the same group (using the activity similarity metrics).

To enable robust labeling, even in the case of unreliably labeled data from users, we handle the common issue of the overlapping boundary of activities (see Peebles et. al.[137] for more details about this type of problem). To handle overlapping class boundary resulting from inaccurate start and ending times, we use multi-instance learning [101, 185]. Although important, other issues like semantic discrepancy of labels and other errors caused by noisy data are not covered in detail given that they are beyond the scope of this work.

In sum, the framework presented in this thesis contributes to state-of-the-art wearable activity recognition systems as follows:

1. It handles behavioral variability within and between users in a way that remains practical even if the number of users increases.
2. It facilitates to realize the learning-upon-use paradigm because it updates the statistical model by incorporating new activities or more examples of existing activities in an incremental manner without requiring the whole system to be retrained.
3. It allows the possibility to transfer statistical model knowledge between users a model that is well trained for one user can be transferred to another user.
4. It allows creating opportunities for incorporating user feedback by collecting and analyzing data as a continuous design cycle.

1.5 Synopsis

The chapters in this thesis are organized as follows:

Chapter 1: Introduction. This chapter describes the research problem and motivation.

Chapter 2: Background. This chapter provides an analysis of the challenges faced by wearable activity recognition systems and highlights key results to illustrate: what is feasible, what are the current difficulties, and where is the potential for facilitating long-term activity recognition at a large-scale. Subsequently, we use this analysis to position the contributions presented in this thesis.

Chapter 3: 3. This chapter describes the proposed algorithm framework and presents the criteria used to select the algorithm parameters and design its properties.

Chapter 4: Datasets. This chapter describes the data used to carry out the algorithm experiments.

Chapter 5: Materials and Methods. This chapter presents the measures used to assess the algorithms performance in the experiments.

Chapter 6: Experiments. This chapter presents the evaluation, the discussion of results, and the final version of the algorithm framework in detail. The evaluation is mainly focused on assessing the algorithm performance in terms of learning and generalization.

Chapter 7: Design Guidelines for Activity Recognition Systems. This chapter describes the design guidelines of the hardware and the software platform used to collect the data for the exploratory analysis, development, and evaluation of the wockets system.

Chapter 8: Conclusion. This chapter summarizes the contributions and concludes the thesis by proposing extensions and future work.

Chapter 2

Background

2.1 Overview

Over the last decade, the rapidly expanding field of personal devices, miniaturized wireless-sensing technologies and machine-learning algorithms have made significant progress in terms of size, cost, power-efficiency and computational-processing capability [132, 35, 45]. These advances have facilitated the development of novel wearable systems that can detect when, how and how often a user performs specific activities. This achievement has caused activity recognition research to move towards a more ambitious arena of investigating human behavior in naturalistic settings during extended periods of time.

This chapter aims to contextualize the work and contributions presented in this thesis by examining state-of-the-art approaches in wearable human activity recognition and their challenges. In particular, it analyzes the evolution of these challenges and their implications for systems intended to be deployed in the real-world. It discusses crucial design and implementation issues for activity recognition systems and examines them from a practical perspective in terms of sensing methods, machine learning algorithms, data annotation techniques, and system-user interaction.

Further, it aims to illustrate that wearable human activity recognition in natural settings is not a trivial problem. Indeed, it is problem that requires an inter-disciplinary approach and well-informed understanding of the design trade-offs and challenges caused by the complexities of recognizing human behavior in natural settings. It shows that many of these challenges are caused by the fact that the requirements of wearable activity recognition systems are not insignificant. For instance, the system needs to be highly power-efficient, low-cost, robust, scalable, fast when deployed in embedded devices, and able to run over long periods of time in unpredictable free-living settings. Besides, there are other challenging requirements related with system usability (e.g., compliance, comfort, and aesthetics) and user-feedback (e.g., inform the user about the system status or prompt her/him for knowledge acquisition for the recognition task).

In sum, this chapter provides a comprehensive analysis of the challenges faced by state-of-the-art activity recognition systems and highlights key research findings to illustrate: what is feasible, what are the current difficulties, and where is the potential for facilitating long-term activity recognition at a large-scale.

2.2 Activity Recognition: More than Overall Movement

By definition, activity recognition aims to recognize actions and goals performed by one or more agents from a series of observations. Such observations can be the sequence of actions performed by the agent or the environmental conditions surrounding such actions. Generally speaking, agents can be any entity that can perform actions (e.g. robots, algorithms performing digital transactions, humans, etc.). In our case, we consider that humans are the agents and human actions and/or its surrounding environmental conditions are the observations.

In practice, both human activity and environmental conditions combined are often used to infer complex activities performed in a wide-range of application scenarios. However, human activity is a very broad term that can involve activity ranging from physical motion (e.g., body movement, gestures, or compound activities) to physiological signals (e.g., heart-rate, electro-dermal activity, body temperature, muscle activity, etc.). Whereas, environmental conditions can involve contextual indicators, specific information or objects that could be used as indicators of an activity. For example, location could be used to infer the activity of cooking when being at the kitchen, whereas, location combined with daylight and time of the day could help to infer sleeping when being in the bedroom at night and there is absence of light.

In recent years, activity recognition research has received vast attention given its applicability to support a wide-range of real-world applications that have a high impact in peoples lives. Such applications range from behavior monitoring to behavior change (see table 2.1). As can be seen, for most of these applications, activity recognition is more than overall movement or actigraphy (in where actigraphy is defined as a person's coarse-grained level of physical activity logged over time and analyzed periodically using a watch-like device referred as an actigraph unit (see figure 2-1). Actigraphy units are mainly used in the healthcare domain and, more recently, in the consumer electronics arena in the form of personal activity loggers such as Fitbit [58], Jawbone [80], or Misfit [117]. The activity measurements that these devices provide are usually very coarse-grained and only give an impression on how much movement was measured over intervals of typically from 10 to 60 seconds.

Application	Description
Seniors healthcare	Monitoring activities of daily living and estimating the quality of self-care.
Intelligent environments	Smart spaces like (homes, hospitals, classrooms, offices, etc.), smart cars, smart interactive public spaces, energy efficiency.
Fitness and well-being	Fitting obesity or motivating sedentary people to be more active.
Workflow monitoring	Tracking repairs or maintenance tasks, support performance
Security and military	Security at airports, train stations, banks, smart parking systems, authentication, solder monitoring in the field.
Work and social networks	Displaying members activity on virtual or collaborative environments.
Mobile computing	Context aware applications and content delivery, task assistance.
Memory support	Creating diaries or journals for later memory recalls.
Psychiatry	Correlating activities with moods, mood swings, episodes of depression in patients.
Medical Applications	Personalized mobile healthcare monitors

Table 2.1: Activity recognition applications in the real-world.

Novel real-world applications require activity recognition solutions that focus on fine-granularity information produced by motion sensors which can be detailed enough to distinguish the specific type of activity and how such activity is performed. For instance, medical experts could see detailed patterns of ambulatory and sedentary motion as key indicators for a persons emotional state and/or physical health [30]. On the other hand, recognition of specific gestures or recurring body movements has received increasing attention in communities interested on the rich information produced by fine-granularity activity recognition (e.g., the recognition of specific gestures by dancers or the recognition of non-verbal/paralinguistic actions to study social interaction). Other examples are professional sports communities for assisting athletes in their training, medical communities for assisting patients with their treatment, medication dosage or physical rehabilitation, or epidemiology and preventive medicine communities to motivate sedentary people to increase their level of physical activity. More examples are listed in table 2.1.

In the past, typical approaches to activity recognition would involve RFID tags and external sensors forming part of an architectural infrastructure (such as video cameras or environmental sensors [79, 88]) or highly instrumented body suits [173, 100] measuring acceleration and physiological signals. However, these approaches are difficult to replicate and, besides that, they are costly and intrusive to capture motion trough out the day. Figure 2-2 shows some examples of these approaches.

New approaches aim to provide high amount of information by providing a characterization of

Figure 2-1: Actigraph unit [4].

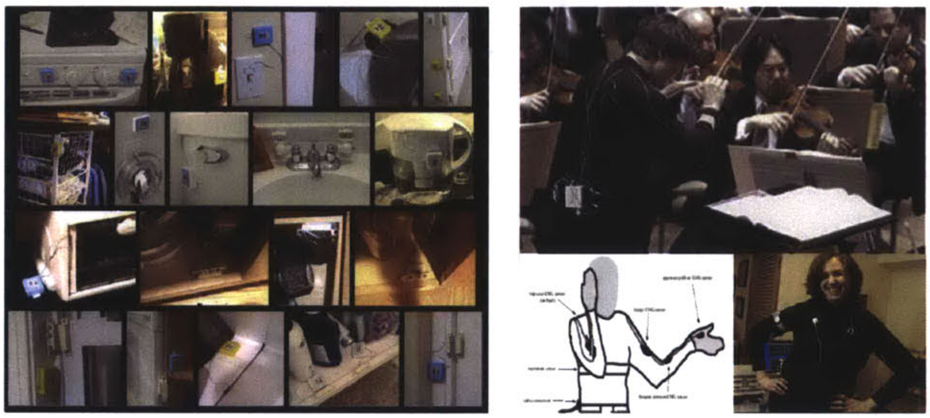


Figure 2-2: Past approaches to activity recognition. Left: MITes installed at the MIT Place Lab (published by Intille et. al.[79] in 2006). Right: Media Lab Conductors jacket (published by Marrin et. al. [173] in 2000).

what type of movement is detected. Novel low-power wireless activity sensors – like the sensor described in chapter 7.2 – provide data that is accurate enough to recognize activities in the users natural environment.

2.3 Types of Activity Recognition Approaches

In general, there are an extremely large number of techniques that differ in both the kinds of activities that they try to recognize (complex vs. low-level locomotion) as well as the robustness with which they accomplish the task depending on the number of specific activities, type of setting, type of subject (children, adult or senior), number of sensors, sensor placement on the body and type of sensor (sampling rate, maximum swing and sensitivity). It is challenging to compare these directly, however, in the next section we will attempt to characterize their main advantages and disadvantages by classifying them according to the type of sensing hardware and the type of data collected. As a result, we will group current activity recognition systems in four fields which are: (1) external sensor-based activity recognition, (2) vision-based activity recognition, (3) large-scale dataset mining activity recognition, and (4) wearable activity recognition.

2.3.1 Sensors Around Us: External Sensor-Based Activity Recognition

This type of activity recognition is based on sensing human activity by using sensors that are around us embedded in the environment (such as environmental sensors fixed in smart infrastructures sensing ambient light, vibration or proximity) or video cameras used as sensors. This approach, which is referred as external sensor-based activity recognition (ESB-AR), consists of decomposing complex activity classes in numerous lower-level activities or specific actions. Thus, the idea is to collect not only observations from the users themselves – such as physical movement – but also, observations about their environment and surrounding objects.

As a consequence, systems based on ESB-AR approach rely on automated highly instrumented sensing environments such as the Neural Network House [120]. The Neural Network House was initially investigated and implemented in the mid 90s within the contest of home automation and location-based applications. This project had the aim to adapt the smart home to its occupants needs based on the recognition of their ongoing activities in the surrounding environment. From there, an extensive research effort was undertaken to investigate the use of sensors to recognize human activity and its context across many application scenarios, effort which originated the fields of context awareness, smart objects, and vision-based activity recognition [156, 144, 64, 182].

Figure 2-3 shows examples of other ESB-AR approaches such as the Home Lab (Philips Research) [140], the Place Lab (MIT) [79], and the Aware Home (Georgia Tech) [88]. These projects (developed from early to mid 2000s) went extreme at instrumenting a home environment with hundreds of sensors. Indeed, even though they did relatively well (˜60%-70%) at recognizing the well-known activities of daily living (ADL) compendium (which is used to estimate the quality of self-care), their impact in the real-world has been rather small. After several years of research, it has been found that it is quite difficult to transfer ESB-AR solutions to environments with fewer resources. Consequently, after numerous extremely expensive projects, the research community realized that such level of investment is rather unreasonable if the end-goal is to develop scalable real-world applications. Furthermore, recent experiments performed by Biswas et. al. [19] have shown that it is quite challenging to replicate ESB-AR solutions across different environments even for the same and simplest single-user single-activity application scenario.

In sum, the previous examples illustrate that activity recognition approaches based on ESB-AR are very difficult to generalize and/or scale up in the real-world. This is mainly due to the extensive instrumentation needed and the difficulty of transferring results from one environment to another. In addition, under this approach, the problem of recognizing activities from a high-level abstraction to low-level one becomes significantly harder since activities performed in natural settings can be executed in highly complex manners activities can be interleaved, overlapped or parallel (e.g., housekeeping, watching TV and eating, etc.).

Figure 2-3: Examples of other external sensor-based approaches. Left: MIT, Place Lab [79]. Center: Philips, Home Lab [140]. Right: Georgia Tech, Aware Home [88].

2.3.2 Sensors Watching Us: Vision-Based Activity Recognition

In addition to external sensor-based activity recognition, there are several approaches that have used environmental sensors alongside video cameras in order to take advantage of the advances in the field of computer vision. For example, Weindland et. al. and Yilmaz et al. [187, 199] have reviewed the tracking of objects for activity recognition using computer vision.

Vision-based activity recognition (VB-AR) alone has been widely explored given its importance in areas like building security, robotics, and surveillance. VB-AR approaches can use a wide range of modalities that go from multi-camera stereo to infrared vision systems, and from single to groups of individuals. In particular, a large body of research has been published surveying vision-based systems for activity recognitio such as the classic review done by Aggarwal and Cai [6] or, more recently, the reviews done by Turaga et. al. and Moeslund et. al. [177, 118]. Specifically, these papers discuss key problems and novel approaches on VB-AR involving body-motion capture by video cameras. More recently, comercial devices (such as the Kinect [201]) have been used for VB-AR. However, they are also subjected to the same drawbacks discussed by Turaga et. al. and Moeslund et. al. [177, 118]. Figure 2-4 shows an example of VB-AR system using the Kinect [91].

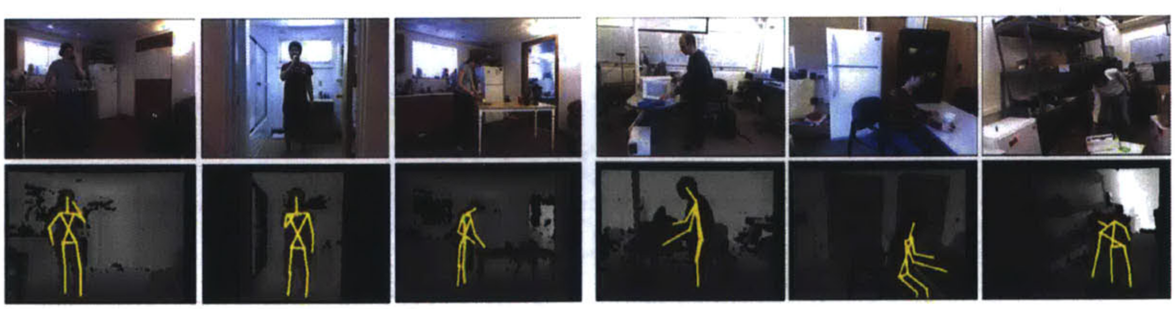


Figure 2-4: Example of a VB-AR system using the Kinect (as contained in the Cornell activity datasets and code repository [91]).

Certainly, the body of research in the field of VB-AR is very large and, despite its maturity and noticeable advantages, VB-AR approaches have still numerous unresolved problems and biases. Many of these problems are linked to privacy and ethics – as cameras are often negatively perceived as very intrusive recording devices, even if they are used only as sensors [32, 199, 63]. Other problems are related to datasets biases [175], limited mobility, high-power consumption, and the need of an external infrastructure, among others.

2.3.3 Sensors Tracing Us: Large-Scale Dataset Mining Activity Recognition

There are numerous types of systems in where human behavior can be recognized using activity models that are learned from large-scale datasets. For example, some of such datasets can be: behavioral information of large communities based on mobile phone communication logs [47, 119], buying activity of credit card transactions or twitter [125, 93], online activity patterns based on weblogs or twitter data [126, 197], or mobility behavior based on human location history collected via mobile phones [116, 203, 46, 67]. In these examples, users behaviors and activities are recognized using data mining and machine learning techniques. This approach, known as the big data, requires creating a statistical model, followed by a learning and a training tasks. Figure 2-5 shows an example of a dataset mining activity recognition approach.

Figure 2-5: Example of dataset mining activity recognition using mobile phone usage logs and wifi and bluetooth proximity detection (as described in [47, 104]).

Since this approach is driven by pre-collected data, it has the advantage of being able to handle uncertainty and temporal information. But it also has several disadvantages. For instance, since it requires having already a large dataset for training, this method can be impractical for modeling fine-grained activity recognition. Depending on the behavior, the required data might be scarce or unavailable (like data for physical activity, activities of daily living, natural gesture understanding, or behavior change). Besides, this method suffers from the problem of cold starting the statistical model [96]. This problem is present given that the system cannot draw any inferences from new users or their behaviors for which it has not yet collected sufficient information. In particular, this problem is more prevalent in approaches relying on information filtering methods such as collaborative filtering or content-based recommendation, which use a user profile to infer information about a new product or movie (like in the Netflix or Amazon recommender systems.).

2.3.4 Sensors Living With Us: Wearable Activity Recognition

More recently, researchers have turned their attention to Wearable Activity Recognition. In general, this approach involves techniques that start by recognizing lower-level actions using wearable sensors and builds up towards recognizing high-level and complex activities. It has the advantage of requiring much less instrumentation, power, and back-end infrastructure than other approaches.

Certainly, the idea of sensing activities through wearable sensors is not new, it had existed since late 1990s/ early 2000s. However, the reason why it became feasible recently is that sensor technologies have evolved at the point of being realistically deployable in terms of network communication, infrastructure, cost, hardware size, and power efficiency. In addition, it has been found that wearable activity recognition better address sensitive issues in terms of privacy, ethics, and obtrusiveness than conventional vision-based approaches [138, 115, 196].

Indeed, in recent years wearable activity recognition has received more attention in the field of mobile computing and, so far, the most popular wearable sensors used in the field are accelerometers. These sensors are very effective for monitoring actions that involve repetitive motion or still positions like physical activities and postures (walking, running, standing, etc.). Bao et al. [15] and Kern et al. [87] deployed a body-sensor network of 3-axis accelerometers distributed on the users body and recognized human activities using acceleration signals in which each accelerometer provided a motion measurement of the corresponding body location. Lee and Mase [112] measured the acceleration and the angle of the users thigh to determine a users location and recognize sitting, standing and walking behaviors using a dead-reckoning method. Whereas, Mantyjarvi et. al. [111] recognized those activities on acceleration data collected from the hip. Figure 2-6 shows past and new approaches to wearable activity recognition.

Indeed, efforts within this line of research have achieved promising results by sensing physical movement using accelerometers worn on the body and by classifying the sensed patterns of body motion using supervised or semi-supervised machine learning algorithms (like Decision Trees (DT) [169, 11] or Hidden Markov Models (HMM) [131]). Notwithstanding the encouraging results, the impact of these approaches on real-world applications has been limited by the challenges characteristic of a natural setting. Once moving beyond demonstrators, these approaches have several important limitations such as: the high degree of user and situational dependency, the need of having

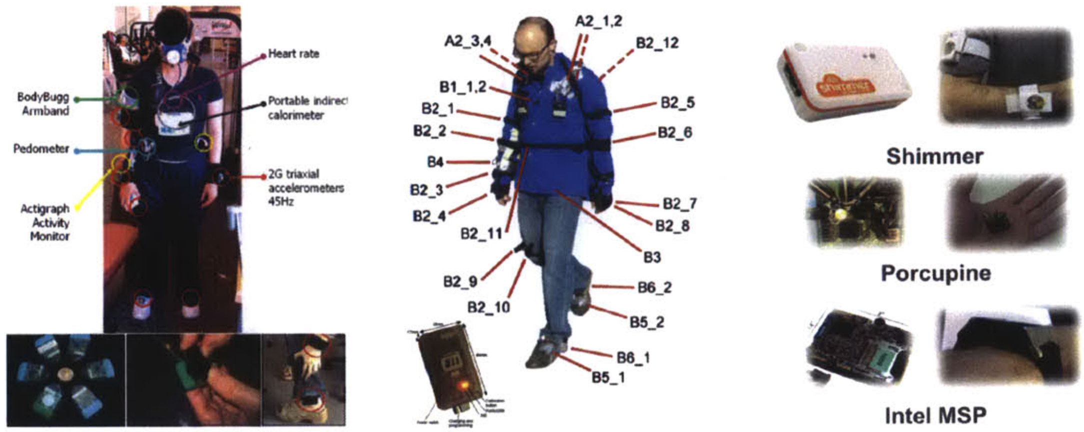


Figure 2-6: Past and new approaches to wearable activity recognition. Left: MITes [169]. Center: Activity Recognition repository setup collected at ETH [150, 51]. Right: Newer sensing platforms used in research [159, 94, 50].

large amounts human labeled data, and the lack of capability to accommodate the high variability caused by the differences in how activities are performed across users and settings. These limitations have been largely identified by the Pervasive, Wearable Computing and Ubicomp communities but surprisingly not solved yet.

2.3.5 Conclusion: There is no Panacea

It is important to note that wearable sensors alone are not suitable for monitoring activities that involve complex physical motions and/or multiple interactions with the environment. In most cases, this type of observations alone are not sufficient to differentiate activities involving physical movements but require a higher-level understanding in terms of context (like making tea versus making coffee). As a result, multi-modal and human-object interaction sensing is needed as part of the dense sensing-based activity recognition approach described earlier.

For example, Tapia et al. [171] used environmental state-change sensors to collect information about human interaction with objects and recognized activities of daily living that could be used for medical professionals to assess seniors health and well-being. In subsequent work, Wren and Tapia [196] employed passive infrared motion sensors to detect presence or movement of heat sources to recognize low-level activities like walking, wandering and turning. With lower accuracy, they also detected mid-level activities such as visiting and meeting someone.

Hence, wearable activity recognition and other approaches - such as external sensor-based activity recognition (ESB-AR)- are not mutually exclusive. Actually, there are several applications that have combined more than one approach successfully. For example, Philipose et al. and Fishkin et al. [139, 57]developed the iGlove and iBracelet which are devices that are used as RFID readers that detect when the user interacts with unobtrusively tagged objects. This approach requires that objects are instrumented with tags and users wear an RFID reader fixed to a glove. Buettner et al. [25] recognized indoor daily activities by using an RFID sensor network. Whereas, Gu et al. [70] combined wearable sensors and smart objects to collect multimodal sensor information. As shown by these examples, wearable sensors and dense-sensing based approaches are complementary and can be used in combination for improving the activity recognition results.

2.4 Activity Recognition in the Wild: Why is it Hard?

In light of what has been discussed up to now, it is evident that human activity recognition is not a trivial problem. Humans are very complex creatures and such complexity increases when their behavior is studied in naturalistic and unconstrained settings. In this thesis, we focus on recognizing activities in this type of settings using a wearable system aimed to work continuously in real-time for long-periods of time.

As seen through the characterization of different wearable activity recognition systems, existing research provides a large number of approaches that differ in both, the kinds of activities that they try to recognize and the robustness with which they accomplish the task depending on ecological factors (e.g., type of setting, subject, device specifications, etc.). But, as the scope of the system broadens from carefully controlled experiments to real-time measurement in natural settings, such approaches are overwhelmed by the intrinsic heterogeneity of users behavioral patterns.

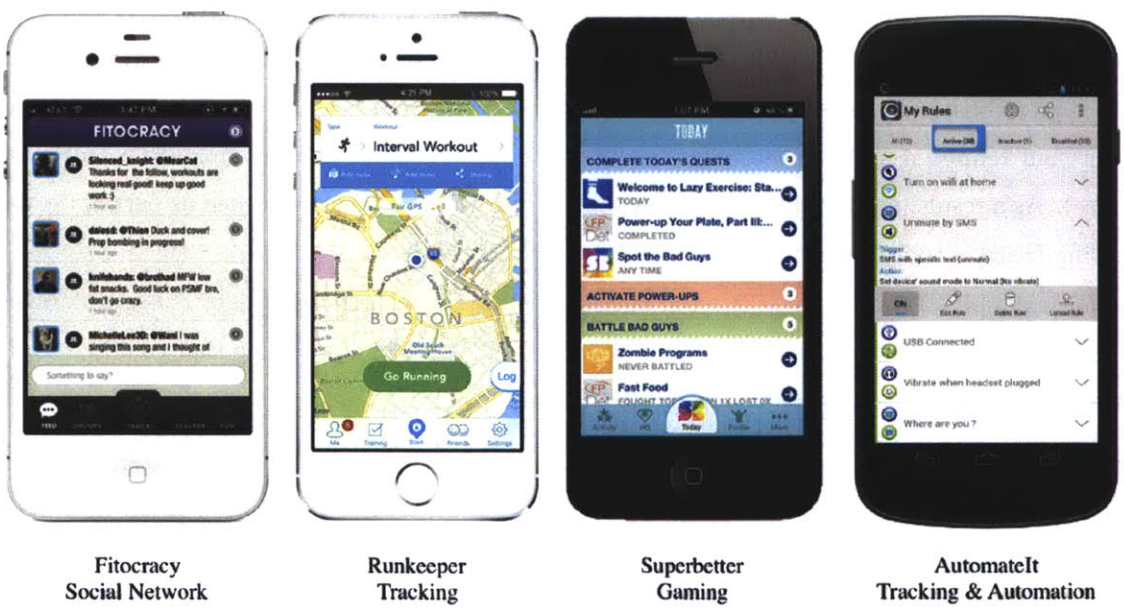


Figure 2-7: Example of mobile phone based commercial activity recognition apps.

As a matter of fact, there are several approaches that can recognize short-term physical activities and postures performed in constrained laboratory settings (like walking, sitting, running, etc.). In fact, such approaches have leaded to the development of several commercial applications, particularly, in the form of mobile apps some of which are Fitocracy [60], Runkeeper [59], Superbetter [167], and AutomateIt [14] (see figure 2-7 to see UIs of these examples). Regardless their wide-spread use and accessibility, in practice, these apps cannot be used reliably to capture continuous behavior for long periods of time since they dont provide good inter-operability, they dont collect high-quality fine-grained raw data and, often, they dont work as advertised. Usually, this is true for medical applications aimed to study motor-related disorders (like parkinsons disease or stereotype movements suffered by autistic children), mood-related disorders (like bipolar disorder or depression), movement disorders caused by physical injure, or sleep patterns.

As wearable activity research has influenced the development in other fields (like commercial mobile apps or mobile context awareness), other fields of research – such as computer vision and speech recognition – have extensively influenced wearable activity recognition. Several concepts coming from those fields, such as crowd-sourcing annotation, face and signature identification, location traces, etc., have started to be explored with in scope of wearable activity recognition research. As a consequence, there are numerous methodological similarities, as well as, distinctive challenges among all these fields.

For instance, as described in Bulling et. al. [26], in the fields of computer vision and speech recognition you can describe a well-defined problem such as detect an object in the image or detect a spoken word in a sentence, which leads to focusing into a specific recognition system. In contrast, wearable activity recognition is significantly more complex since it has higher intrinsic variability, it has several interleaved or/and overlapping levels of abstraction, and it requires more degrees of freedom in terms of recognition and implementation. Besides, there is no common definition, language or structure of human activities that would allow formulating directly a well-defined problem (e.g., how an activity is characterized). Inclusively, for continuous long-term activity recognition in natural settings, relevant activities cannot even be defined upfront across all users.

In addition, human activity is highly diverse and, therefore, its recognition is subjected to sensor arrangement (on body location) and variability due to environmental or contextual factors. As a result, wearable activity recognition systems face distinctive challenges that require a dedicated set of computational methods and evaluation metrics than those used in other fields. In sum, the challenges faced by wearable activity recognition systems are an overlap between the sensing hardware, the activity recognition algorithms, and the user interaction with the system (as depicted in figure 2-8). The specifics of such challenges will be discussed in detail in the rest of this chapter.

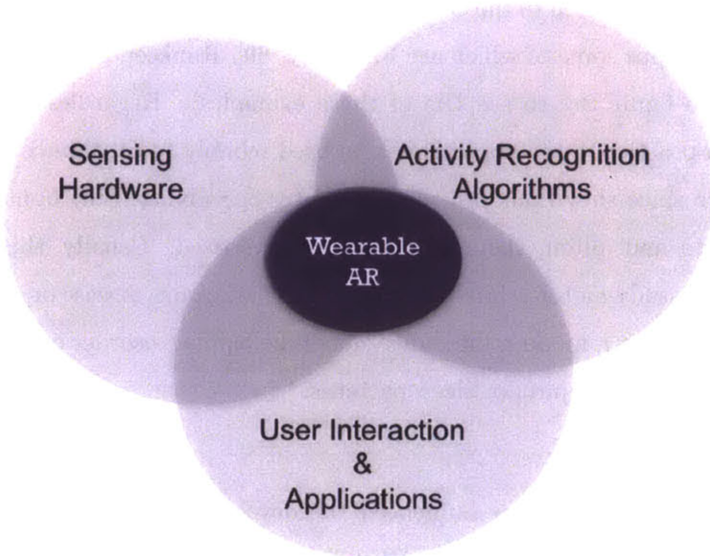


Figure 2-8: Challenges faced by wearable activity recognition systems are an overlap between the sensing hardware, the activity recognition algorithms, and the user interaction with the system.

2.5 Critical System Design Challenges

This section discusses the main challenges faced by wearable activity recognition systems, their practical impact on real-world applications, and critical design trade-offs.

2.5.1 Sensing and Hardware

Wearable activity recognition has many limitations caused by the system hardware. For instance, most wearable sensors need to run continuously and be operated/programmed wirelessly. This carries many challenges for real-world applications deployed in naturalistic settings. Practical issues include the acceptability and viability to wear the wearable sensors (specially for applications that require 24/7 behavior monitoring). Technical issues include the size, ease of use, battery life, and effectiveness of the hardware design to cover the needs of real-world scenarios.

Indeed, the solution of technical issues has been discussed for many years (over a decade). This slow pace reflects how hard it has been, for instance, to go from placing sensors on the body without having to wear-on a backpack with a laptop and several wires/cables connecting the on body sensor network - as shown in figure 2-9-, which only would work for a couple of hours. In fact, the battery life span has been a significant limiting factor.

In general, it is important to realize that wearable sensors are highly variable; not only in their types and output signals but also in their size, weight, and cost. Even for a particular type of sensor (e.g., inertial sensor), the output, size, weight, and cost can vary significantly across different models. For instance, there are several types of accelerometer monitors (loggers and smart phone

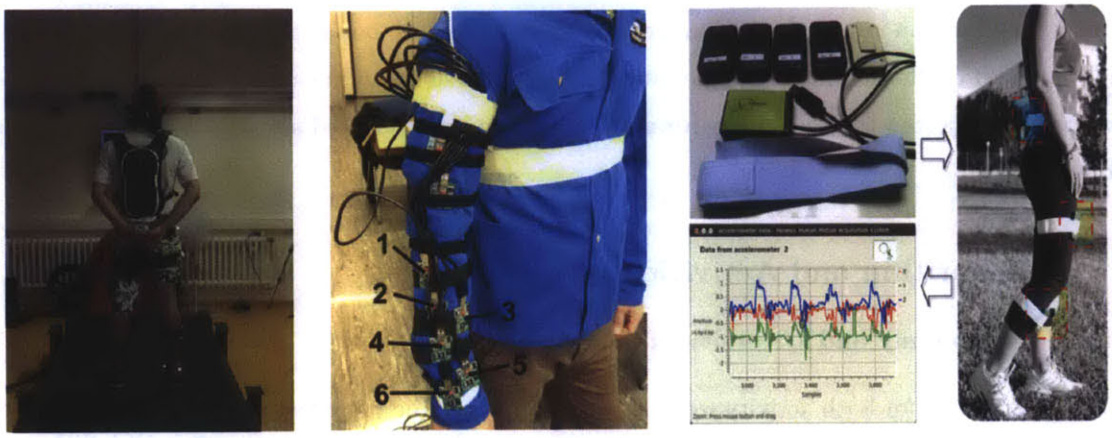


Figure 2-9: Examples of approaches to wearable activity recognition requiring a PC and developed as a body area network (WBAN). Left: NT-Neuro eegosports wireless EEG and motion system [49]. Center: Gesture & physical activities database ETH [52]. Right: HenesisWii Module [68].

based sensors) that are available from commercial vendors like (actigraph [4], sensewear [22], fitbit [58], nike fuelband [127], jawbone [80], etc.). Figure 2-10 shows examples of devices available in the market today. These sensors have different sizes, cost, measurements algorithms, communications, battery life, etc. Although, the specifics of these differences are beyond the scope of this thesis, we want to get across that these aspects have to be taken into account when designing the wearable activity recognition system itself in order to be able to compare and interpret the bias introduced by these hardware differences.

Figure 2-10: Example of currently available wearable devices measuring acceleration.

More recently, several researchers have opted for making use of existing gadgets that people carry in a daily basis (like smart phones, smart watches or more recently google glass) as activity monitors (see figure 2-10). This approach has been practiced for a while [64, 155] and this trend is expected to increase and have an impact for the understanding of large-scale population-level behavior given the wide-spread use, affordability and increasing processing capability of such devices. Nonetheless, these sensors are subjected to the same hardware biases than the dedicated activity monitors previously mentioned.

2.5.2 Algorithm and Activities

2.5.2.1 Taxonomy and Diversity of Activities

A major challenge for both activity recognition systems and their applications is the development of a clear definition of the activities under investigation and their specific features. At first, this might seam trivial. However, human activity has a complex structure, it can be performed in different manner, and it is highly subjected to contextual factors. Issues, such as overlapping or interleaving execution, variability, etc., are less of a problem for short-term simple activities performed in controlled environments. However, they are not trivial for long-term activities performed in natural settings or activities that involve complex actions. Figure 2-11 shows examples of these issues.

Figure 2-11: Taxonomy issues typically faced by activity recognition systems. Left: walking or eating?. Right: Activities could be performed in unexpected manner by different people and could happen anywhere with high variability.

Providing a well-defined compendium of activities has served as a guideline for recognizing activities relevant to the real-world applications. Typical compendiums include: (a) Katz et al. 1970 [86] who introduced for first time the activities of daily living (ADL) index as a tool of estimating quality of self-care among seniors; and (b) Ainsworth et al. [9] compendium, which groups physical activity in categories based on metabolic equivalent (MET) and it is often used in energy expenditure applications [11].

Another resource for activity recognition definition is given by time use databases like ATUS [27], MTUS [34], and HETUS [168] . These databases are assessed by the government to understand citizens time use. Borazio et. al. [23] and Partridge et. al. [135] have investigated the use of these databases for activity recognition. In particular, besides providing prior probabilities for activities for a certain time of the day and location, Borazio et. al. [23] compared the time use databases between USA and Germany. This comparison reveled that there are significant differences on when, where and how activities are performed due demographic differences (e.g., subject age, country, geographic location, culture, occupation, etc.). For example, differences were found due to occupation (rhythmic routine of a construction worker versus chaotic routine of a student) and models of transportation within a city (USA west coast which is heavily based private cars versus Germany which is heavily based on public transport).

These examples highlight that cultural and occupational impact should not be disregarded when designing activity recognition systems. In general, time use databases offer a valuable resource that can serve as taxonomy for activity recognition researchers investigating long-term activity patterns and characterizing daily routines. For example, activity descriptions can be mined from the Web or contributed by users like in the work done by Philipose [170]. In such case, the hierarchical recognition approach would be greatly simplified if there were a set of key relevant activities and temporal and causal models that can be used for a broad group of people. For example, a simplified set of activities that describe how people use their time in the morning, etc. Nevertheless, it is evident that a complete and a consistent categorization is not feasible since we have to learn what information about the activity is relevant for the potential application (level of abstraction) and, in addition, activities might have a considerable variation in execution (be performed in an interleaved fashion, in different order, at different speeds, etc.).

As a result, hierarchies become relevant since they are allow recognition at different levels decomposing the problem into simpler ones. For example, the work carried out by Blanke and Schiele [20] argues that hierarchical structures can improve the performance of activity recognition systems and facilitate their use across multiple types of applications. Indeed, the work presented in this thesis has being inspired by this line of thought. Specifically, this work supports the idea that the activity recognition problem can be simplified by recognizing activity primitives based on simpler activities (e.g., physical movement, interaction with objects, and so on) and, the knowledge required to recognize such simpler activity primitives can be used to learn and transfer knowledge across activities and users.

2.5.2.2 Complexity of Activities

If we go back to the definition of activity recognition introduced at the beginning of this chapter (section 2.2), it is not difficult to realize that the recognition of actions and goals is not a trivial problem given that human behavior is intrinsically heterogeneous and complex.

Some of these complexities are as follows:

1. Intra-class variability: Activities performed in naturalistic settings exhibit high intra-class variability because there are many possibilities in how the same activity can be performed by different individuals. People perform activities in a diverse manner depending on time, place, social surrounding or, inclusively, hand dominance. Moreover, the interpretation of the activities varies among people. There is even differences in multiple executions of the same activity by a single individual (for example, vacuum cleaning is an example of such type of activity). All these factors make the general signature characterization of natural occurring activities across people to be a very challenging problem.

2. Class imbalance: The duration and incidence of natural occurring activities have a high variation across different activities. There are activities that can last for few minutes only (e.g., smoking a cigarette or taking a pill), whereas others can last up to a couple of hours (e.g., cooking or working in front of the computer). Some other activities might occur on a daily basis (e.g., sleeping, eating), whereas others are less frequent (e.g., doing the laundry or leisure activities).

3. Inter-class similarity: Contextual factors (e.g., object usage and location) can aid the recognition of an activity based on movement. Nevertheless, none of these characteristics alone can be used as a unique representation of an activity. For instance, several activities might require similar hand movements (e.g., booming and vacuuming). Analogously, the same object might be used in different activities (e.g., dinner table can be used for eating or working) or the same activity can take place in different locations (e.g. one might eat at the kitchen counter, dinner table, or office desk). In some cases, there is a one-to-one mapping between the activity and the object or/and location (dishwashing in front of the sink or showering in the bathroom). However, often those mappings present exceptions and, in most cases, there is a lack of specificity among many natural occurring activities.

4. Simultaneity: Activities can be performed in overlapped or interleaved manner and they can be composed of sub-activities which order may vary. For example, when two or more activities are performed at the same time (e.g., driving and talking over the phone, walking and writing a text message on the phone, or watching TV and eating), they are defined as overlapping activities. On the other hand, interleaved activities happen when the next activity starts before the previous activity is completed (e.g., cleaning the house in which dishwashing, mopping and folding clothes activities can overlap at different points in time). Finally, an interrupted activity takes place for a certain period of time and, then, stopped and continued in a later time (e.g., writing, playing music, and other leisure activities).

5. Null-class problem: Typically when deploying activity recognition systems in the real-world only a few parts of a continuous data stream are relevant for recognition task. Given this imbalance of relevant versus irrelevant data, activities of interest can easily be confused with activities that have similar patterns but that are irrelevant to the application of interest which is called NULL CLASS. The NULL CLASS is a large unknown space that that can be ambiguous and increases the confusion among the activities of interest. To generate an explicit model of the NULL CLASS is difficult if not impossible since it represents a theoretically infinite space of arbitrary activities. In chapter 3, we discuss how this problem impacts the approach presented in this work.

Figure 2-12: Examples contrasting long-term versus short-term activities and simple versus complex activities.

Finally, it is important to highlight that continuous long-term activity recognition goes beyond recognizing simple body locomotion typically lasting for short periods of time (actions). It is the recognition of more complex behaviors (activities) consisting of a sequence of overlapping or interleaving actions. If we make an analogy with the field of speech recognition, recognizing actions is like recognizing words whereas recognizing activities is like recognizing the meaning of a sentence being said. Similarly to natural language understanding, activities performed in natural settings are continuous and can have different interpretations according to context, previous knowledge and objects/people in the surroundings. Figure 2-12 shows examples contrasting long-term versus short-term activities and complex versus simple activities.

2.5.3 User Interaction and Ground Truth Annotation

Ground truth annotation is an important problem in the field of activity recognition given that most approaches rely on annotated data. This problem is significantly more difficult when activities are collected for long-term in naturalistic settings than in controlled experimental settings.

Experimental settings introduce several biases that affect the validity of the data collected. One of them is related to the observer effect or Hawthorne effect [5], which is a form of reactivity in where subjects modify their behavior in response of the fact that they are being studied and aware of the presence of the activity recognition system. Another one is that typically activity recognition experiments involve few subjects (sometimes/often, inclusive, the researchers who develop the system) and sets of scripted activities. In general, these factors tend to oversimplify the recognition problem and, additionally, the activities can unconsciously be performed in a way that helps the system during the recognition task.

Hence, long-term collection of activities in naturalistic settings enables the capturing of more realistic data that better represent what is observed in real-world applications. For this reason, activity recognition research has been moving towards the collection of data captured in naturalistic manner. However, capturing accurate ground truth annotations in naturalistic settings is a challenging problem. Previous work in activity recognition makes the assumption that labels are consistent. While this might be true in simple experimental settings, in naturalistic settings labeling issues (such as annotator annoyance or disruption, assignment of different labels to activities that are similar in a particular hierarchy, activity labels boundary overlapping, etc.) start becoming a concern [137].

There are several annotation methods to capture ground truth for long-term in naturalistic settings. Nonetheless, most of those methods are error-prone and time-consuming, can use invasive sensors that not acceptable due to privacy reasons, or can be disrupting or annoying for the user who typically has to annotate the data specially when detailed annotations are needed.

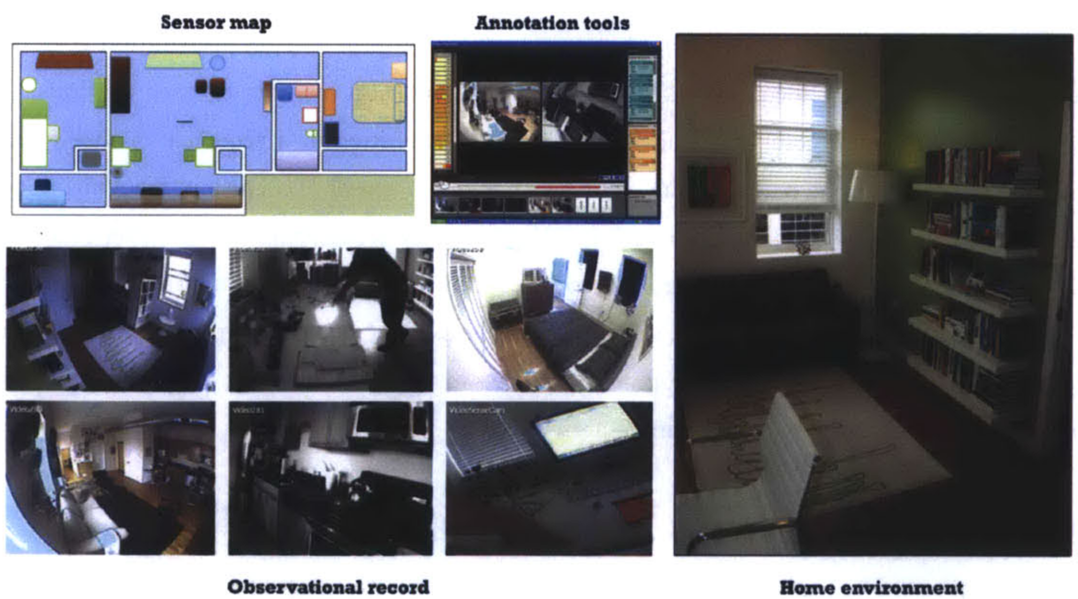


Figure 2-13: Example of a direct observation offline system called Box Lab [77], which annotates activities using video footage captured in a post-hoc manner.

An important distinction between them is the time when the annotation takes place. In general, there are three types of annotation methods: offline, online and hybrid. The offline annotation method relies on annotations that have been assigned after the recording has been finished (generally assigned by an observer). Whereas, the online annotation method depends on annotations given in real-time during the execution of the activity (generally assigned by the user). Hybrid methods combine offline and online annotations and annotators (observers and users). This method attempts to minimize bias by annotating the activities from more than one standpoint combining different techniques that balance each other out: user vs. observer, real-time vs. offline, self-reported vs. prompted, short-engagement vs. long-engagement, etc. A more detailed analysis about the pros and cons of each method will be introduced in the next subsections.

2.5.3.1 Offline Methods

These methods are based on annotating the data after that the recording has happened.

In many cases annotations are obtained by direct observation (often by an external observer) in where an observer labels the data in a post-hoc manner based on video or footage [20]. This technique has been often applied in controlled and short-term experiments[204] given that it can be very accurate and unbiased. Nevertheless, it is difficult to use it in long-term studies taking place in natural settings in where annotations 24/7 are not possible or labels obtained in this way are costly. Logan et. al. [107]and Intille et. al. [79] use it for annotating a long-term activity recognition experiment in an instrumented home. Logan et. al. [107] reports that, on average, an hour was spent to annotate 1.5 hours of data. Besides this disadvantage, this technique has low user acceptance due to privacy concerns and scales poorly to a large number of users. Figure 2-13 depicts a typical annotation setting based on this method.

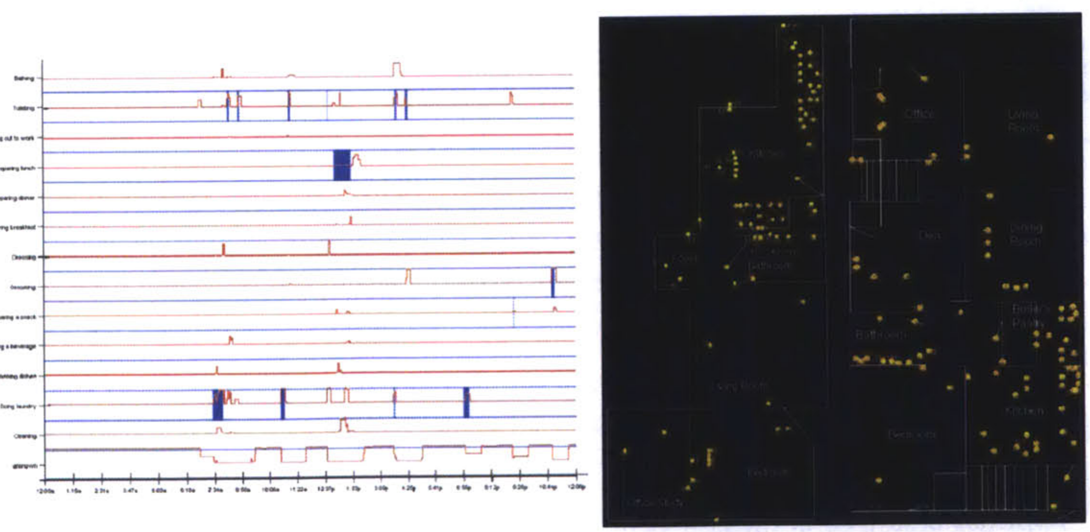


Figure 2-14: Example of an offline method based on self-recall or self-report (as used in [172]).

Other technique includes daily self-recall or self-reporting, which is less intrusive than direct observation but suffer from recall and selective reporting biases [184]. As a consequence, the quality of the ground-truth annotations is poorer than other approaches. Tapia et. al.[172]used indirect observations of sensor data along with users self-recall to label the data. They reported that environmental sensors - such as RFID tags, switch sensors, and light sensors might contain enough representative information that a user can recall or identify the activity from a sensor stream. However, in the case of wearable sensors like data recorded from accelerometers or gyroscope, the data is generally not intuitive and more difficult to interpret than data from other sensors (RFIDs, environmental sensors or cameras) [189]. Thus, they might require an experienced user to identify the activity from the data stream. Figure 2-14 shows a typical example of this approach.

2.5.3.2 Online Methods

Online methods involve frequent onsite annotations while the activity data is recorded. Similarly to the offline methods, online methods can be categorized in terms of whom is annotating the data (an external observer or the user herself).

The first category involves the online annotation of the data by an external observer. This method is mostly used in short-term controlled experiments and it slightly reduces the annotation cost when compared with offline direct observation methods since the annotator only needs to verify the accuracy of the annotations done online. Therefore, it is mainly implemented to improve the annotations reliability.

The second class involves techniques in which the user annotates the data. This can be carried out through time diaries (TD) or ecological momentary assessment (EMA).

When using a time diary the user has to log her ongoing activity along with the activity start and end time (passive annotation). Researchers who have investigated this type of annotation method [184, 74, 83] have reported that is highly prone to self-recall errors and requires a high-level of awareness about the ongoing activity.

On the other hand, ecological momentary assessment (EMA) - also called experience sampling - consists of capturing online annotations during the data collection by periodically prompting the user to provide information about her current activity. Several computerized versions of EMA have been developed [62, 78] in which, essentially, a mobile device (phone or tablet) prompts for information about the users activity by an audio or visual cue. Barrett et. al. [16] provides a comprehensive review of computerized versions of this technique. Figure 2-15 shows a typical use scenario of the EMA method.

EMA has been used in the field of physiology for many years [43, 188, 72]. More recently, it has generated substantial interest not only among the activity recognition community but also in the ubiquitous [40, 41] and mobile health [41] communities. Consequently, several lessons about its implementation have been learned. For instance, it has been seen that it has several advantages such as: (1) the user is reminded to annotate her ongoing activity in a way that requires less awareness; (2) the prompts can be randomly sampled through out the day to capture a wide-range of circumstances

Figure 2-15: Example of a typical use scenario of the EMA method.

experienced in everyday life by the user; and (3) it is less prone to recall errors and, consequently, the annotations are more reliable.

However, despite all these advantages, it also has several disadvantages. For instance, naturalistic data annotated in online manner could be biased simply because, by annotating the activity, the user might change the activity itself. Other disadvantages are: inaccurate annotations due to low-level of users attention, impossibility to respond to prompts due to the type of activity that is being performed (e.g., office related activities versus sport related ones), and missing annotations due to users perception of interruption. In fact, the frequency of the prompts is directly proportional to the degree of user interruption. Thus, the higher the number of prompts the higher the method becomes burdensome, annoying and/or irritating.

Nevertheless, work carried by Picard et. al. [141] suggests that this effect might be mitigated if the prompting is done when the user perceives that is a good time to interrupt. For instance, if the user is prompted during a break when she is alone, she might perceive it as a welcomed interruption. In contrast, if she is prompted during a meeting or during a conversation, the prompt is likely to be perceived as an annoying or a burdensome interruption.

Moreover, work carried out by Intille et. al. [73] and Froehlich el al. [62] indicates that the prompts could be initiated at more appropriate times by context-aware events. In this approach sensor events are used to determine when to prompt the user in order to minimize interruptions and maximize the capture of valuable and interesting information. The sensors could be embedded in the mobile phone (e.g., GPS, call logs, etc.) like shown in [62], they could be embedded in the environment like shown in [78, 171], or they could be worn on the body like shown in [105, 76].

However, the caveat is that in many cases there is not possible to determine in advance what valuable or interesting events might be present and if they could be detected by sensor data. Thus, the system could end up missing valuable information or causing higher number of interruptions than in the non-sensor case – just simply caused by imprecise sensor tuning.

2.5.3.3 Hybrid Methods

Another possibility of annotating the data is by using hybrid methods. These methods try to deal with the disadvantages of conventional approaches by providing support to users to make more accurate online annotations or to trigger their memories when annotating the data offline.

Regarding online annotations, Kapoor and Horvitz [85] used experience sampling and active learning to determine when to prompt the user. The authors developed a method to weight the trade-off between the information collected by a prompt versus the degree of interruption caused to the user.

On the other hand, a more hybrid approach has been investigated by Huynh et al. 2008 [74]. This approach combines online and offline annotation methods for allowing the user to choose a method that is more suitable according to her ongoing situation.

Finally, several hybrid offline approaches have been developed. For instance, Tapia 2004 [171] clustered sensor data (state change sensor data collected in a home environment called PlaceLab by activation time and location to make annotations via indirect observation. Another approach is to use wearable microphones and cameras worn by the users as shown in [189]. This method, referred by the authors as context-aware recognition survey (CARS), groups the sensor data in contextualized clusters called episodes, which are converted in a sequence of meaningful images that are shown to the user in an offline manner via a game-like computer program. Specifically, the user attempts to correctly guess the ongoing activity after seeing a series of images representing sensor values generated while the activity is performed in an instrumented home environment. More recently, Stikic et al. [164] have used a semi-supervised approach which provides a graphical representation of the recorded sensor data to the user. The graphical representation contains both labeled and unlabeled data. The method aids the annotation process by using feature similarity and time.

2.5.3.4 Methods in Comparison

As it can be seen, to obtain a reliable ground truth annotation requires a compromise between accuracy and the timing of the annotation. In addition, despite their numerous advantages, state-of-the-art methods still have several drawbacks. For instance, current offline methods require arduous labeling, scale poorly to large number of activities and users, and are subjected to many privacy concerns. In addition, since they are based on post-fact labeling, they can cause self-recall errors, lack of temporal precision, and inaccurate annotations. On the other hand, online methods such as

EMA require user participation, which could lead to frequent interruptions that might change the activity itself or/and disrupt the user. As a consequence, for real-world problems, annotating the collected data with existing online or offline methods is an expensive task. However, in contrast, obtaining large quantities of unlabeled sensor data is relatively easy - since often the activities of interest are performed in a daily basis.

Thus, the work presented in this thesis focuses on the case of decreasing the number of user interruptions and creating an algorithm capable of learning upon use from a small number of training examples. To effectively reduce the number of user interruptions, such examples need to be collected in a way that the information provided to the classification task is maximized while the number of instances is minimized. This is achieved by using an active learning approach, which is explained in detail in chapter 3 in this thesis.

2.5.4 Complexity of the Experimental Design and Evaluation

Across this chapter, it can be seen that state-of-the-art activity recognition approaches are hard to reproduce and evaluate. This results from the fact that, differently to other fields, such as vision or speech recognition, activity recognition is not yet mature and well-established field. Consequently, several non-trivial challenges are present in existing research. For instance, activity recognition systems are difficult to reproduce, there are not clear activity taxonomies, there is lack of well-defined experimental protocols, there are not standardized datasets to use as benchmark, and there are not sufficient adequate evaluation metrics.

Activity recognition systems are difficult to reproduce because most data collections focus on quite diverse requirements - like high quality of the data, fine-granularity of sensor recordings, large number of users, large number of sensors, sensor multi-modality, or long-term sensor recordings- and there is not effort of collecting more comprehensive and collaborative general-purpose datasets.

Collaborative general-purpose datasets could uncover which is the distinctive variability among different activities and places providing a collection of representative training examples for activity recognition systems. If collaborative datasets are combined with a well-defined activity classification scheme, the research community could better investigate the problems of variability and subject-dependent recognition. In fact, through out the multiple arguments presented in this thesis, we advocate for making these datasets public and well-defined in terms of their activity taxonomy description since we believe that this will result on faster development and advancement of the algorithms used in the field.

Other important aspects to be considered are the use of unambiguous and well-designed experimental protocols and appropriate evaluation metrics. These aspects are crucial for reproducing proposed approaches and making solid comparisons across different recognition methods which is only possible when they are tested with similar or comparable conditions. They can make a significant impact on moving system prototypes into wide-scale practical deployment. Currently, such type of real-world deployment is quite challenging.

However, to conduct an unambiguous and well-designed experiment or study in activity recognition is more difficult than it might be thought at first. For instance, there are several issues faced such as: to maintain a balance between ergonomics and unobtrusiveness of the sensors versus ease-of-use and performance of the system; to allocate the time and resources required to prepare, conduct, and maintain the experiment; to cover the cost for participants, staff running the experiment, the equipment, and data or phone subscriptions used by participants.

On the other hand, the appropriateness of the evaluation strongly depends on the application; thus, the natural question is to establish what is appropriate. Specifically, in terms of recognition performance, it is not possible to directly compare or evaluate in the same manner the recognition performance of a system evaluated with data collected in an experimental setting covering only few activities and few known-subjects versus a system that is evaluated with data collected in semi-naturalistic or naturalistic settings covering a wide variety of subjects. Obviously, the later is a significantly harder case, which involves much higher degree of variability. Nowadays in current research, these factors are barely considered or properly addressed.

A part from an appropriate evaluation performance, in many cases, it is extremely informative for the algorithm design to provide complete and specific information about the criteria used for optimizing the recognition task since it provides a better understanding of in which cases the algorithms fail.

A comprehensive solution to all these challenges is beyond the scope of this thesis. However, this work provides a detailed account of the criteria used for the activity recognition algorithm design and its evaluation, as well as, the lessons learned from the design and practical implementation of semi-naturalistic and naturalistic data collections and the hardware used to collect the data. It also includes the details of the study protocols and information to allow reproducing the results with the aim that the approach can be implemented in real-world deployments. In general, since this work aims to establish a set of best-practice guidelines that could contribute to the solution of the reproducibility, experimental design, and practical implementation challenges.

2.6 Proposed Framework

In general, many state-of-the-art activity recognition approaches are often proof-of-concept systems carefully tailored to well-specified simplistic scenarios and a small set of users. However, the interest of monitoring a large number of users over long periods of time has increased, as mobile devices have become more accessible and wearable sensors more powerful.

There is an increasing body of research investigating large-scale monitoring of coarse-granularity activity patterns [2]. But, as discussed previously, the field is moving towards fine-granularity wearable activity recognition -which is harder to deploy and research of large-scale systems in this area is not very common. Indeed, typically, wearable activity recognition studies have few participants - especially if they are monitored for several weeks such as the work carried out by van Laerhoven et. al. [184]. Berchtold et. al. [18] have approached the scalability problem by crowdsourcing the activity annotations using an online annotation approach built on the users mobile phones.

In this thesis, we focus on an activity recognition framework that is scalable and can handle an increasing number of users. As most of today state-of-the-art approaches rely on machine learning recognition techniques that require prohibitive amounts of training data for each new user or activity. Thus, the starting point of this approach is to take a fresh look at the problem and design a novel activity recognition framework that is suited to scale-up, learn from few examples, and transfer or add-up new knowledge from previously learned activities across users. The objective is to reduce the required amount the training data to the minimum while reliably recognizing new activities - without lowering the recognition rate of activities that were previously learned by the system.

Moreover, the work introduced in this thesis argues that the proposed framework could be used to create robust building units part of a hierarchical activity recognition model, in where the recognition of high-level activities is based on the recognition results of other simpler activity instances. The motivation is to let the simpler activity instances - which are easier to identify- be first, and subsequently use them as building units for recognizing higher-level ones.

For instance, a high-level activity like fighting may be recognized by detecting a sequence of several pushing and kicking interactions. Thus, in hierarchical approaches, a high-level activity is represented in terms of sub-instances which themselves might be decomposable until atomicity (last level of division of an activity) is obtained. As a result, sub-instances could serve as observations or entries for a higher-level activity recognizer.

Even though, this work doesnt focus on the hierarchical model per se, it is important to explain how it connects with a more generic framework and the logic behind its design criteria. We consider that the hierarchical activity recognition paradigm not only makes the recognition tasks computationally tractable and conceptually understandable, but also scalable and reusable by reducing the redundancy and utilizing common acknowledge or recognized sub-activity instances multiple times.

In general, common patterns of physical motion that appear frequently during high-level human activities are modeled as primitive-level activities. However, as discussed along this chapter, such modeling is not trivial due to high variability and other intrinsic complexities characterizing behavior happening in natural settings. Thus, this thesis aims to provide a framework that allows modeling such primitives in an efficient and scalable manner.

In addition, via lessons-learned and design guidelines, we aim to inform designers about the challenges and trade-offs faced when designing a scalable activity recognition system intended to be used for long-term in natural settings. In fact, we consider that - depending on the available resources and the specific recognition problem- these challenges are not only exclusive of activity recognition systems but also they are common across many wearable systems intended to recognize human behavior or physiological signals for long periods of time in natural settings.

Chapter 3

Algorithm

The automated detection of human physical activities via wearable devices has been suggested for over a decade as an attractive and fundamental component for a wide-range of pervasive applications. However, the monitoring of physical activities at a fine-granularity level in natural settings (under realistic conditions in where users follow their regular routines) has been neglected or, inclusively, completely overlooked by the research community.

Indeed, many state-of-the-art activity recognition approaches are often proof-of-concept systems carefully tailored to well-specified simplistic scenarios and a small set of users. However, the interest of monitoring a large number of users over long periods of time has increased, as mobile and wearable devices have become more accessible and wearable sensors more powerful.

Therefore, this chapter investigates the development and evaluation of a robust machine learning method for recognizing physical activities occurring on everyday life settings, with focus not only on the methods recognition accuracy but also its modularity, performance when adding new information and scalability to a large number of users and activities.

In terms of robustness of the recognition, three important aspects are investigated: dealing with (unknown) other activities, user adaptability and computational efficiency when new information is incrementally added, both explained in more detail in the next subsections. Methods to handle these issues are proposed and compared.

The method proposed in this thesis has been thoroughly evaluated using publicly available datasets (described in section 6), with the aim of being accessible, reproducible, comparable, and adaptable to support several hardware and software platforms and not only the one described in this thesis. Also, experts and non-experts researchers or developers can utilize it to design robust physical activity recognition systems with the selected generalization characteristics.

3.1 Classification Problems

Building on the idea of incremental learning, the algorithm proposed in this thesis aims to provide a solution that addresses the scalability, adaptability, and computational efficiency requirements imposed by long-term physical activity recognition systems deployed in uncontrolled settings at population-scale.

Specifically, this thesis argues that the performance of current activity recognition algorithms is highly conditional to the algorithm capability to learn efficiently with few examples and quickly adapt to and/or learn new activities and variations of the user behavior. Therefore, the activity recognition algorithm proposed in this thesis is built on the idea that simple action detectors can be pre-trained in a user-independent way and delivered ready to use. Thus, while the user might have given an application that initially doesnt fully work, the activity recognition system could learn as the user interacts with it.

In order to investigate and handle these requirements, this thesis has analyzed the following key classification problems:

1. The null-class problem
2. The subject adaptability problem
3. The computational efficiency problem

3.1.1 The null-class problem

The recognition of basic ambulatory and sedentary activities using wearable accelerometers has been well researched over a decade [109, 132]. Though useful, these approaches have investigated a limited set of activities collected in controlled settings and, for instance, they only apply to specific situations or settings.

As a result, a current problem in the field of activity recognition is how to effectively increase the number of activities to recognize. However, there are numerous different activities that could be recognized (e.g. It can go from 60 activities as listed in [202] to 605 as listed in [9], consequently, it can be soon realized that is not feasible to recognize all of them. This is not only due to the increased complexity of the recognition problem due to the multiple sources of variability (that go from hardware differences to differences in user behavior), but also to the fact that collecting data from all possible activities across hundred of users is a highly impractical as well as a very expensive task.

Thus, a more practical approach is to focus on the recognition of few activities of interest, but with the capability of discriminating the activities that do not need to be recognized or are irrelevant. Of course, given the imbalance of relevant versus irrelevant data, activities of interest can be easily confused with activities that have similar patterns. This problem is often referred in the literature as the null class problem, in where the irrelevant or other activities are called null class.

To generate an explicit model of the null class is very difficult if not impossible since it represents a theoretical infinite space of arbitrary data. However, it is possible to handle irrelevant activities by incorporating a rejection mechanism based on the likelihood of the classification result. Indeed, this problem has been investigated in previous research by [130, 20] and successfully used for activity and gesture spotting using Hidden Markov Models (HMMs).

In this chapter, the performance of the proposed algorithm for recognizing - not only few activities but also other activities that are part of a null class- will be investigated. In fact, the inclusion of this class considerably increases the complexity of the recognition task (as it will be shown through out the experiments presented in subsequent sections). However, it is an important case to solve when it is desired to increase the applicability and usefulness of the algorithm in real-world systems.

3.1.2 The user-adaptability problem

It is well known among the activity recognition community that a large difference in recognition accuracy is often reported when classifiers are evaluated in a subject-dependent versus a subject-independent manner.

Thus, in general, the goal of most activity recognition systems is to perform well and be validated in a subject-independent manner. This means that usually most systems are trained on a large number of users and then used by a new subject, which is unknown during the system development stage. This measure is usually a pessimistic systems performance measure. In contrast, a subject-dependent evaluation leads to a very optimistic system performance measurement.

Even though typically it is a highly recommended practice to test the system performance using both evaluation methods for having a good picture of the overall systems behavior, still many approaches proposed recently use subject-dependent evaluation methods(e.g., [99]. Indeed, this type of evaluation is problematic because the very optimistic performance results might not have much practical meaning for real-world applications in which new subjects and new information is often been observed.

Nevertheless, the subject-dependent evaluation is desirable when the explicit goal is the development of a tailored algorithm for a particular user. In such case, the focus is the measurement of the algorithm performance when few examples are available or when new information is introduced to the system. In other words, how the tailored algorithm learns new information in an incremental manner.

Since the method proposed in this thesis is trying to be robust in both cases, both evaluation techniques will be investigated.

3.1.3 The computational efficiency problem

An important difference between the algorithms for wearable systems versus other types of systems is that the computational resources of wearable systems are limited and, for instance, a low-cost computational solution is crucial.

The method described in this thesis proposes a solution in where the user can receive a new model within a short period of time to start using the system and, subsequently, adapts accordingly to new incoming data and activities.

To make this approach to work is necessary that the tailored algorithm can handle complex activity recognition tasks (e.g., recognition not only of the few basic activities which the model uses to start, but also other activities of interest to the user). However, in contrast to other approaches, the user should not be required to provide data from all the activities the system recognizes. The system should only require data for the activities of interest and, using the new labeled data previously unknown, only the relevant parts of the system should be retrained (only the activity to which new data is incorporated or the new activity of interest), the rest of the system should remain the same. The main goal allowing the system to preserve previous information already learned.

3.2 Contributions and Main Idea

This chapter presents the proposed method and its design criteria.

In sum, the first main contribution of the work presented in this thesis is to demonstrate that the proposed concept can realize the learning-upon-use paradigm, as well as, demonstrate that such concept is a valid approach to solve the main classification problems currently faced by real-world activity recognition systems.

The second contribution is the introduction of a novel algorithm based on the proposed concept.

Since the proposed algorithm is based on the Support Vector Machines (SVM) classifier using the all-pairs training method, the experiments section of this thesis presents first the optimization of the model parameters of this basic classifier. Subsequently, the optimized SVM classification model is compared to other widespread activity recognition approaches.

Moreover, the optimized SVM model is modified and extended to further increase the classification accuracy of the activity recognition task and incorporate the user-adaptability features proposed in this thesis. In particular, this is realized by introducing a meta-level learning algorithm based on a weighted voting method. This method is often used in the field of game theory and has numerous variations, thus, this work investigates several approaches to construct it.

The rest of this chapter presents the proposed algorithm, its data processing pipeline, and relevant related work related to other activity recognition approaches and the basic techniques presented in this work.

3.3 Algorithm Description

In recent years, the technique of Support Vector Machines (SVMs) has been increasingly investigated for activity recognition applications [106, 110, 24], due to its classification and estimation proficiency [28]. In fact, SVMs have been widely investigated and used in problems for isolated handwritten digit recognition [42], object recognition [179], speaker identification[160], and face detection in images [48]. In most of these cases, SVM generalization performance either matches or is significantly better than the one obtained by competing methods.

Although SVMs have good generalization performance, they can be very slow in training phase. This problem has been addressed in [28], [81] and [142], which introduced the widely used SVM light and SMO training algorithms. Currently, there are several modern implementations of these algorithms that have significantly increased the computational efficiency of SVMs during the training phase [44]. As a result, the significant improvements in low computational cost along with the SVMs excellent estimation capabilities reported in prior research make this technique to be an attractive approach for activity recognition systems.

Roughly speaking, SVM is a machine learning discriminative modeling approach that was originally designed for binary classification. Various methods exist where typically a multi-class classifier is constructed by combining several binary classifiers. Also, there are methods, which consider all the classes at once in where a single joint optimization problem is solved. The single joint optimization problem is computationally and mathematically harder to solve than the binary problem.

In this work the all-pairs binary classification method [92] is used because it is less computationally expensive than considering all classes at the same time. This method decomposes the multi-class classification problem in several binary tasks that require solving simple optimization sub-problems that consider only two classes at the time. Besides its low computational cost, this method provides the advantage of adding classes incrementally (on the fly). Indeed, this offers a significant advantage for activity recognition systems since activities could be added over time according to their relevance, as well as, their number could increase only limited by the mechanism to discriminate them at the top classification layer.

3.3.1 Basic formulation

Specifically, the all-pairs method constructs $\frac{k(k-1)}{2}$ binary classifiers, with k = number of classes where each classifier is trained on data from two classes. Each SVM is trained with all examples from the i^{th} class with positive labels, and the examples from the j^{th} class with negative labels. Thus, given training data $l(x_t, y_t)$, where $x_t \in \mathbb{R}^n$, $t = 1, 2, \ldots, l$ which is the data in both classes and$y_t \in \{i, j\}$ which is the class of x_t, the SVM solves the problem:

$$\min_{w^{ij},b^{ij},\xi^{ij}} \left(\frac{1}{2}(w^{ij})^\top w^{ij} + C\sum_t \xi_t^{ij}\right) \tag{3.1}$$

Where

$$(w^{ij})^\top \phi(x_t) + b^{ij} > 1 - \xi_t^{ij}, \text{if } y_t = i, \xi_t^{ij} \geq 0 \tag{3.2}$$

$$(w^{ij})^\top \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{if } y_t = j, \xi_t^{ij} \geq 0 \tag{3.3}$$

Where the training data x_t are mapped to a higher dimensional space by the function ϕ, C is the penalty parameter for the error, and ξ_t^{ij} is the slack variable that counts the classification errors made. In other words, the SVM will construct a separating hyperplane in that higher dimensional space, one that maximizes the separation ("*margin*") between the two classes. The margin hyperplanes can be written as the set of points $\phi(x_t)$ in the higher dimensional space satisfying equations 3.2 and 3.3, where w^{ij} is a normal vector perpendicular to the hyperplanes and b^{ij} determines the offset of the hyperplane to the origin along the normal vector w^{ij}. Then, minimizing $\frac{1}{2}(w^{ij})^\top w^{ij}$ means that we would like to maximize $\frac{2}{\|w^{ij}\|}$ which is the margin between the two groups of data. When data are not linear separable, there is a penalty term $C\sum_t \xi_t^{ij}$which can reduce the number of training errors. The basic concept behind SVM is to search for a balance between the regularization term $\frac{1}{2}(w^{ij})^\top w^{ij}$ and the training errors.

3.3.2 Typical meta-level layer

There are different methods for doing the multi-class testing after all binary classifiers are constructed. However, there is not a direct technique that can be used for solving a specific type of problem.

Typically, simple voting strategies (as suggested in [92]) are used. For example, one of the most common strategies involves the using of a sing function that predicts according with the class with the largest number of votes.

The sign function typically used is as follows:

$$sign((w^{ij})^\top \phi(x) + b^{ij}) \tag{3.4}$$

Basically the sign function compares the input vector of features to a decision boundary in the high dimensional space. Then, if the result is positive and we say that x_i belongs to the i^{th} class, the vote of the i^{th} class is added by one. Otherwise the j^{th} class is increased by one.

However, although this strategy has found to give reasonable results, it has the problems of when two or more classes have identical number of votes, the selection of the activity class often leads to

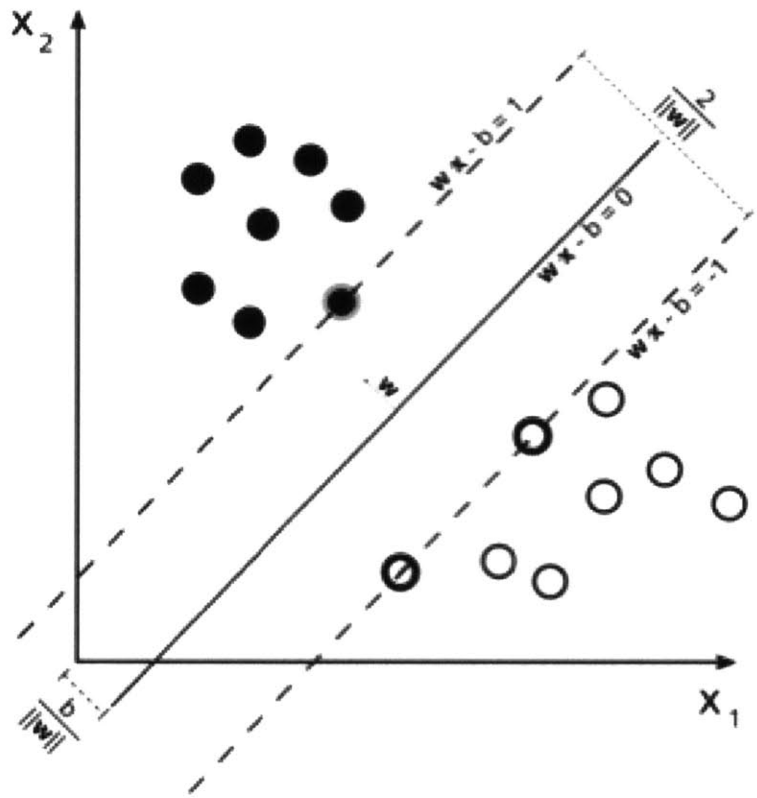


Figure 3-1: SVM Hyperplane

high number of errors. In such case, a common strategy is to select one of the classes with similar number of votes at random.

In summary:

1. $\frac{C(C-1)}{2}$ classifiers are constructed in where each one is trained on data from 2 classes.

2. To train data from the i^{th} and the j^{th} classes, we have to solve the following binary classification problem:

$$\min_{w^{ij},b^{ij},\xi^{ij}} \left(\frac{1}{2}(w^{ij})^\top w^{ij} + C\sum_t \xi_t^{ij} \right)$$

$$(w^{ij})^\top \phi(x_t) + b^{ij} > 1 - \xi_t^{ij}, \text{if } y_t = i, \xi_t^{ij} \geq 0$$

$$(w^{ij})^\top \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{if } y_t = j, \xi_t^{ij} \geq 0$$

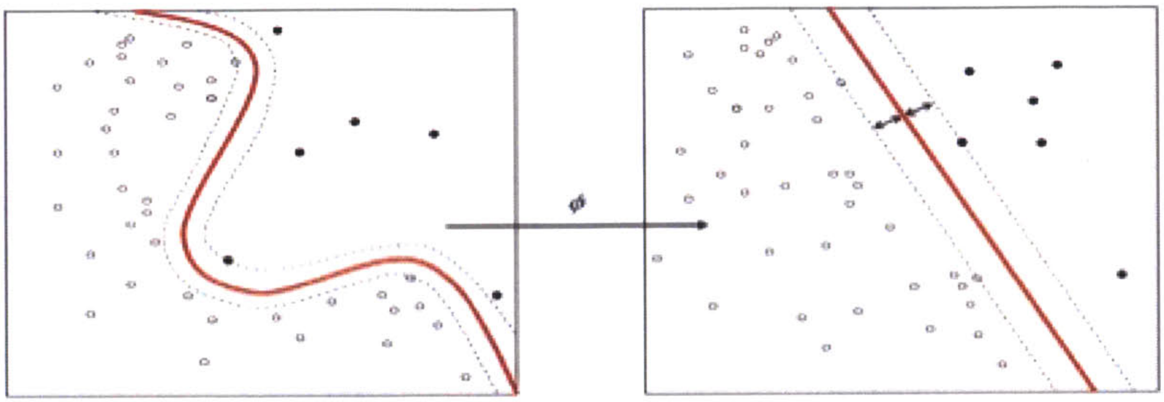


Figure 3-2: SVM Kernel Function

3. A voting strategy is used to predict x according with the class with the largest vote:

4. **IF** $sign((w^{ij})^{\top}\phi(x) + b^{ij}) > 0$ **THEN** say x is in the i^{th} class, then the vote of the i^{th} class is incremented by one. **ELSE** the j^{th} is increased by one.

If 2 classes have identical votes, one of the classes with similar number is selected at random.

3.3.3 Proposed Extension

Although the performance of the SVM algorithm often gives reasonable results, when the variability in how users perform the activities is high, the SVM algorithm alone or other approaches cannot perform beyond of a modest level of accuracy ($\sim 75\%$, see 3.4 for more details). Thus, to further increase the accuracy, the SVM approach has been extended to include a meta-level learning mechanism, which efficacy is supported by the experimental results obtained in the next chapter of this thesis.

The main model consist of a set of S SVMs binary classifiers, in which each classifier corresponds to a particular activity pair created from the original training data. Moreover, each set of classifiers corresponds to a single subject from the training dataset (subject dependent method). Thus, in the original SVM model, each SVM binary classifier in the set S has the same weight: $w_i = 1, i = 1, \ldots, S$ and no retraining of the weights is performed. Indeed, this is what is used as the baseline performance.

In the original SVM model, a new data example is classified with equal weighted binary classifiers, which give a classification prediction. Subsequently, such prediction is used to determine the final activity class based on the highest overall accumulated number of votes (in case of multiple classes having the number of votes a random selection is made).

In the proposed model, a new data example is classified by each binary SVM classifier and the resulting prediction is weighted according to the classifier associated retrained weight. Subsequently, the final class is selected based on the highest weight computed on the accumulated weight of each binary classifier and the updating overall weight coefficient.

In the approach proposed in this thesis, the weights w_i are retrained using a method based on an extension of the weighted majority of voting technique. The proposed novel extension is described below.

3.3.3.1 Extended formulation

Similar to other majority of voting methods (like the ones formulated in [174]), the proposed approach uses a set of new-labeled examples to train the weights of the binary classifiers, and uses weighted majority of voting to predict a new data instance.

It proposes a novel approach to deal with the question of what is the confidence of a binary classifiers decision when predicting a new unlabeled example. Thus, the main idea is that this confidence should depend on the prediction of all the other binary classifiers in the SVM model.

- Training Step

  Therefore, the result of training the weights with the new labeled example is a matrix W size SC (see figure 3-3 line 13). In where $w_{i,c}$ stands for the weight of the ith binary classifier when the majority of vote of all other classifiers is the class c. Indeed, this is defined as the weight in performance rate of the i^{th} binary classifier on this sub-set of examples (see figure 3-3 line 8-10).

- Prediction Step

  In the prediction step, the label of the xnew instance is determined with a binary classifier and with the resulting classification of all the other binary classifiers together (see figure 3-3 and line 18-19).

The resulting weight obtained this way is added to the accumulated weight for the label predicted by the corresponding binary classifier. This procedure is repeated for each of the individual binary classifiers.

Figure 3-3 Implementation pseudo-code and mathematical formulation.

Most existing majority of voting based methods only train an overall weight for each classifier, whereas this method trains each of the $w_{i,c}$ weights in an overall but dependent manner. This makes this approach more robust when a set of classifiers performs well on some cases but poorly on others. Also, it handles better the case of missing data when the classifier has no knowledge on a part of the problem space.

3.3.4 Complexity and Scalability

Since the proposed technique is based on SVMs, it inherits most their properties. One of the most striking properties is that both, the training and testing functions depend on the data only through

Require: **S** is the set of S different experts (classifiers): $s_i, i = 1, \ldots, S$
C is the set of C classes the classification task is composed of: $c_i, i = 1, \ldots, C$
N is the set of N new labeled samples: $\underline{n}_i = (\underline{x}_i, y_i), i = 1, \ldots, N$ ($\underline{x}_i$: feature vector, $y_i \in [1, \ldots, C]$)
New instance to classify: $\underline{x}_{new}$

procedure TRAINING_WEIGHT(**S**,**C**,**N**)
    for $i \leftarrow 1, S$ **do**
        for $j \leftarrow 1, N$ **do**
            Predict label of $\underline{x}_j$ with expert s_i: $\hat{y}_j$
            Predict label of $\underline{x}_j$ with the ensemble $\mathbf{S} \cap s_i$ (all experts but s_i), using majority voting: $\hat{\hat{y}}_j$
        end for
        for $c \leftarrow 1, C$ **do**
            $\mathbf{P}_c = \{\forall \underline{n} \in \mathbf{N} \mid \hat{\hat{y}} = c\}$
            % samples where the majority vote of the ensemble $\mathbf{S} \cap s_i$ *is the class* c
            $\mathbf{P}_{c\_good} = \{\forall \underline{n} \in \mathbf{P}_c \mid \hat{y} = y\}$
            % correctly predicted samples by s_i *from the set of* $\mathbf{P}_c$
            $w_{i,c} = |\mathbf{P}_{c\_good}| / |\mathbf{P}_c|$
            % the performance rate of the ith expert on $\mathbf{P}_c$
        end for
    end for
    W is the return matrix of weights, composed of elements $w_{i,c}$ where $i = 1, \ldots, S$ and $c = 1, \ldots, C$
end procedure
procedure PREDICTION(**S**,**C**,**W**,$\underline{x}_{new}$)
    $\mu_c = 0, c = 1, \ldots, C$ *% initialize prediction of* $\underline{x}_{new}$
    for $i \leftarrow 1, S$ **do**
        Predict label of $\underline{x}_{new}$ with expert s_i: class $\hat{c}$
        Predict label of $\underline{x}_{new}$ with the ensemble $\mathbf{S} \cap s_i$: class $\hat{\hat{c}}$
        $\mu_{\hat{c}} \leftarrow \mu_{\hat{c}} + w_{i,\hat{\hat{c}}}$
    end for
    The output class is $\arg\max_c \mu_c \quad c = 1, \ldots, C$
end procedure

Figure 3-3: Algorithm pseudo-code.

the kernel functions $K(x_i, x_j)$.

Even though the kernel K itself only corresponds to a dot product in the space of dimension dH , where dH can be very large or infinite (as it often happens in activity recognition given to the large feature vectors computed), the complexity of computing K can be fairly small (as it is shown in [39]).

For example, for kernels of the form of $K = (x_i, x_j)^p$, a dot product in H would require an order of $\binom{d_L+p-1}{p}$ operations. In contrast, the computation of $K(x_i, x_j)$ requires only $O(d_L)$ operations (in where d_L is the dimension of the data).

In fact, this property is the one that allows constructing simple hyperplanes in these very high dimensional spaces and still can be tractable with low computational cost. For this reason, this method can bypass both forms of curse of dimensionality. One caused by the proliferation of parameters causing intractable complexity and the other caused by the proliferation of parameters causing overfitting. These properties offer a significant advantage for activity recognition systems, which typically have to deal with these problems due to the high complexity generated due to the wide-ranging diversity in how the users perform activities.

3.3.5 Limitations

One of the biggest limitations of this approach is the choice of the kernel. Once the kernel is fixed, the individual classifiers have only one free-to-chose parameter (the error penalty). However, the kernel is a very big rug under which we could sweep parameters. Therefore, some work has been done on limiting kernels using prior knowledge [157, 28], but the best choice of a kernel for a given problem is still a research problem.

For instance, in the field of activity recognition and biomedicine, prior work [Gupta-MIT] has found that polynomial kernels (from order 2 to 4) give reasonable results when using basic SVMs classifiers (without any meta-learning layer). As shown in Chapter 6of this thesis, our analysis also supports these findings.

On the other hand, [110] has reported that SVMs with radial basis functions (RBF) provide also good classification results. However, their work doesnt systematically compare the RBF kernel against other kernel functions. In contrast, [Gupta-MIT] provides a more comprehensive and systematic analysis and has reported that polynomial kernels exhibit better performance in activity recognition accuracy. Despite these results, since current dataset are very limited and the hardware systems used to collect the data very diverse, the best choice of kernel for physical activity recognition is still an open research question.

A second limitation is speed and size, both in training and testing. While the speed problem in test phase is largely solved in [142, 44], this still requires two training passes for the multi-class problem.

As discussed in the beginning of this chapter, the all-pairs training method greatly alleviates this limitation by breaking the problem in a set of simple binary classifiers. However, it carries the complexity of the classification task to the meta-layer, solving this problem for very large datasets is still an open area of research.

3.4 Related Work on SVMs extensions

Although some work has been done on extending SVMs, finding the optimal design for multiclass SVMs classifiers that offers robust classification and low computational cost is an open area of research.

In particular, some of the most noticeable extensions for improving the performance of SVMs are the virtual support vector method, the reduced set method and, more recently, some meta-learning methods combining SVMs with generative probabilistic models such as HMMs (like the method described in [SVM-HMMs]) or combining SVMs with the AdaBoost algorithm [103].

3.4.0.1 The Virtual Support Vector Method

As described in [157], this method attempts to incorporate known invariances of the problem by first training a system and then creating new data by distorting the resulting support vectors, and finally training a new system on the distorted (and the undistorted) data. The idea is easy to implement and the method has been reported to give good results when the data has high variance or for incorporating invariances.

3.4.0.2 The Reduced Set Method

This method was introduced to address the speed of support vector machines in test phase, and also starts with a trained SVM. As defined in [161], the idea is to replace the sum corresponding to the support vectors weighting function $w_{original} = \sum \alpha_i y_i x_i$ with a similar sum ($w_{efficient}$). Thus, instead of support vectors, computed vectors -which are not elements of the training set- are used and, instead of the α_i, a different set of weights is computed. The resulting vector is still a vector in the hyperplane H, and the parameters are found by minimizing the Euclidean norm of the difference between the original vector $w_{original}$ and the approximated vector $w_{efficient}$. The same technique could be used for SVM regression to find much more efficient kernel function representations (in fact, this method is frequently used for data compression).

3.4.0.3 The SVM-AdaBoost Method

More recently, SVMs have been combined with the AdaBoost algorithm [103]. Specifically, the aim of the AdaBoost algorithm is to be used along with other types of learning algorithms to improve their performance. The output of the other learning algorithms called weak learners (in this case these learners are the SVMs) is combined into an overall weighted sum to determine the final output of the classifier. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can converge to a strong learner. This technique in general is sensitive to noisy data and outliers. However, in some problems can be less susceptible to overfitting than other learning algorithms.

While this algorithm will tend to suit some problem types better than others, in the field of activity recognition, AdaBoost has reported to perform better when used in combination with decision trees than when used with SVMs (with a linear kernel) as the weak learners [148].

However, it is important to highlight that these results need to be interpreted with some reserve. Since the performance of SVMs are highly dependent on the choice of their parameters (e.g., type of kernel, training method, and error penalty among others). For instance, the experiments reported in [148] use SVMs with linear kernel rather than other higher degree polynomial kernels that have been shown in this thesis (see section 6) and other prior work [AR-SVM-polynomial] to be more effective for solving the multi-class classification problem. Thus, despite the Ada-Boost is able to improve the performance of the weak classifiers, such performance improvement might not be sufficient if the weak classifiers can hardly perform slightly above chance even in the best case.

3.4.0.4 Other Approaches

Techniques based on an ensemble of classifiers like adaboost, boosting or random forest have become very popular within the machine learning community, in particular, when combined with decision trees (DT).

As these methods are constructed using DTs, they suffer from some of the same downsides than when using basic DTs (except for over-fitting). Of course, the fact you have a set of several DT classifiers instead of one increases the classification rate. However, the classification rate increases not because an individual algorithm performs better but because the ensemble is classifying using many models that are re-weighted according to new information. Indeed, one of the original purposes of this method is to solve the over-fitting problem typical of DTs.

In the other hand, ensemble based methods are often implemented over the cloud or a server in where the training or classification is mostly done in an offline manner. Thus, these methods start to be less optimal when online or near real-time classification with limited resources, many classes and high data dimensionalityis needed. In fact, these are precisely some of the most important requirements of wearable activity recognition systems.

In terms of computational requirements, techniques based on SVMs can be also computationally intensive depending on the kernel used. Nevertheless, if simple kernels are used as proposed in this work- the computational cost can be significantly reduced. In addition, SVMs framework offers important advantages like theoretical guarantees regarding overfitting, power of flexibility by using the kernels, and good performance even with data that is not completely separable (which is the case of many activity recognition problems).

In terms of bias variance tradeoff, it is important to highlight that activity recognition data collected in naturalistic settings usually has lots of variability, overlapping, and uncertainty generated by the lack of precise or clean annotations or hardware diversity. Thus, the choice of the best activity classification model should not only be based on overall recognition accuracy with few samples but also it should be based on the properties of the classification method. Some of such properties are related to how the model handles high-variance in data or overfitting, speed, ease of use, scalability, and other theoretical properties associated with the bias variance tradeoff - which is the tradeoff of choosing a model that both accurately captures the regularities of its training data, but also generalizes well to unseen data.

In general, ensemble techniques are very popular since they have been found to be very useful for a wide-range of applications in a wide range of domains. However, this doesnt mean that they are optimal for all types of problems.

Certainly, within the field of activity recognition is well known that sometimes better algorithm properties and good data, often beat better algorithms. Thus, the choice of algorithm properties and representative features are very important design considerations for having a good classification performance. In fact, this is the reason why collecting data in naturalistic settings to test new approaches is an important concern within the activity recognition community.

3.5 Chapter Contributions

This chapter introduced a novel general concept for realizing the learning-upon-use paradigm.

This concept uses a set of binary SVM classifiers as a general model, and retrains the weight of the classifiers using new-labeled data from an unknown user.

In the next chapter we will present experiments that try to show that this is a valid approach. Moreover, a novel algorithm is presented and compared to other existing activity recognition classification methods, showing that the recognition performance of the proposed method is significantly better than those. This is supported with a systematic evaluation comparing the general basic model with HMMs, decision trees, and other SVMs models using different kernels.

The main benefit of the introduced concept is that, instead of retraining all the activity classes contained in the general classification model, only the relevant class/classes to the new-labeled examples and their weights are retrained. As a result, this makes the re-training a new class or an existing class much less computationally intensive, since primarily the prediction of the new training example is required. Consequently, this approach can be used for mobile or wearable systems, inclusively for difficult classification tasks requiring more complex classifiers (see section 6). An analysis of the computational cost of the proposed approach (as shown in the experiment on computational cost section) shows its feasibility for online wearable applications. Furthermore, another advantage is that the proposed concept allows users only to collect data incrementally for a subset of recognized activities, a single activity, or a new activity without having to lose what the classifiers previously learned or without having to retrain the entire model every time new information is added. These features make the proposed approach more practical and usable for real-world applications.

Finally, most physical activity monitoring systems are usually trained in controlled laboratory conditions on user groups of healthy adults (in fact, usually grad or undergrad students). As a consequence, such systems often perform poorly when used by significantly different users (e.g. young children, elderly individuals, or people suffering from overweight).

Chapter 4

Datasets

The effort of collecting data for physical activity recognition in controlled or uncontrolled settings is significantly harder than collecting data in, for example, home environments. As result, existing datasets for wearable activity recognition are less comprehensive and harder to standardize because a robust wearable hardware and software system is required.

Indeed, some of the activities that are high aerobic can stress the sensor placement, can produce body blockage or, in general, stress the system setup (see section 2.5.1 for a detailed discussion).

Furthermore, as discussed in the background section 2.5.3, offline annotation using video recording (commonly used in home environments) observer is not feasible if outdoors and 24/7 activities are wished to be included in the data collected. As a result, online annotation by a human observer is commonly used for creating ground truth. Of course, such type of annotation has a very high cost in terms of human labor. There is why annotation based on experience sampling techniques is also used with the cost of lower label accuracy, lower amount of data labeled, and high user annoyance. Thus, given the high human cost of creating ground truth, there is a lack of commonly used, standard datasets and benchmarks for wearable activity recognition.

The following section introduces two datasets for physical activity recognition, in where one of them is publicly available (see [146] and [147]). The other dataset was collected earlier at our lab as described in [121].

The reason why these datasets were selected is they offer a wide-range of common activities performed by an adequate number of subjects wearing more than one sensor on common locations on the body (at least, wrist, hip, and ankle).

The rest of this section is organized as follows: first the selection of the sensor placement is described, then the method for creating ground truth for both datasets and, finally, the data collection details of both datasets.

Specifically, the MITEs dataset contains data from 52 activities and 20 subjects, wearing 7

MITES sensors (as described in [121]). Whereas the ICU-ETH dataset contains data from 18 activities and 9 subjects, wearing 3 IMUs and a HR-monitor (as described in [147]). For all datasets, the hardware setup, the data collection protocol, etc. are described in the corresponding publications and/or datasets referenced public dataset repository website.

4.1 Sensor Placement Selection

Previous work in e.g. [121] showed that in the trade-off between classification performance and number of sensors, using 3 sensor locations (hip, dominant wrist, dominant ankle/food) is the most effective. In systems for physical activity recognition the number of sensor placements should be kept at a minimum, for reasons of practicability and comfort since users of such systems usually wear them for many hours a day.

Moreover, prior work doing a detail analysis of all possible combinations of sensor positions shows that at least two [15, 38] to three [121] are necessary for the accurate classification of a wide range of activities. Indeed, the MIT MITES and UCI-ETH datasets were selected because they contain analogous data from the sensor positions of interest, they were collected in both controlled and semi-naturalistic settings, and they involve a relatively similar set of activities.

4.2 Ground Truth Annotation

Both datasets were annotated using an online annotation method. Essentially, this method involves a researcher accompanying the study participants. Then, the researcher marks the beginning and end of each of the different activities in an online manner during the data collection. The time stamped activity labels are stored on the data collection unit/units. The data format used in the published dataset can be found in [147] and [121] correspondently.

Since the beginning and end of each of the different performed activities are marked on set, time stamped activity labels are provided along with the raw sensory data. In both datasets, the collection of all raw sensory data and the labeling were implemented in separate applications and their synchronization was carried out offline.

4.3 The MIT MITES Dataset

4.3.1 Data Collection

This dataset is described in [121] and it was collected collaboratively by MIT Changing Places Laboratory and the Medicine Prevention Research Center in Stanford University.

The data collection procedure followed a protocol in which subjects were asked to perform a series of tasks. Each task involved the performance of a specific physical activity e.g. running 3 miles on the treadmill or washing specific windows. One researcher monitored the procedure and annotated the start and the end of the whole task associated with the activity. Data collected outside of start-end intervals was labeled as unknown. 24 subjects participated in two sessions. The subjects were 12 Females and 9 males with age, height and weight averages of 44 years, 171 cm and 71 kg (see details in [121]). A set of 6 and 20 activities were collected per each session. Each session involved a different set of activities.

Postures	Aerobic	Complex
sitting	running on treadmill	folding and stacking laundry
slouching	cycling on airdyne	vacuuming and moving chairs
sitting at desk	swinging arms	washing windows
standing with hands in pockets	push ups from knees	sweeping and mopping
lying down	deep knee bends	stretching
stretching and standing up	squats	swinging arms
	stepping on platform (1-stair)	turning and pivoting
	arm curls with hand weights	
	sit ups or crunches	
	picking up	

Table 4.1: Activities included in the MIT MITES Dataset. .

Notice that from the 24 participants for who the data was collected, only 20 turned out to have usable data. The data for one participant was incomplete, whereas, the data for the other participants was corrupted (sensors were misplaced, time stamps were corrupted, and/or information was missing within one accelerometer axis data stream).

In general, the data collection was designed to investigate the relationship between physical activity and energy expenditure as well as to validate sensors and algorithms used to measure biological signals and body movement. Specifically, heart rate, respiration, oxygen consumption and seven body joints acceleration signals were monitored using five types of sensors[1].

The data procedure was highly controlled given that several sources of error exist when measuring biological signals. Error associated to sensors measurements is introduced because biological signals are typically very noisy. In addition to this error, uncertainty is introduced by the algorithm classifying physical activities. Hence, the data collection was done in such a way that contribution from

[1]The sensors used are the Zephyr Bioharness monitor, Actigraph monitor, Omron monitor, Polar heart rate monitor (strap version), Oxicon respiration monitor and MITES sensors. All except the MITES sensors are commercially available health monitors.

known sources of error is minimal and constancy of sensor error variation across physical activities can be verified.

As reported in [121], the tasks assigned to participants were performed as natural as possible. For example, subjects were let to wash windows in the way how they will typically do it (e.g., using one hand versus both hands). Thus, activities were performed in a wide range of ways involving different speeds, joints moved and number of repetitions. As a result, the data set involves a diverse set of samples and a considerable number of classes, which makes the physical classification tasks to be not trivial.

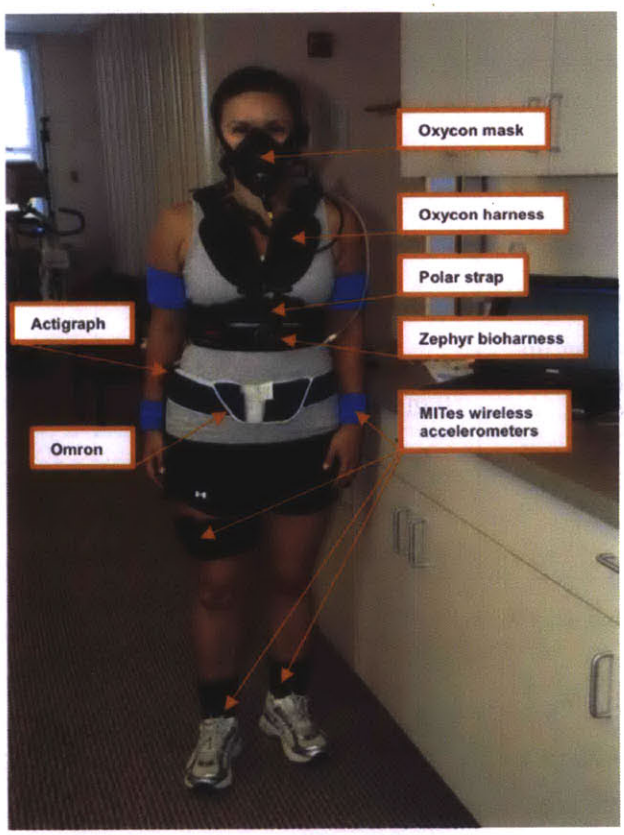


Figure 4-1: Participant wearing the sensors1 used in the experiment.

4.3.2 Apparatus

Data was collected using six types of sensors1 that were worn by the participants during the experiment session. Figure 4-1 shows a participant wearing the sensors. In this work, only the MITES data was used since the scope of this analysis is on activity recognition algorithms. In particular, the data consists of measurements coming from seven 3-axis acceleration sensors with a range of 2G, a resolution of 9-bit, dimension 3.2x2.5x0.6cm, weight 8.1g including battery, and 45-90Hz sampling rate. Each sensor was placed on different parts of the participants body (hip, wrists, ankles, upper dominant arm and upper dominant tight), was attached to the participants body by a flexible elastic strap.

4.4 The UCI-ETH (PAMAP2) Dataset

4.4.1 Data Collection

This UCI-ETH (PAMAP2) dataset is described in [147] and involves nine subjects, eight males and one female aged 27.22 3.31 years, and having a BMI of 25.11 2.62 kgm2. One subject was left-handed, all the others were right-handed. The protocol of performing indoor and outdoor activities for the data collection is described in [147].

A total of 18 different indoor and outdoor activities were included in the data collection protocol (see table 4.2). Participants wore 3 inertial measurement units (IMUs) and a heart rate (HR). A semi-naturalistic data collection was carried out in which participants were asked to follow the protocol, performing all established activities in the way most suitable for them.

Since a heart rate data was also included in the hardware setup of the data collection, one and two minute breaks were inserted in the data collection protocol after most activities. The goal was to ensure the measured heart rate was unaffected by previous activities. However, since such situation is idealistic and only usable for experimental purposes, the authors included in the data collection protocol cases in where activities were performed directly one after the other. See [147] for more details.

Posture	Aerobic	Complex
lying	ascending stairs	ironing
sitting	descending stairs	vacuum cleaning
standing	Normal walking	folding laundry
	Nordic walking	house cleaning
	Cycling	watching TV
	Running	computer work
	Rope jumping	car driving
		playing soccer

Table 4.2: UCI-ETH PAMPAP2 Dataset: Data Collection Protocol.

4.4.2 Apparatus

As described in [147], inertial data was recorded using 3 Colibri wireless IMUs from Trivisio [176]. The sensors are relatively lightweight (48 g including battery) and small (56 42 19 mm). Each IMU contains two 3-axis MEMS accelerometers (range: 16 g / 6 g, resolution: 13-bit), a 3-axis MEMS gyroscope (range: 1500/s, resolution: 13-bit), and a 3-axis magneto-resistive magnetic sensor (range: 400 T, resolution: 12-bit), all sampled at 100 Hz. To obtain heart rate information, a BM-CS5SR heart rate monitor from BM innovations GmbH [21] was used, providing heart rate values with approximately 9 Hz.

In this work, only the 3-axis accelerometer data is used from the IMU. Of the 3 IMUs, one was attached on the wrist of the dominant arm, one on the chest of the test subjects, and one sensor was on the ankle bodys dominant side.

A Viliv S5 UMPC (Intel Atom Z520 1.33GHz CPU and 1GB of RAM [186]) was used as data collection unit. A custom bag was made for the collection unit and 2 USB-dongles additionally required for the wireless data transfer one for the IMUs and one for the HR-monitor. The bag was carried by the subjects fixed on their belt and the device can run on batteries for up to 6 hours. Figure 4-2 shows the UCI-ETH (PAMAP2) dataset placement of IMUs (red dots) and the data collection unit (blue rectangle) as depicted in [147].

Figure 4-2: Sensor locations placed on the participant's body during the experiment.

4.5 Conclusion

In the field of physical activity monitoring there is a lack of a commonly used, standard dataset and established evaluation benchmarks. This chapter introduces the two datasets used in this thesis. One of them is publicly available at [178][166].

The MITEs dataset contains data from 20 activities and 20 subjects, wearing 7 MITES sensors (as described in [121]). The UCI-ETH dataset was recorded on 18 physical activities with 9 subjects, wearing 3 IMUs and a HR-monitor (as described in [147]). In the corresponding sections of this chapter the data collection procedure, apparatus, and participating subjects have been described for both datasets.

Apart from using these two datasets in this work, they have the advantage of being comparable to each other. In addition, one of them is publicly available. This has been of great advantage for the research community, which can make use of them. They contain a reasonable number of activities performed by a reasonable number of subjects, which make them usable to define challenging activity recognition problems, benchmark the evaluation tasks, and compare results. This entirely leads to the improvement of novel activity recognition approaches like the ones described in section 3.4.

Chapter 5

Materials and Methods

5.1 Data Processing

The data processing used in this work follows a classical approach used in activity recognition systems. Such approach involves processing steps that goes from raw sensor data to the prediction of an activity class. Figure 5-1 shows these steps.

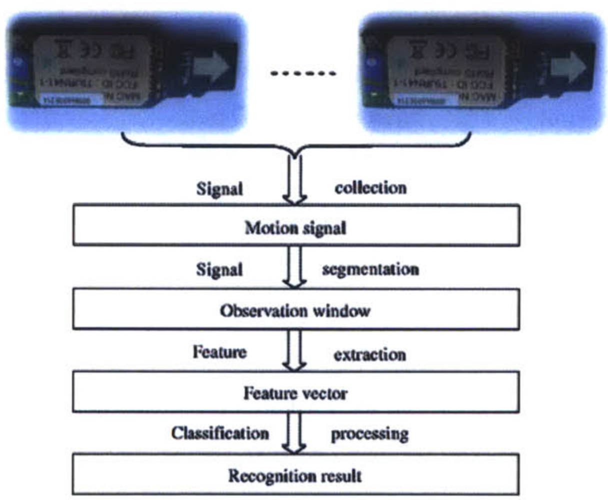


Figure 5-1: Typical data processing approach.

The first stage involves the common steps of pre-processing, segmentation and feature extraction. Then, the second step involves the classification step. In this second step various classifiers can be used (section experiments compares various classifiers and underlines their benefits and drawbacks).

This second step can be seen as the decision making module consisting of one or more than two steps (as described in the algorithm section). With one step, a classifier takes the entire feature set as an input, and outputs a class according to the specified classification problem. With two steps the classifier output can be taken as an input for a meta-learning layer and/or, subsequently, a filter

Time Domain	Frequency Domain
Mean	Energy
Total Mean	Entropy
Mean differences between axis	Dominant frequency
Range or maximum signal amplitude	Power ratio of certain frequency bands
Standard deviation	
Covariance	
Correlation between axes	
Absolute integral	
Peak of absolute data	

Table 5.1: Extrated features in the time and frequency domain.

that can smooth the output over time.

In this work, the estimation of the activity is regarded as the main classification task. However, the goal is not aiming for the best performance on the classification tasks, but to provide a baseline classification characterization based on a benchmark.

The results, challenges and discussion presented in this work aim to serve as motivation to improve existing methods with the introduction of new approaches or meta-learning layers.

For both datasets the following features were extracted:

Features are computed over sliding windows of 5s in length after interpolating and filtering the raw accelerometer signals by a band-pass and low pass filter as explained in [121]. For feature computation, this window is used with 50% overlapping between consecutive windows and 50% data loss tolerance. Thus, features are extracted from the sliding window signal, shifted by one second between consecutive windows. The signal features extracted from the acceleration sensor data are computed for each axis separately, as well as, for the 3 axes together.

When computing all features, the resulting feature vector contains 379 (54 features per sensor). In where, each axis measurements involve the value of mean, total mean, variance, covariance, range, signal absolute values, FFT (3 maximum frequencies) and energy.

5.1.1 Preprocessing

The datasets mentioned in chapter 4 provide time stamped raw sensor data and time stamped activity labels, which is synchronized in what is usually referred as the preprocessing step.

One of the problems faced in this step is the wireless data loss issue, which causes missing values. In this work, the common approach of linear interpolation is used for simplicity. Also, further preprocessing of the raw signal (like filtering) is used before extracting features. Finally, to avoid dealing with potential transient activities, 5 seconds from the beginning and the end, respectively, of each labeled activity is deleted.

5.1.2 Segmentation

Previous analyses carried out among many types of activities [121][38] show that there is not a unique window size that works for all activities. Lara et al. [99] and Munguia-Tapia [121] show that for obtaining at least two or three periods of all different periodic movements, a window length of about 3 to 5 seconds is practical choice, specially, to guarantee an efficient discrete Fourier transform (DFT) computation for the frequency domain features (see table features). As a result, a window of 5.12s was selected. Since the sampling rate of the raw accelerometer signals collected in the datasets are 45Hz (MITES) and 100Hz (UCI-ETH), over a window of 5.12s, represent a window size of 256 and 512 samples respectively.

5.1.3 Feature Extraction

The most commonly used features reported in literature are: mean, median, standard deviation, peak acceleration, DFT, and energy. Other features have been used too such as the absolute integral, the correlation between each pair of axes, power ratio of the frequency bands 02.75 Hz and 05 Hz, peak frequency of the power spectral density (PSD), Spectral entropy of the normalized PSD, among others. Table 5.1 shows a list of all computed features in both time and frequency domain and table 5.2 shows a list of alternative features and their usefulness for detecting specific activities.

Features	Usefulness
Absolute integral	Successfully used to estimate the metabolic equivalent in e.g. [T-Reiss-118].
Correlation between each pair of axes	Useful for differentiating among activities that involve translation in just one or multiple dimensions, e.g. walking, running vs. ascending stairs [131].
Peak frequency of the power spectral density (PSD)	Useful for the detection of cyclic activities in e.g. [42].
Spectral entropy of the normalized PSD	Useful feature for differentiating between locomotion activities (walking, running) and cycling [42].
Power ratio	Frequency bands 02.75 Hz and 05 Hz. proved to be useful in [134].

Table 5.2: Feature usefulness.

See appendix [56] for a mathematical definition of the extracted features the following notation is used. [56]provides a review of extracted features from accelerometer sensor data.

5.2 Performance Evaluation Metrics

As mentioned before, the algorithm performance score is the single dependent variable used in this study. In particular, this score is not based exclusively on the algorithm accuracy but also on its precision.

Table 5.3 shows a confusion matrix describing some of the performance evaluation metrics commonly used for learning algorithms. In this analysis, algorithm performance is computed via the harmonic mean between the algorithm accuracy and its precision.

The reason why the score considers a combination of these metrics is they have some disadvantages when used independently. For example, if we consider the case 1 shown in Table 5.4, we can see that both classifiers obtain 60% accuracy, but they exhibit very different behaviors. The left has weak positive recognition rate but strong negative recognition rate, whereas the right has strong positive recognition rate but weak negative recognition rate.

On the other hand, if we consider the case 2 shown in Table 5.5, we can see that in this case both algorithms obtain the same precision and recall values 66.7% and 40% but they exhibit different behaviors. They have the same positive recognition rate but extremely different negative recognition rate (strong on for the left and almost nothing for the right). In contrast, accuracy doesnt have any problem detecting this.

Confusion matrix		
	True class	
Hypothesized class	Pos	Neg
Yes	TP	FP
No	FN	TN
	$T = TP + FN$	$F = FP + TN$

Common evaluation metrics
Accuracy $= \frac{TP+TN}{P+N}$
Precision $= \frac{TP}{TP+FP}$
Recall/TP Rate $= \frac{TP}{P}$
FP Rate $= \frac{FP}{N}$

Table 5.3: Common performance evaluation metrics

5.3 Performance Evaluation Method

The classifiers are trained for each activity following K-fold cross-validation approach [89]. For the subject-independent case, k represents the number of subjects. Whereas for the subject-dependent case, k represents the number of partitions in a subject's data per activity.

	True class		True class	
Hypothesized class	Pos	Neg	Pos	Neg
Yes	200	100	400	300
No	300	400	100	200
	$P = 500$	$N = 500$	$P = 500$	$N = 500$

Table 5.4: Algorithms performance case 1

	True class		True class	
Hypothesized class	Pos	Neg	Pos	Neg
Yes	200	100	200	100
No	300	400	300	0
	$P = 500$	$N = 500$	$P = 500$	$N = 500$

Table 5.5: Algorithms performance case 2

For the subject-independent case, the data is divided according to the number of subjects, in which each partition corresponds to the data coming for a particular subject. Then, a single subject is retained as the validation data for testing the model, and the remaining subjects are used as training data. This operation is repeated as many times as the number of subjects, each time using different subject to test classification model accuracy.

For the subject-dependent case, the data from a particular subject is randomly divided in 10 (k=10) partitions, each of them containing 1/k from the total number of observations for each activity class. Then, a single partition is retained as the validation data for testing the model, and the remaining partitions (k 1) are used as training data. This operation was repeated k times, each time using different training sets to obtain the K-fold cross-validation accuracy.

5.4 A Priori Power

Even though the physical activity data was already collected prior the design of the evaluation experiments, I computed the sample size that would be recommended if the sample size needed to be planned as well as, the power expected with the actual sample size. Specifically, in the field of physical activity, the power calculation is somewhat problematic because few evaluations report their variances in accuracy. As result, the minimum range between treatments means was specified only based on one set of experiments previously performed in our laboratory [121]. Thus, $\Delta = 1.25\sigma$ was selected. For this value, the a priori power having 20 subjects is 0.90 for $\alpha = 0.05$ and 0.80 for $\alpha = 0.01$.

Chapter 6

Experiments

In this section different aspects of the suggested approach are analyzed. In a systematic evaluation of the proposed general concept and the introduced algorithm, results are presented and discussed.

6.1 Experiment 1: Choice of SVM model parameters

Experiments were carried out testing different number of feature vectors {379, 64, 42, 18} and four SVM configurations (three polynomial kernels and one radial basis function (RBF)). Specifically, the experiments used the following parameter values C=1 with KKT tolerance 0.001. The polynomial kernels considered degrees 1,2 and 3. The RBF kernel used . For this test, we used a subset of 20 activities from the MITES dataset.

Table Cross-validation accuracy for each activity class across different SVM models. shows the specific activity clases contained in the dataset and their cross-validation accuracy across different SVM models. For instance, (64F, K2) means the SVM was trained using 64 features per observation vector and a polynomial kernel of degree 2. Given the space restriction for this paper, table 2 shows only the best SVM models for a specific number of feature vectors. To see additional results for different configurations (see appendix 9.1).

Interestingly, when using 379 feature vectors, the linear kernel of degree one had a reasonable performance (80%) whereas the RBF kernel had a poor one (68%). Additionally, the SVM with RFB kernel took very long time to converge (approximately $\sim 2hrs+$) versus few minutes for polynomials. Part of the problem is the large number of features. Indeed, the number of vector features (in this case 379) is sufficiently large that it is not necessary to map the data to a higher dimensional space using a kernel such as RBF. This also can explain why, when using the largest set of feature vectors, a simple first-degree polynomial kernel works well. Because computing 379 features (in particular FFTs) is a very expensive computationally intensive task, the second experiment consisted in training a multi-class SVM classifier with a reduced number of features. Features were reduced in 3 ways.

Class	Cross-Validation Accuracy		
	64F,K2	**42F,K2**	**18F,K3**
sitting slouching	1	1	1
sitting at desk	1	1	1
standing with hands in pockets	1	1	0.972
running on treadmill	1	0.996	0.999
cycling on airdyne	1	1	1
folding stack laundry	0.956	0.903	0.795
vacuuming move chairs	0.971	0.892	0.815
washing windows	0.928	0.823	0.666
sweeping and mopping	0.969	0.933	0.597
stretching	0.88	0.766	0.684
swinging arms	0.84	0.639	0
push ups from knees	1	1	1
deep knee bends squats	0.922	0.702	0.291
stepping on platform	0.99	0.935	0.563
arm curls hand weights	0.938	0.654	0
sit ups or crunches	1	0.972	1
lying down	1	1	0.988
picking up	1	1	1
stretching standing	0.951	0.941	0.939
swinging arm turning	0.573	0.455	0.366
Overall Accuracy	**97.26%**	**93.55%**	**83.99%**
Ave. Training time	97.17s	1948s	194.63s

Table 6.1: Cross-validation accuracy for each activity class across different SVM models.

One considered 64 features corresponding to the basic statistics on 3 sensors, the second one considered a subset of 42 features, and the third one considered a subset of 18 features. The sensor positions were selected according the sensor placement described in section 4.1.

The results show that the reducing the features to 64 can give still an accuracy of at least 93% (which is similar when using 379 features and a polynomial kernel degree 1). In contrast, the accuracy drops to 83% when using 18 features. It is important to notice that the classes of folding clothes, vacuuming, washing laundry, stretching and swinging arms have a low accuracy across the different multi-class SVM classifiers. The next section focuses on the analysis of these activities using other widely used models such as Hidden Markov Models and Decision Trees.

6.2 Experiment 2: Comparing SVM versus Hidden Markov Models

6.2.1 Choice of HMM Model Parameters

The HMM is a generative probabilistic model consisting of a hidden variable and observable variable at each time step. In this case, the hidden variable corresponds to a fraction of the movement that composes the activity and the observable variable is the vector of sensor readings.

There are 2 dependency assumptions that define this model:

(1) the hidden variable at time t depends only on the previous hidden variable at time $t-1$ (Markov assumption).

(2) the observable variable x_t at time t, depends on the hidden variable at time t.

These assumptions allow representing an HMM model using 3 parameters associated with the following distributions:

- the distribution over the initial states $\pi = p(y_1)$
- the transition distribution $A = p(y_t \mid y_{t-1})$
- the emission distribution $E = p(x_t \mid y_t)$ that state y_t would generate observation x_t .

Given a training set of the observation sequences $\underline{\mathrm{x}} = x_1 \ldots x_T$ corresponding to each activity, the model parameters $\lambda(\pi, A, E)$ need to be learned. Learning the parameters of these distributions corresponds to maximizing the joint probability $p(\underline{\mathrm{x}}, \underline{\mathrm{y}}) = p(x_1 \ldots x_T, y_1 \ldots y_T)$ of the paired observations and state sequences in the training data. This joint distribution can be factorized in terms of the model parameters:

$$p(x, y) = \pi(y_1) E(x_1 \mid y_1) \prod_{t=2}^{T} A(y_t \mid y_{t-1})$$

Because we are dealing with continuous data, the emission probability E is defined as a distribution of a mixture of Gaussians:

$$E(x_t \mid y_t) = \sum_{q_t=1}^{M} g(q_t \mid y_t) N(x_t; \mu_{q_t}, \Sigma_{q_t})$$

for $k = 1 \dots M$. Where M is the number of possible gaussian components and $g(q_t, y_t)$ specifies a distribution over the M possible mixture components, which depends on the underlying state y_t. Once a mixture component q_t is chosen, the emission x_t is generated from a Gaussian distribution with mean μ_{q_t} and Σ_{q_t}.

Given that we have partially observed data (we dont know the state sequences), the parameters that maximize the joint probability $p(\underline{x}, \underline{y})$ are estimated using the EM-Algorithm [1].

First the parameters are initialized to some value. Subsequently, the parameters π, A, and $g(q_t \mid y_t)$ are maximized recursively until converging into a local maximum of the likelihood function. As result, the estimates updates are:

$$\begin{aligned}
\pi_y^{c+1} &= \frac{\sum_{i=1}^{n} count[i, y_l = y; \Theta^c]}{N} \\
A_{y,y'}^{c+1} &= \frac{\sum_{i=1}^{n} count[i, y \to y'; \Theta^c]}{\sum_{i=1}^{n} \sum_{y=1}^{S} count[i, y \to y'; \Theta^c]} \\
g^{c+1}(q \mid y) &= \frac{\sum_{T=1}^{T} \gamma^c[t, y, q]}{\sum_{q=1}^{M} \sum_{t=1}^{T} \gamma^c[t, y, q]} \\
\mu_q^{c+1} &= \frac{\sum_{t=1}^{T} \sum_{y=1}^{S} \gamma^c[t, y, q] x_t}{\sum_{t=1}^{T} \sum_{y=1}^{S} \gamma^c[t, y, q]} \\
\Sigma_q^{c+1} &= \frac{\sum_{t=1}^{T} \sum_{y=1}^{S} \gamma^c[t, y, q](x_t - \mu_q)(x_t - \mu_q)^\top}{\sum_{t=1}^{T} \sum_{y=1}^{S} \gamma^c[t, y, q]}
\end{aligned}$$

Where $\gamma^c[t, y, q] = p(y_t = y, Q_t = q \mid x_1 \dots x_T; \Theta^c)$ is the posterior probability given the current parameters, T is the number of observations in the sequence, S is the number of hidden states, M is the number of mixture components and N the total number of sequences.

6.2.2 Results

To narrow down the HMM analysis to the most interesting cases, I decided to consider activity classes that obtained low, medium and high cross-validation accuracy with the multi-class SVM classifier. For instance, I am considering all the hard cases such as folding clothes, washing windows, vacuuming and stretching.

An additional difficulty when analyzing this data set with HMMs is that there is not information

Class	Model S,M	CV-Accuracy
running on treadmill	3,3	100%
cycling on airdyne	3,3	99.89%
folding stack laundry	4,4	82.72%
vacuuming / moving chairs	4,4	84.20%
washing windows	5,5	85.56%
sweeping / mopping	3,3	91.98%
stretching	2,2	97.16%
swinging arms	3,3	98.77%

Table 6.2: HMM Models (64 features)

about the activity segmentation. In this case, activities were segmented from the start to the end of the whole task rather than each activity instance by itself. Therefore, in this problem, there is not a well-defined dictionary of movements or phonemes like in speech. In other words, I dont know the best number of hidden states for modeling the movement transitions of each activity. Thus, it is necessary to determine the best model for each activity.

For selecting each activity model, an HMM was trained using different numbers of hidden states $S = 1 \ldots 6$ and mixture components $M = 1 \ldots 6$. The training was done using labeled data for each activity in nine data sets as (training set D_k) and the testing was done using one data set (validation set v_k). This operation was done 10 times, each time using a different data set as v_k.

The HMM toolbox for Matlab developed by K. Murphy [122] was used to train and test the different models. The log likelihood of each model was calculated for each observation sequence corresponding to the activity in the validation data set. The model with the lowest average log-likelihood was selected.

Table 6.2 shows the model parameter values that obtained the maximum log-likelihood for each activity, where is S is the number of hidden states and M is the number of gaussian components. After selecting the best activity model according with table 6.3, an HMM was trained using different number of features for each activity class ($C_1 \ldots C_8$), where C_i indicates the learned HMM model parameter values for each activity class, and 8 is the total number of classes considered.

Again, using K-fold cross-validation with new partitions of randomly selected examples, a train set D_k with 9 data partitions and a test set v_k with one partition were formed. The log-likelihood of each activity model was calculated for each observation sequence in the test set v_k. Each observation sequence $X^l = \{x_1^l \ldots x_T^l\}$ (with $T = 5$) in the validation data set $v_k = \{X^l\}_{l=1}^{L}$ was classified according with the activity model C_i that gave the maximum log likelihood. The final classification was obtained as follows:

Class	CV-A 64F	CV-A 42F	CV-A 18F
running on treadmill	97.78	94.44	83.21
cycling on airdyne	98.89	95.43	86.54
folding stack laundry	82.72	74.90	63.80
vacuuming / moving chairs	84.20	74.28	54.44
washing windows	85.56	77.44	57.78
sweeping / mopping	91.98	79.26	62.22
stretching	87.16	77.42	61.98
swinging arms	98.77	89.85	67.65
total accuracy:	90.88%	82.88%	67.20%

Table 6.3: HMM Cross-Validation Accuracy (%)

$\hat{\Theta}(X^l) = \underset{c}{\text{argmax}} L(\lambda_{c_i})$. The classifier $\hat{\Theta}$ takes values in the class set $\Theta = \{1, \ldots, C\}$

This process was repeated 10 times using K-fold cross-validation. The average cross-validation accuracy per class and per number of features considered was computed as shown in Table 6.3. The results show that HMMs has a significantively lower performance than SVMs, specially, when the number of features is decreased to 18.

Activities that are considered to be hard cases for the SVMs classifiers had also low recognition accuracy for HMMs, which was even lower). One reason why these activities are hard to recognize is linked to the fact that they involve more complex and less structured movements than the rest of the cases in the data set.

Given that these activities involved more complex movements, one initial assumption was that HMM could model such complexity (specially in the temporal domain) more effectively. However, this turned out not to be the case given that the HMM were not able to deal with the high varibility characteristic of most activity data collected using wearable accelerometers.

The results presented so far were obtained by modeling each activity with an HMM and using observation sequences which had the same length for all activities. However, the classification results could be improved by modeling each activity with a different sequence length. In some cases, a short sequence duration might not be enough for capturing the patterns or periodicity of a particular activity as discussed in section 5.1.2, especially when the activity involves different movements and body positions that are relatively separated in time. Moreover, results could be improved if prior information of activity units or activity phonemes were available.

Finally, it is important to notice that the SVMs classifiers were not significantly affected by having activities with relatively few examples. However, this was not the case for HMMs, which depend on the counts of how many times states and states transitions are seen in the sequences.

The results to this condition (commonly encountered in the real-world) makes SVM to offer a big advantage over HMMs when implemented for real-world applications.

6.2.3 Discussion of Results

This first experiment compared two types of classifier (the optimized multi-class SVMs versus HMMs) when recognizing a set of 20 physical activities using the MIT MITES dataset described in section 4.3. The results show that a multi-class SVM classifier performs significantly better than a classifier based on HMMs for all types of activities. Using a multi-class SVMs classifier, an overall accuracy of 97%, 93% and 83.99% was obtained when using 3 sensors and 64, 42, 18 features correspondely. These results imply that it is possible to reliable recognize common activities using sensors embedded in commercial electronic devices placed on the body or commonly used mobile phones, watches or shoes. Finally, the all-pairs binary decomposition training method for SVMs facilitates the realization of such scenario in where activities classes can be added incrementaly (on the fly). As discussed in chapter 3 sections 3.1, when adding a new class, this method keeps what it has been learned (by retaining the models for the already trained classifiers) and only trains the binary classifiers involving the new class. This approach has the benefit of being sognificately less computationally expensive than training a model that contains all the classes at the same time.

6.3 Experiment 3: Comparing SVM versus Decision Trees

6.3.1 Choice of Decision Trees Parameters

Decision tree (DT) classifiers such as the C4.5 algorithm [154] are among the most used to recognize activities from wearable accelerometers [17, 114, 134]. This is because of the following reasons:

1. They learn classification rules that are believed to be easier to interpret than the ones learned by other methods such as neural networks (although for real-world problems, these rules can be quite complex and not trivial to interpret).

2. They incorporate information gain feature selection during the learning process that identifies the most discriminatory features to use.

3. They perform fast classifications making them suitable for real-time implementation. A disadvantage of decision trees is that they tend to overfit the data if they are trained on small datasets and may not combine probabilistic evidence as well as other methods. Furthermore, decision trees are static classifiers that do not incorporate any temporal transition information of the modeled activities unless it is encoded in the features used

Specifically, a decision tree model consists of a set of rules for dividing a large heterogeneous population into smaller, more homogeneous groups with respect to a particular target variable [75]. A decision tree may be grown automatically by applying any one of several decision tree algorithms to a model data set. In this analysis the decision tree C4.5 algorithm is used to classify an activity feature vector by assigning it to the most likely class.

The C4.5 algorithm is Quinlans extension of his own ID3 algorithm for generating decision trees [154]. The C4.5 algorithm recursively visits each decision node, selecting the optimal split, until no further splits are possible. In general, steps in C4.5 algorithm to build decision tree are:

1. Choose a feature for root node.
2. Create branch for each value of that feature.
3. Split activity feature vectors according to branches.
4. Repeat process for each branch until all samples in the branch belong to the same activity class.

In the activity classification task, the selection of the root feature is based on the highest information gain [154] (difference in entropy) that results from choosing a specific feature for splitting the training data. At each node of the tree, C4.5 chooses one feature of the data that most effectively splits the set of activity samples into subsets augmenting one class or the other.

6.3.2 Types of Algorithms Used

The independent variables are type of algorithm and type of activity intensity. The algorithm variable is composed of the three algorithms introduced before: decision trees (DT), SVM with a linear kernel and SVM (SVMK1) with a quadratic kernel (SVMK2).

The activity intensity variable is given by the four types of intensity associated with the activities contained in the data set. The intensity levels are postures (no intensity), light, moderate and vigorous intensities.

6.3.3 Experiment Design

To select the experimental design, two aspects were taken into account. First, past research shows physical activities have a high variation across subjects. Second, besides algorithm performance, the effects of activity intensity on performance are also of interest. As a result, two experimental designs were conducted. The first aims to compare the overall algorithms performance based on a single-factor repeated measures model. The second aims to compare two-factors (algorithm performances and activity intensity) and is based on a two-factor repeated measures model with repeated measures in one factor.

- Single-factor repeated measures design: this design involves one independent variable (type of algorithm) as the repeated measure and a single dependent variable, which is the algorithm performance. Algorithms were tested in a subject-dependent manner for all the 20 subjects.

- Two-factor design with repeated measures in one factor design: this is a design that involves two independent variables: type of algorithm and type of activity intensity, and a single dependent variable (algorithm performance). In this approach, subjects were randomly assigned to each activity intensity level (5 subjects per level). Similarly to the single-factor design, the algorithms were trained in subject-dependent manner. However, this time only considering data from activities contained within the same intensity level.

6.3.4 Experiment Results on Performance

The algorithm differences on performance were evaluated using a single-factor repeated measures design. When the model assumptions were tested, it was found the normality assumption was not accomplished (p=0.019) whereas the homogeneity of variances was met with a probability of 0.36 via the Brown-Forsythe test. Figure 6-1 shows the normality plot of error terms.

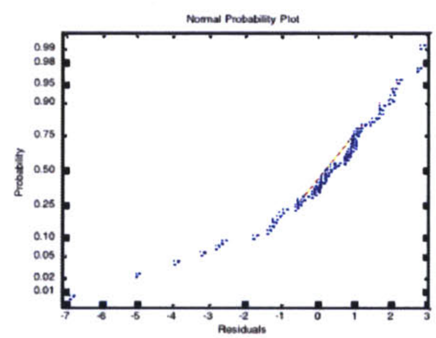


Figure 6-1: Normality plot for the error terms for the algorithm factor

With basis on the assumption the ANOVA model is robust to deviations in normality, an ANOVA test based on the single-factor repeated measures model was performed. The test found significant main effects in the algorithms performances $F(0.95, 2, 38) = 3.25$, $p < 0.001$. According to Bonferroni pairwise comparisons, the overall performances were significantly different for all pairings of the algorithms ($p < 0.001$). The overall performance score tended to be higher for the SVM algorithm with quadratic kernel, followed by the SVM algorithm with linear kernel. Finally, overall performance tended to be lower for the decision trees algorithm. Figure 6-2 shows a comparison of the algorithms performances.

As was explained before, the ANOVA repeated measures model was applied because the homogeneity of variances assumption was not violated and the ANOVA model is robust to departures from normality. However, if a strict criterion is applied, the violation of the normality assumption suggests the need of using a non-parametric method. To verify if the parametric results still

Algorithms compared	Significance	
	ANOVA	Friedman
DT-SVMK2	$p < 0.0001$	$p < 0.0001$
DT-SVMK1	$p < 0.0001$	$p = 0.033$
SVMK1-SVMK2	$p < 0.0001$	$p = 0.002$

Table 6.4: Probability values for pairwise comparisons across algorithms

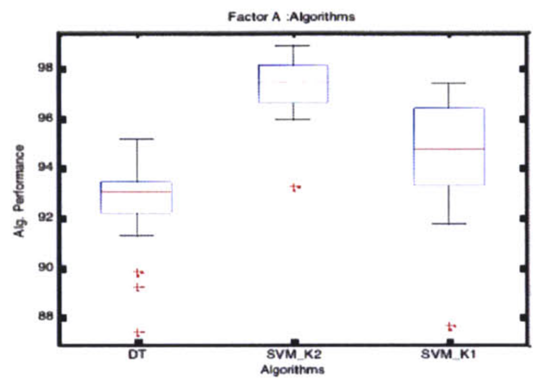


Figure 6-2: Median of the performance score across algorithms

hold when the data is analyzed with a non-parametric model, the Friedman test (which is the non-parametric equivalent to the parametric repeated measures ANOVA test) was employed. As a result, the Friedman test gave similar conclusions than the ANOVA test confirming that the differences in the algorithms performance are significant. Figure 6-3 shows the mean rank intervals and the probabilities (p-values) for the algorithms according to the Friedman pairwise test. Although the ANOVA leads to the same than Friedman test, the ANOVA p-values are considerable smaller ($p < 0.0001$). Table 6.4 shows the probabilities for both tests.

6.3.5 Experiment Results on Activity Type and Intensity

Whereas experiment 1 addressed the differences of overall algorithms performance, a second experiment was designed to investigate more specific algorithm differences according to the activity type and its intensity.

In the context of energy expenditure, activity intensity is important because, once the intensity is known (e.g. light, moderate or vigorous), the activity can be mapped to its rate of energy expenditure in METs according to the Compendium of Physical Activities [9]. In particular, moderate and vigorous intensities are relevant for the medical community because, when medical interventions are designed, it is important to know how much a subject spend in sedentary or light activities and which is the exercise base line of a target population.

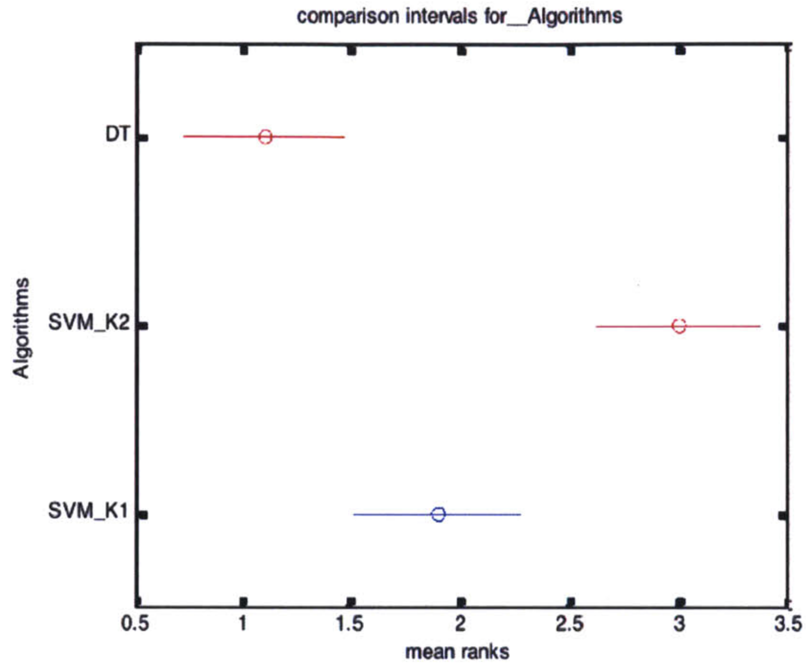


Figure 6-3: Comparison of algorithms' mean rank intervals

After assigning the activities intensities according to the Compendium of Physical Activities [9]. The 2-factor repeated measures assumptions were tested. The Kolgomorov-Smirnov test indicated the data was not normal ($p < 0.001$). The Brown-Forsythe test indicated the variances between subjects for the activity intensity factor were not homogeneous ($p < 0.001$). Figure 6-4 shows the results of the tests.

Because the model assumptions were not satisfied, the Friedman test (which is the non-parametric equivalent to the parametric repeated measures ANOVA test) was used instead. Specifically, the Friedman test (FR) was adjusted according to Oron et. al. [13] for accounting the two factors of interest given that the classical version of the test only considers one repeated factor across subjects. Therefore, the adjusted test calculates separate Friedman statistics for the algorithm effects (repeated factor) at each level of activity intensity.

The FR test indicated there were significant main effects for activity intensity. Specifically, at the level of 95% confidence, Friedman pairwise comparisons indicated algorithms performed significantly better when recognizing postures than activities with light and vigorous intensities. It also indicated that moderate intensity activities tend to be recognized better than those with light intensity. However, no significant difference was found between light and vigorous intensities for which all the algorithms exhibited the lowest performance. Figure 6-5 shows the mean rank intervals and probabilities for the pairwise comparisons.

On the other hand, the adjusted FR test indicated that were algorithm effects for all levels

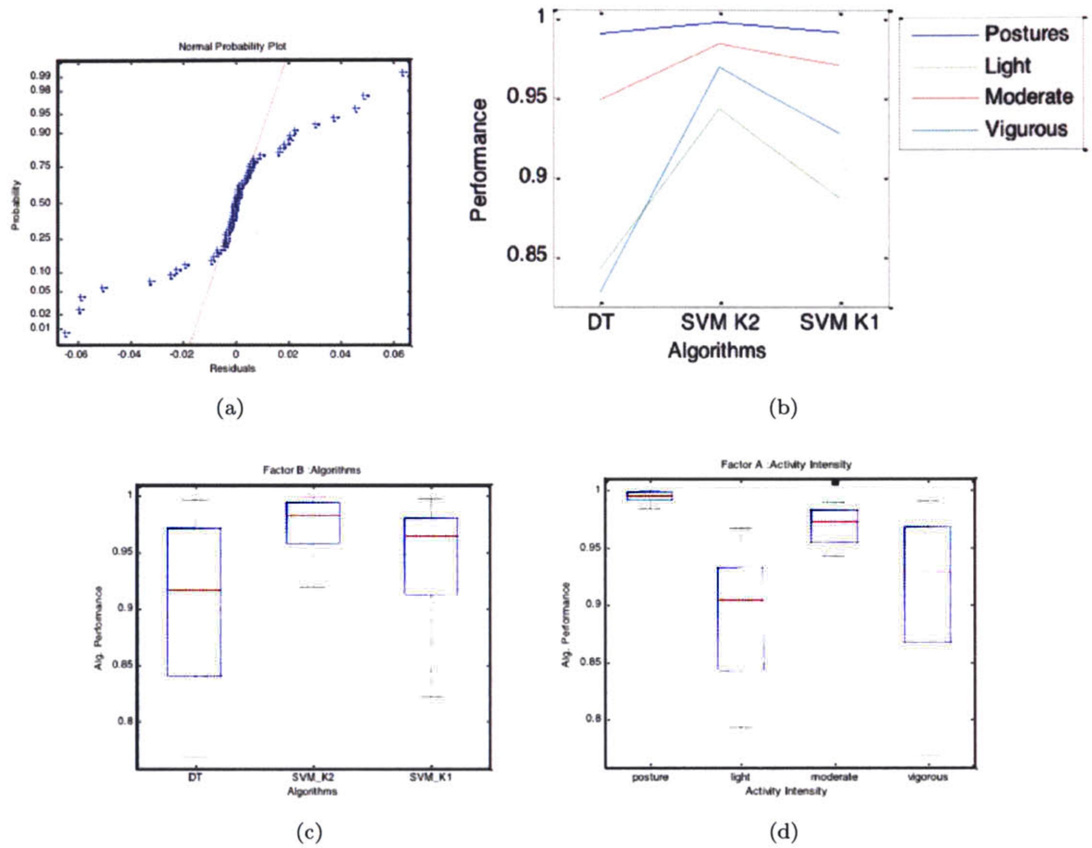


Figure 6-4: (a) normality plot of the error terms, (b) interaction plot, (c) median of the performance score across activity intensities, and (d) median of the performance score across algorithms.

of activity intensity. As can be seen in Table 6.6, Friedman pairwise comparisons indicated the SVMK2 algorithm performed significantly better than the decision trees algorithm at all levels of activity intensity. As in experiment 1, the algorithms performed in the same order being the SVMK2 algorithm the best and the DT algorithm the worse. However, in this experiment, the SVMK1 was caught in the middle overlapping the other two algorithms given that no significant differences were found. Figure 6-6 shows the results of the test.

6.3.6 Discussion of Results

This set of experiments investigate whether the observed differences in algorithms performances can be attributed to significant differences in their characteristics or they were obtained by chance as well as, whether or not activity intensity has any significant effect on the algorithms recognition.

First, results from the first experiment on algorithm performance show the differences in performance between the three algorithms are significant when all activities are taken into account. Specifically, the results indicated the SVMK2 algorithm had the highest performance whereas the

Intensities Compared		Significance
Posture	Light	0
Posture	Moderate	0.051
Posture	Vigorous	0.0001
Light	Moderate	0.0015
Light	Vigorous	0.2379
Moderate	Vigorous	0.1124

Table 6.5: Significance of activity intensity pairwise comparisons

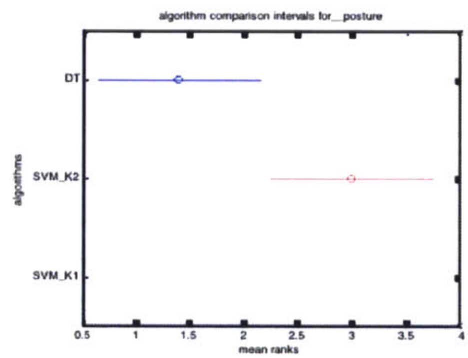
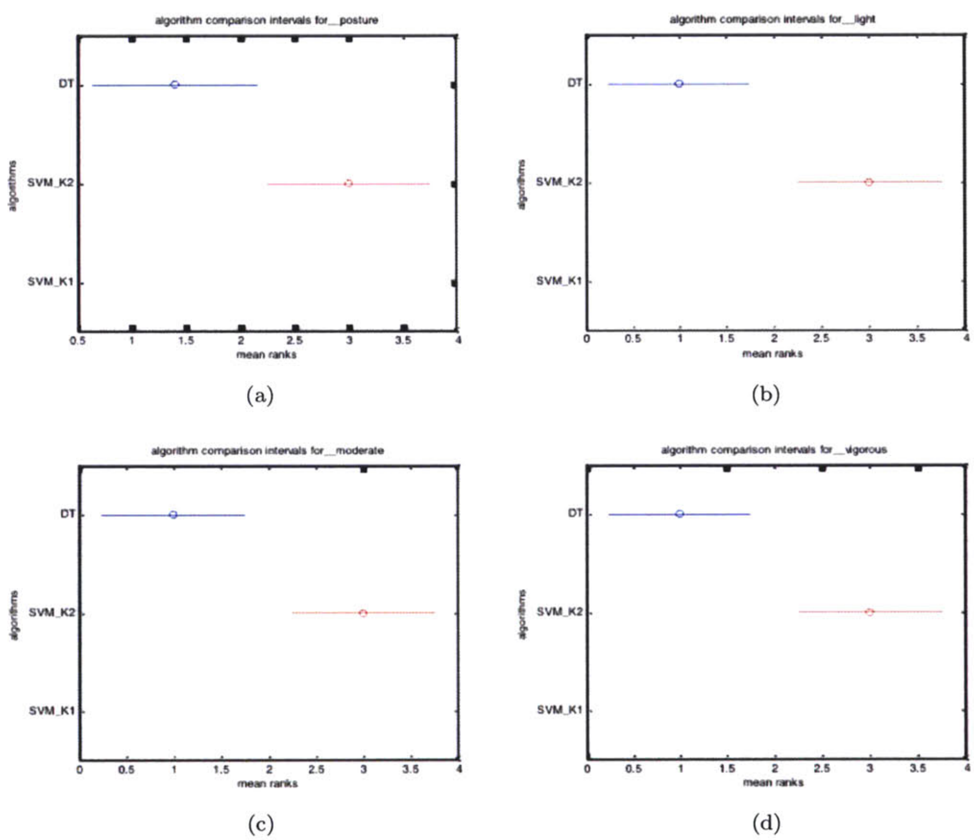


Figure 6-5: Comparison of activity intensity mean, rank intervals

DT algorithm had the worse. These results were consistent when using the ANOVA (parametric) and Friedman (non-parametric) tests. The only difference was the significance obtained by the ANOVA test was considerably smaller ($p < 0.0001$) for the comparisons between DT-SVMK1 and SVMK2-SVMK1 which for the Friedman test were $p = 0.033$ and $p = 0.002$ respectively. This result reflects the fact that the non-parametric test is less sensitive than the parametric test. However, because the data set is not normal, I was especially careful interpreting this result. In this case, agreement in the conclusions of both tests as well as, the inspection of the confidence intervals confirms the significance of the differences.

Second, results from experiment on activity type and intensity show that only algorithms SVMK2 and DT have significant differences in performance when analyzed at each level of activity intensity ($p = 0.03$ for the posture category, $p = 0.0054$ for the other categories). In this modality, no significant differences between the SVMK1 algorithm and the other two algorithms were found ($p = 0.75$ and 0.06 for postures, $p = 0.2286$ for the rest). Pairwise comparisons showed the SVMK1 overlaps the lower bound of the SVMK2 interval and the upper bound of the DT interval. These results suggests that partitioning the according to intensity levels makes the data within each intensity category more homogeneous shrinking the differences in performance between the algorithms.



(a) (b) (c) (d)

Figure 6-6: Comparison of algorithms mean rank intervals for each activity intensity. (a) Postures, (b) light, (c) moderate, (d) vigorous.

Algorithms Compared		Significance for activity intensity			
		Pos	Light	Moderate	Vigorous
DT	SVMK2	0.0325	0.0054	0.0054	0.0054
DT	SVMK1	0.759	0.2286	0.2286	0.2286
SVMK2	SVMK1	0.0689	0.2286	0.2286	0.2286

Table 6.6: Significance of algorithms pairwise comparisons at different activity intensity levels

Third, the experiment on activity type and intensity also shows that the algorithms recognize significantly better postures and moderate intensity activities than light and vigorous activities. Specifically the significance of postures compared to light and vigorous intensities is $p < 0.0001$ whereas, the significance between moderated and light intensities is $p = 0.0015$. A reason why postures might be recognized well is there is not much variation to model in the accelerometer signals generated as these activities are static in nature. In contrast, light activities are harder to recognize because they involve high degree of variation. These activities such as folding and stacking laundry, stretching, swinging arms and turning and pivoting involve multiple postures and movements that change depending on how the subject interacts with the environment. On the other hand, vigorous activities (which only involve resistance exercises) exhibit low recognition rate even though they involve less complex body movements than light activities. This might be due to the fact that accelerometers are not good at detecting changes in work load. Hence, it seems the algorithms have difficulty in recognizing activities with similar motion patterns but different levels of work load or activities that are highly variable within an individual. Consequently, further investigation of the effects of additional features, which could account for time or energy of the signal (such as Fast Fourier Transform) needs to be done.

With regards the experiment design, a repeated measures model was used based on the fact that physical activities have high variability between subjects. In addition, there were not enough subjects to carry out a fully crossed 3×4 design (algorithms and intensities as factors) for which at least 24 subjects are necessary for a minimum of 2 replicates per treatment combination.

On the other hand, because the data was collected prior designing the evaluation, there are some aspects to be improved. For example, sessions collected at different days should include the same activities in order to have better information about within subject variability. In this data set, even though more than one session was collected; each session contains a set of different activities. Hence, inferences about activity variability within the same subject are restricted to comparing activity samples extracted from one continuous data stream (e.g. 6 min of the activity collected at a specific time is partitioned in 10-folds).

Finally, it turned out that a good algorithm performance metric is the combination of accuracy and precision measures rather than accuracy or precision/recall alone. Because the number of physical activity samples was sufficiently large, the data was partitioned in two sets with equal amount of instances (one data set was used only for training and the other only for testing). A cross-validation re-sampling method was used to train each algorithm. However, if the data set were smaller and comparisons between the k-fold partitions need to be made, the decision about applying a cross-validation method needs to be treated with care since the training sets obtained from the re-sampled partitions are not independent. In such case, other re-sampling techniques (e.g., bootstrapping or randomization) could be used. Lastly, it is important to verify the appropriateness of a parametric test. As was seen in this analysis, data from algorithms performance often violate the normality and homogeneity of variances assumptions, which are necessary for applying a statistical test based on a parametric method. For this analysis, a non-parametric method based on the Friedman statistical test was employed.

6.4 Conclusion

Significant differences in algorithms performance were found when all the set of activities was taken into account. These differences indicated that the support vector machine with quadratic kernel algorithm (SVMK2) had the highest performance whereas the decision trees algorithm (DT) had the worse. In contrast, when the algorithms were analyzed at each activity intensity level, only significant differences between the SVMK2 and DT algorithms were found at all intensity levels. In this modality, no significant differences between the support vector machine with linear kernel algorithm (SVMK1) and the other two algorithms were found. Pairwise comparisons showed that the SVMK1 performance was overlapping the lower bound of the SVMK2 interval and the upper of the DT interval. In general, algorithms performed significantly better detecting postures and moderate intensity activities than activities corresponding to light and vigorous intensities. This finding suggests that additional features encapsulating energy of the signal and/or time need to be explored for improving the recognition of activities with light and vigorous intensities.

Chapter 7

Design Guidelines for Activity Recognition Systems

7.1 Disclaimer

Via lessons-learned and design guidelines, this chapter aims to illustrate important qualitative design aspects, challenges and trade-offs faced when designing a scalable activity recognition system intended to be used for long-term in natural settings. In fact, this qualitative analysis considers that - depending on the available resources and the specific recognition problem- many of these challenges are not only exclusive of activity recognition systems but also they are common across many wearable systems intended to recognize human behavior or physiological signals for long periods of time in natural settings.

The critical qualitative analysis is based on the work developed under a NIH-Initiative called GEI exposure biology program grant number #5U01HL091737.

In particular, this grant involved the development of a system called wockets and the systematic data collection of physical activities using such system. This project was made as a collaborative effort between MIT Changing Places group and the Preventive Medicine Research Center at Stanford Medical School as described in [76].

These two chapters only involve my perspective and analysis of the design process of the wockets system from the angle of my participation in the project.

In this collaborative effort, it was demonstrated the feasibility not only of embedding a full multi-sensor robust wireless activity recognition system in a small factor, but more importantly, of sensing in real-time fine-granularity activity data for long periods of time. As one would imagine, we tackled numerous hard problems around miniaturization, wireless power, wireless communications, ergonomics, usability, adherence, algorithm optimization, data storage, and phone OS programing, among others.

The result was an example of a great collaborative project, which still was left with many open questions. Indeed, as it will be illustrated and discussed in the following chapters, such open questions are commonly encountered within the wearable computing community and, in some cases, still unresolved. This is why other researchers must continue to invest in research, open source hardware and software, publicly available datasets, and pushing the boundaries of science and technology in an effort to advance the knowledge of the entire activity recognition field to help to create systems that can effectively be integrated in real-world applications. This open approach to working with others and across teams has consistently produced outsized results for many research projects and for the entire scientific community at large.

As described in [76] and , besides myself, the project involved a great collaborative effort across institutions and people with various levels of expertise, whose names are listed as follow:

Stephen S. Intille, Fahd Albinali, Jason Nawyn, Benjamin Kuris, Pilar Botana, William L. Haskell, Mary Rosenberger, Denise Shine, and Abey Mukkananchery.

The specific contribution of each person is listed inline with the material presented. In particular, I was involved and collaborated in various design aspects of the hardware, the software, the gathering of the system usability requirements, and technical support of short and long-term of the data collection.

All the material is referred to online project webpage [195] in where more detailed information can be found.

7.2 System Design Guidelines

7.2.1 Motivation

Wearable devices and the mobile phone sensing platform are central components in implementing an activity recognition system that can collect and understand behavior in natural settings.

In particular, from some years now the phone operating system platforms (Windows Mobile, Android, iOS) have allowed researchers and developers to do more things with embedded devices than it was ever seen before. Nevertheless, there is still a great gap between having code for accessing on the phone sensors or collecting high quality raw data from external sensors, to having a deployable system or an end-user application that fully utilizes this access. Some of the components that make such difference are the ones related to the specialized algorithms that can enable a more efficient data collection that maximize the battery life, delay tolerant communication with the sensors or with the back-end server, the ability to remotely configure data collection settings, and so on. Thus, for example, the data needs to be collected locally when the server is not available and needs to be uploaded in the background whenever users connect to the network or when users are not actively using their phones.

At the time when we started our project, there was no available software system or open source platform that supported these needs or that we could adjust or modify to our needs. As a result, we needed to develop a system (Hardware and Software) in our own.

This problem was common in the field of activity recognition and many of our colleagues in other research centers were facing similar problems. They had also the need to develop their own solutions from scratch since nothing with the appropriate requirements was available to use. For example, some of them involve only hardware like preocupine [94] and MSP [50]whereas others run only over the phone like Funf [123], Purplerobot [33], SystemSens [55], and Paco [53].

Many of these systems, as other smaller research initiatives, were first developed for specific data collections experiments and are mostly tailored for the needs of a specific project. However, despite their differences, they also share many common requirements, for example, all of them needed to deal with accessing sensors, storing data on the mobile device, and transferring it to a server. All of them could take advantage of the users phone data connectivity by having the ability to do remote configuration and debugging. All share the need for dealing with battery processing, storage, and bandwidth limitations.

As a result, this leads to duplicated efforts that go towards achieving the same functionality. Thus, based on the experience gained through our own deployments, we aim that releasing the system as an open source and free framework could help future research initiatives in the field.

Since the beginning of the project, it was decided we wanted to share our system and our design experience of the system with the community. This effort turned the project into an open source hardware and software framework for researchers and developers called Wockets.

At the first design iteration, the Wockets system was used in experimental and semi-naturalistic settings. In particular, one of the goals of this first iteration was to validate and assess the functionality and quality of the data provided by the system. To do so, in collaboration with Stanford Medical School (see team section at [71]), physical activity data was collected among 33 individuals. The data was collected for each person using a protocol that required performing a variety of physical activities (see next chapter for more details). During this design iteration we (the Wockets team) redesigned the sensing system and its settings, optimized its performance, fixed data quality issues, and added features to the firmware and software (see details in section 2.5.1).

The second iteration of system took place when the Wockets were deployed in the real world in the SWAP study [71]. In this iteration, the system was fully able to collect raw and summary data and upload this data to a back-end server automatically according to the phone usage conditions. For example, data will be uploaded at the end of the day, when the phone was plugged to the power supply and the user was not using the device.

The first version of the system was developed using the windows mobile platform (OS version

6.x). Then, the system was upgraded to Windows mobile OS version 8.x, which is the one described in this thesis. The first version (OS version 6.x) was the one used to carry out the data collections described in Section 7.3 and corresponds the system described in Intille et al [76]. Whereas the upgraded version (OS version 8.x) is the one used in the projects described in [chang, 2014] and corresponds to the framework described in this thesis.

The basic system (Windows Mobile OS version 6.x) was also upgraded to support Android OS based devices. The Android OS version was further developed and maintained at Northeastern University by Prof. Stephen Intilles research team (see project web pages [195] [194]).

7.2.2 Concept

The core concept is to provide an open source system that enables the collection of fine-granularity body movement data, analyzes it in real-time, and uploads it to a back-end server to keep researchers informed continuously about what is happening during the data collection. In this manner, the researchers could analyze the data continuously, learn about the users behavior in a regular basis and use this knowledge to identify problems in the study or design/test a particular intervention (e.g. behavior change intervention or ask to the user relevant questions about particular events).

Together with the Wockets system simple apps could be deployed based on the Ecological Momentary Assessment (EMA) or experience sampling [16] methods to survey the users about their activities, provide summary information about what they were doing, and gain insight about other events of interest. Besides that, researchers, self-trackers, and independent enthusiast could use the system to collect and explore information related to the users behavior and its correlation with his environment, his social fabric, and/or his health.

The project originally aimed to foster a community around it and the ideas of code sharing and reuse and low-cost manufacturing were paramount to its design. The main idea behind such design was that new development efforts would go towards extending a common platform flexible enough to adapt to a wide range of experiments and able to be easily reconfigured.

This chapter describes the system used in the data collections carried out by the team at Stanford Medical School (see team section at [71]) and briefly described in section 7.3 of this thesis and in [76]. The system is described in terms of its hardware and software architecture. Specifically, I use the term Wockets alone to refer to the sensing units, Wockets framework to refer to the software components, and Wockets system to refer to the system as a whole.

7.2.3 System Interaction

The interaction with the Wockets system is designed around a multi-user model. In the higher level, no technical or software development skills are needed. While in at the lower level, developers can add new functionality and enhance the system, or easily package their activity recognition or other inference algorithms for use by others users. Figure 7-1 Illustration of the multiple levels of interaction with the Wockets system, which are detailed in the following subsections.

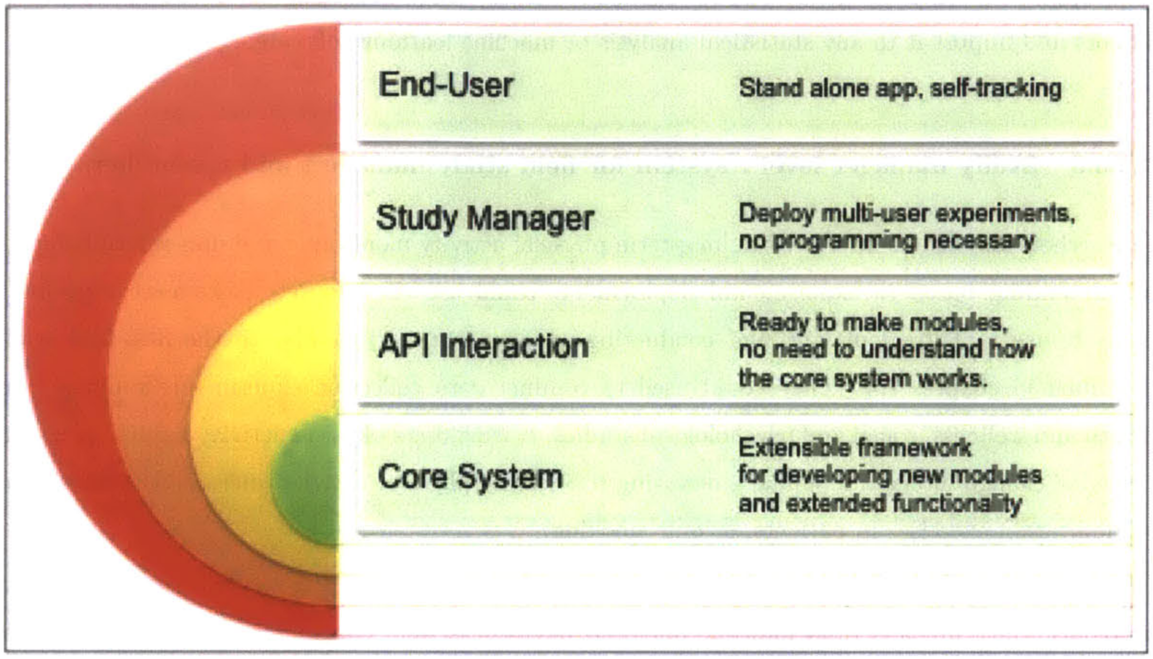


Figure 7-1: System levels of interaction.

7.2.3.1 Levels of Interaction

7.2.3.2 End-user level: stand-alone app for end users

At the highest level a stand-alone mobile application provides an easy-to-use tool for the collection of physical activity data from a single phone and/or up to 7 Wockets simultaneously connected.

This application allows any user to collect and explore information related with body movement and users behavior that can be inferred using EMA along with other sensors on the phone. It is well aligned with the type of end-user data collection used on self-tracking and the quantified-self movement [143].

On the other hand, students and researchers could use the application to easily conduct automatic sensing and data gathering of body movement using Wockets and a common mobile phone. For instance, it could be used by researchers to gather body movement data to train and test a gesture recognizer or study characteristic movements associated to motor related disorders (e.g., like the ones observed in people suffering Autism disorder or Parkinsons disease).

The Wockets Data Collection Tool was the first application release for this level of user interaction and it is available and free to download at the Wockets developers website [194].

It enables manual configuration of all data collection features via an XML configuration file and exports the data in multiple ways (e.g., via csv values, xml, and binary files). A simple pc-based script for visualization of the data signals was created as a tool to verify the quality of the data and export it as comma separated values (.cvs) text files to combine with data collected with other sensors and import it to any statistical analysis or machine learning software.

7.2.3.3 Study manager level : system for field study managers and researchers

Researchers and others interested in long-term physical activity monitoring and non-verbal behavior understanding based in experimental deployment, could use the Wockets system as a pre-built ready-to-use tool for deploying and conducting experiments, very similar to the Stanford study described in chapter 7.3. They could used to conduct data collection experiments ranging from health and wellness, social and psychological studies, tracing users physical activity habits, or testing behavior change interventions like motivating to increase physical activity, increase the number of rehabilitation exercises in patients, or quit smoking.

The Wockets Data Collection Tool supports a networked mode where a server is set up to receive automatic data uploads from multiple deployed devices, as well as a remote configuration of the phone-side data collection settings via the server.

Researchers can install the application on the devices that they want to collect the data from remotely, send the wearable sensors (wockets) to the participants, set up the data-collection back-end server, remotely configure the desired sensors (wearable ones and on-the-phone) and data collection behavior, and set up optional components like user surveys or specific interventions. Full instructions and tutorials can be found on the Wockets Wiki Website [195], in particular the getting started guide at [191].

7.2.3.4 Developers level : ready-to-use or extensible framework

Ready-made building blocks for mobile application development

The wockets framework is available as a pre-compiled C# library file (which we call the kernel), which exposes an application interface (API) for a set of functionalities and building blocks that can be easily integrated with 3rd-party applications.

This allows app developers to build mobile applications that leverage the capabilities and services of the Wockets system through its third-party developer API, without the need to go into the Wockets internals. This allows developers to save time and focus on the crucial parts of their app while the Wockets kernel takes care of things like logging and uploading the app data and sensor data. Detailed API instructions, documentation, and tutorials, can be found at the wockets developer website [193].

Extensible framework for developing new building blocks, tools, and algorithms

Core-developers can go into the specifics and use features that are outside the scope of third-party API. They could leverage the Wockets framework modular architecture and implemented features and to focus on the new functionalities that they care about; for instance, a new feature extraction or activity recognition algorithm. They could also contribute to new capabilities of the framework, add new sensors (external or built-on-the-phone), improve the performance of existing functionality, or export the Wockets framework to other mobile platforms like iOS or Android. Details of the Wockets framework source code and example code can be found at the Wockets developers website at [194].

7.2.4 Hardware

As described in [76], inertial data is captured using a MMA7331LCR1 accelerometer from Freescale Semiconductors. The sensor is very lightweight (8g including battery) and small (3.2x2.5x0.6cm). The sensing unit contains one 3-axis accelerometer (range 4g, 9g, 10-bit per axis resolution). It can sample data at different sampling rates, but for the application of physical activity recognition, the sensing unit is configured to sample at 40Hz. An A TMEGA1284P processor and an RN-41 Bluetooth module class 1 was used, with a voltage operation of 2.2v- 3.6v and lithium polymer battery of 3.7V 240mAh with PCB. More details about the hardware specifications can be found in [192]. Figure 7-2 shows the evolution of the sensor hardware design.

7.2.5 Software

7.2.5.1 Software Architecture on Mobile Device

The data collection and activity recognition modules are the basic software components used by the Wockets framework. Each module is a contained unit responsible for collecting a specific signal or type of information. The term module is used rather than sensors or accelerometers, as modules could encompass not only accelerometers, but also other sensors like GPS or other types of information not traditionally considered as collected by sensors, like file system scans or the logging of user behavior inside the phone. Using the modular probe architecture, it is easier to add new modules to the system, or swap existing modules with an improved version.

All modules support a common set of behaviors (like module registration, type of data collected, etc.), and each defines a set of configuration parameters that control it and format its output. Modules can be configured locally on the device or remotely through the back-end server. Also, the modules can be connected to one another. In this way, the one modules output can be the input of the other, the latter acting as a client of the first. This architecture allows the creation of a hierarchy or even a network of modules.

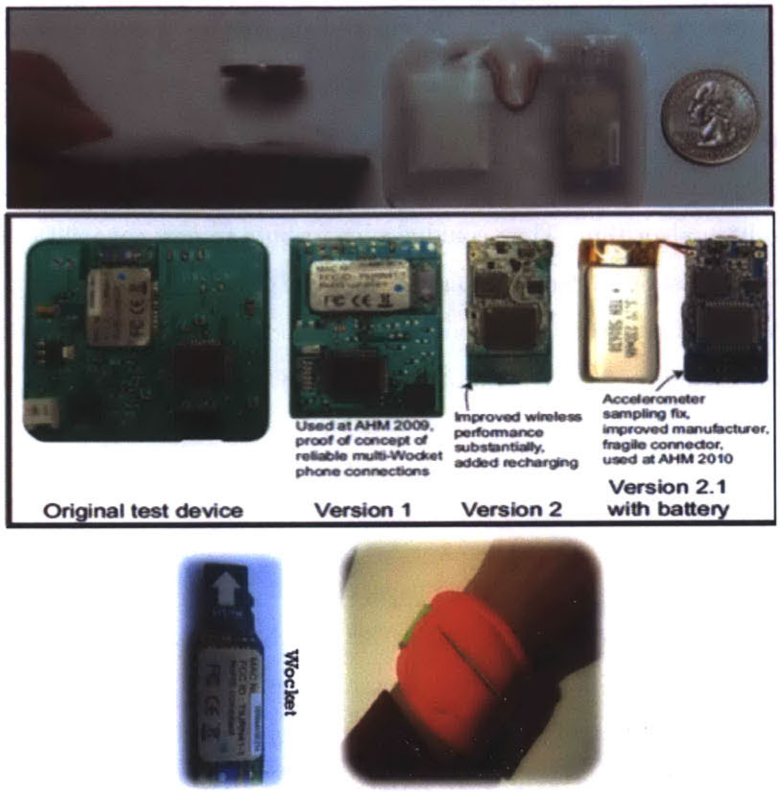


Figure 7-2: Wockets sensing units.

The system includes a set of built-in modules as well as a modular architecture, which allows the addition of new modules by third-party developers.

For example, one might want to develop an app and would like to ask users what they are spending time on, annotate an event, or record other information that might help researchers to contextualize the data collected.

One option to do that would be to develop totally independent code modules for logging the data captured by the UI, saving it to disk, and then sending it back to the developers servers, while dealing with a whole range of issues as added battery consumption, memory storage, protecting users privacy, and so on. Alternatively, with the Wockets system, the developers might write a simple Wockets module that logs all needed information, and leverages the Wockets existing modules and tested software for dealing with these issues.

In general, there are two strategies for implementing modules: a fixed-sampling strategy and an opportunistic-sampling strategy.

A fixed-sampling strategy explicitly requests data to be collected at a certain time and might need to turn on phone resources (e.g., turns WIFI, GPS or internal accelerometer if they were off), which might add battery expenditure. Modules supporting this strategy usually include a definition for periodic execution, with a maximum interval between executions.

An opportunistic-sampling strategy registers the modules as a listener for collecting different messages sent as broadcast within the phone operating system service framework. These could be built-in messages like battery state or screen on/off state changes, or third-party messages (like an experience sampling application that triggers a message every time the user sets an alarm or presses a button event).

Examples of implemented modules currently part of the Wockets system are: accelerometer raw data, activity counts data, accelerometer data features, activity recognition module based on the algorithm implemented in this thesis, activity recognition smoothing output, Bluetooth proximity, file scans, screen on/off state, installed and running apps, sms, call and browse history logs, GPS, WLAN, cell tower id, battery status, etc. Current implemented modules can be found on the developer documentation site [194].

7.2.5.2 Remote Server

As described in Intille et. al. [76], the Wockets software uses the phones data network to send motion summary data to a secure remote server on an hourly basis. This allows the physical activity summary data be viewed hourly, as well as, other types of data such as, for example, wearability patterns of sensors and EMA self-annotations about users physical activities, cognitive or emotional states, and context.

Every 24 hours, raw data is uploaded to the server via the WIFI network if available. In addition, in order to minimize user disruption and maximize data integrity and transfer speed, this operation is done over night when the users phone is charging and not in-use. After the data has been successfully uploaded to the server, it is deleted from the local storage on the phone.

Simple code was developed on the server side to track sensor wearability, detect any data anomalies or system performance problems, and send immediate feedback to participants over their phones to improve compliance. As future work, we believe this information can be used as a basis to create just-in-time interventions that can aid the user to achieve his health or performance goals.

Figure 7-3 shows an screen capture of the website used in the study when the sensing system was deployed along with windows mobile phones.

7.2.5.3 Software Key Features

The Wockets system incorporates a set of build in data collection tools and functionalities. Key features are listed below. Detailed and up-to-date features and specific components can be found at the developers website [194].

- Several optimizations for prolonging battery life. For example, some of these optimizations are: a delta compression algorithm coded in the firmware to minimize the memory space used to storage raw data, a state machine that adapts to the activity logging frequency whether the

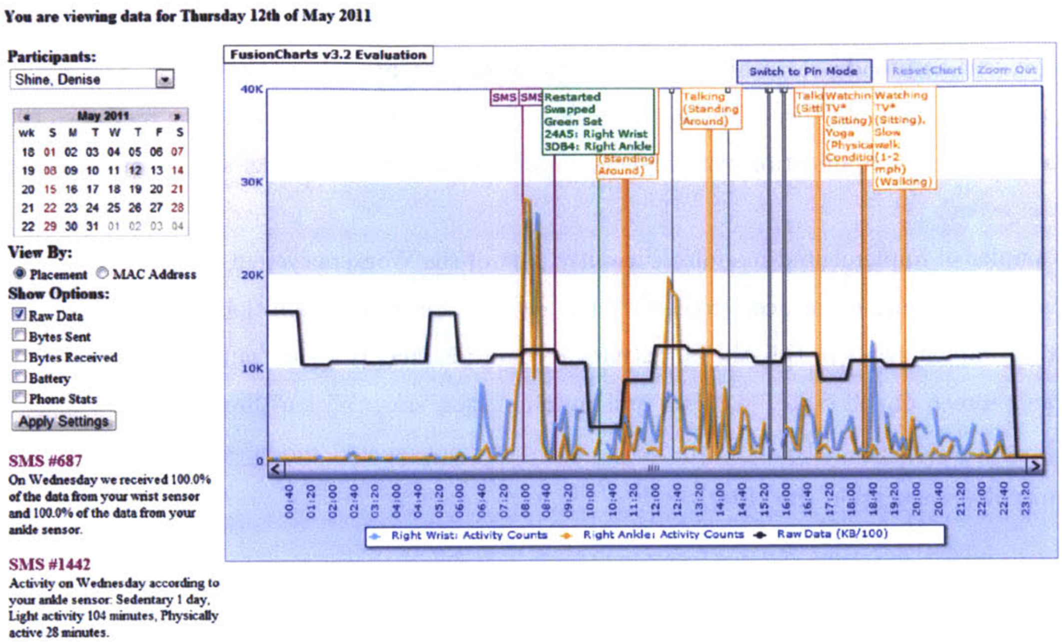


Figure 7-3: Study website.

sensors are within the phones range or their battery level, and solutions for everyday use-cases (e.g., only upload data when the phone is plugged to the charger, the WI-FI or cell network is available, and the user is not actively interacting with the phone UI or making a phone call).

- Delay tolerant implementation used when the Bluetooth or Internet connection is not available. It stores data locally until the Bluetooth receiver or server connection is restored. The raw data stored can be constrained by configuration in order that it doesnt take over the device storage space, but this is ultimately limited by the amount of space on the devices storage memory. The data can be compressed and/or aggregated by computing summary measures (e.g. activity counts or the extracted signal features).

- Remote configuration from a back-end server. The phone-side application can be configured to routinely check a remote server and download any configuration files or software updates. Configuration files can be defined as formatted XML files.

- Automatic or manual data upload. Automatic upload is done via built-in mechanisms for server communication for data upload and synchronization.

- Modular architecture. It allows for adding core functionalities and modifying existing behavior.

- Survey app for manual data collection. Surveys can be defined as a text file that is synchronized with the device as part of the remote configuration protocol. An example of a survey is shown in Figure 7-4.

- Field proven system. The system was validated and deployed in various laboratory settings, as well as, semi-naturalistic and naturalistic long-term field studies (for more information see [76] and [152]).

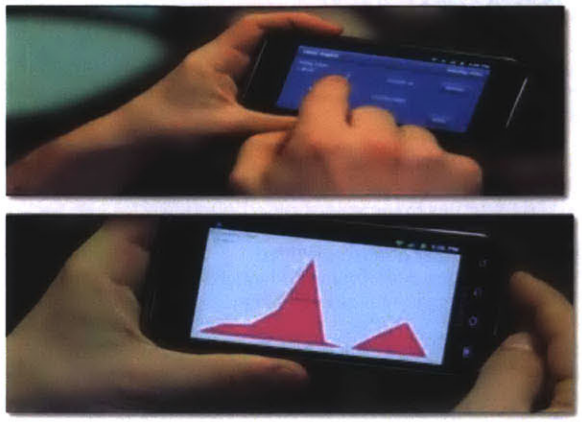


Figure 7-4: Example of survey protocols.

7.2.6 Power Consumption

We ran experiments to evaluate the power consumption of the Wockets System in both continuous and power-efficient transmission modes. The power measurements were taken by measuring the voltage drop across a resistor during phone use. When the Wocket is not connected to the phone, the radio is continuously awaiting a connection and therefore the current consumption is a relatively high 11.88 mAmps. When the Wocket is connected and transmits continuously, the power consumption registered 26.63 mAmps on average. With a 240 mAmps battery the Wocket operates for approximately 8.3 hours. In power-efficient mode, the Wocket shuts down its radio for approximately 45s, during which the power consumption is dramatically reduced. The average power consumption for the Wocket in power- efficient mode is 6.18 mA, lower than a Wocket waiting for connections. The lifetime of a 240 mAh battery exceeds 32 hours. For the phone, the baseline power consumption with the screen off is 10.86 mA. With the screen off and the Wockets connected, the power consumption jumps to 109.42 mA. For an 1100 mAh standard battery, the phone would operate for approximately 10 hours. Once the Wockets are configured to run in power-efficient mode, the consumption drops as low as 31.5 mA that allows the phone to run for over 34 hours (if no other functionality is being used). Table 7.1 shows the power consumption scheme.

Phone Usage	Power Consumption (mAmps)*
Phone doing nothing, screen off	10
Phone doing nothing, screen on	+132
Phone doing nothing, backlight	+105
Making phone call	+128
Playing MP3	+67
Downloading MP3 using GSM network	+352
Browsing internet using GSM network	+267
Bluetooth on	+6
Bluetooth discovering	+136
Bluetooth connect /reconnect	+6
1 Wocket transmitting in continuous mode (90Hz)	+86
2 Wockets transmitting in continuous mode (90Hz)	+97
3 Wockets transmitting in continuous mode (90Hz)	+101
1 Wocket transmitting in sniff mode (90Hz)	+95
2 Wockets transmitting in sniff mode (90Hz)	+96
3 Wockets transmitting in sniff mode (90Hz)	+97
Saving to external memory at 90Hz (1 wocket)	+1
Saving to external memory at 180Hz (2 wockets)	+2
Saving to external memory at 270Hz (3 wockets)	+3
Internal accelerometer at 25Hz	+3

* The power was profiled using a HTC Diamond Touch phone.

Table 7.1: System and phone's power consumption.

7.3 Data Collection Design Guidelines

7.3.1 Investigating Behavior in Free-Living Environments

The Stanford study has given us unprecedented insights into the understanding of fine-granularity physical activity. The great number of diverse data collected per person and about the study as a whole allow us to interpolate a behavioral image of a persons behavior throughout the studys duration.

This chapter starts with an overview of the Stanford dataset and basic statistics about the data. We then present analysis and results from several components of the Stanford study. These are but initial forays into the studys comprehensive dataset, some of which were done, as the study was ongoing, and served to inform the design of subsequent components and parts of the system design.

Aside from the direct research questions that we can answer, these first investigations have helped us formulate directions for further analysis, and demonstrate the potential of the Wockets data-rich methodology.

These components are aligned with one of the studys high-level goals of understanding the behavioral mechanisms related to motivate to increase physical activity and health, as well as designing and evaluating new tools and mechanisms to help people make better health-related decisions.

Besides that, there are other investigation trajectories currently ongoing by similar research initiatives. For instance, they are looking into the connection between behavioral patterns and wellness related topics, like sleep, stress, or mood, and/or investigating connections between personality traits of individuals and the social networks or contagious behaviors that form around them. Besides physical activity, these examples make use of a variety of signals collected during the study, including physical collocation of participants, self-reported social closeness, and mood, among several others.

7.3.2 Data Collection for System Validation

In the Stanford study, validation data was collected in four phases.

- Phase one was in a laboratory controlled setting and consisted of structured exercise activities with participants wearing a variety of sensors.
- Phase two added additional everyday activities with subjects wearing the same suite of sensors (and the latest version of the hardware).
- Phase three was the final validation study of the Wockets system, comparing the system classification output to self-report and selected sessions for a extended period of time.

To provide a sense of the effort, the data analyses were conduced during phase one and concluded on phase two, which represents the first 2.5 years of he starting of the project. There were many challenges along the path, from getting many sensor systems to work in sync in a reliable and robust manner to optimizing the power consumption of the sensing system and the phone, etc. As a result, in phase three, a simplified and more robust system and study protocol was used.

7.3.3 Remote Study Administration

In the validation phase of the study (phase four), summary activity data was sent back to researchers hourly using the phone's data network, allowing remote compliance monitoring and troubleshooting. It was also possible to trigger simple EMA self-report prompts on the phone and provide feedback to each participant about his/her overall physical activity level. These additional tools facilitate remote monitoring using exchange of data via SMS messaging, which is a more efficient way to send small amounts of data than requiring the phone to connect to the Internet via the data network.

7.4 Remarks

The Wockets system was first implemented for phones running the windows mobile OS version 6.x. Then, it was upgraded to Windows mobile OS version 8.x, which is the one described in this thesis. The first version (OS version 6.x) was the one used to carry out the data collections described in

[190] and corresponds the system described in Intille et al [76]. Whereas the upgraded version (OS version 8.x) is the one used in the projects described in [chang, 2014] and corresponds to the system described in this thesis. In particular, the upgraded version (OS version 8.x) extends the basic software version of the system to include the algorithm module described in chapter 3.

The Android OS version was developed and maintained at Northeastern University by Prof. Stephen Intilles research team (see project page [195]).

As it can be seen in [192], the original system hardware parts, including housing and materials, can be assembled in quantities of 100 by companies that make printed circuit boards. The cost of per sensor was approximately $63 US dollars including battery and housing. This cost is expected to be lower with large-scale production and over time -as the electronic components and batteries tend to be cheaper with time.

Chapter 8

Conclusion

8.1 Conclusion

The advances on miniaturized sensing, wireless communication technologies and smartphones have started to make possible to collect daily fine-grained activity data over a large number of individuals. These advances and the availability of large amounts of data have open up new possibilities for developing innovative applications that explore novel forms of interaction and behavior understanding. In particular, they have made possible to start exploring activity recognition systems that can be deployed in the real-world over long-periods of time. Henceforth, the focus of this thesis is on the development of a novel method for the recognition of physical activities, which can be used for wearable activity recognition systems deployed at scale in unconstrained natural environments.

A great attention has been placed on identifying important distinctive challenges faced by the activity recognition research community and discussing the crucial role of the interplay between the sensing, the algorithm and the user interaction components for the design of the computational methods and algorithms used by physical activity recognition systems. The specifics of these challenges are discussed with great detail in the background and algorithm sections of this thesis.

Additionally, a high emphasis was put on the evaluation of the feasibility of the proposed concept and method via the experiments introduced in chapter [chapter experiments] of this thesis. Finally, practical design guidelines are discussed based on the lessons learned throughout the implementation of a wearable activity recognition system called wockets, which was a large collaborative project funded by the NIH GEI biology exposure program [54] (see disclaimer in section 7.1 and project website [195] for more details about the project).

8.2 Contributions

- Assessment of various activity recognition problems with commonly used classification algorithms (Hidden Markov Models and Decision Trees).

- Analysis of the means to create robust activity algorithms for real-world systems, which allows to realize the concept of learning-upon-use.

- Demonstrate that the proposed algorithm concept is a valid approach to solve three main classification problems currently faced by real-world scalable activity recognition systems.

- Introduction of novel evaluation techniques based on design of experiments methods and techniques that simulate the learning-upon-use concept (learning previously unknown activities or learning them in an incremental manner using only limited amounts of data).

- Introduction and validation of a novel algorithm, based on binary SVMs models combined with a meta-learning majority of voting algorithm.

- Introduction and validation of the idea of a modular activity recognition, where the user is not required to provide data from all the activities the system recognizes. The system only requires data for the activities of interest and, using the new-labeled data previously unknown, only the relevant parts of the system are retrained the rest of the system remains the same- allowing it to preserve the information that has been already learned.

- Via lessons learned and design guidelines, this thesis informs designers about the challenges and trade-offs faced when designing a scalable activity recognition system intended to be used for long-term in natural settings. In fact, since these challenges are not only exclusive of activity recognition systems but also they are common across many wearable systems intended to recognize human behavior, some of the lessons learned and the practical approach used by the methods presented can benefit not only the activity recognition community but also other communities such as the Wearable Computing, Ubicomp, Behavioral Science communities.

- The listed contributions are both of theoretical (the novel concept and algorithm) and of practical value (the proposed evaluation technique). Furthermore, this thesis also discusses the practicalities of implementing the presented methods, in order to realize the envisioned learning upon use concept for activity recognition systems intended to be deployed for long-term in unconstrained natural settings.

8.3 Future Work

The contributions of this work can be viewed as several important milestones towards creating a wearable physical activity system that can be deployed at scale in natural unconstrained environments. Nevertheless, several questions still need to be answered. Therefore, future directions to continue and extend this work are introduced below.

First, this thesis argues that the proposed framework could be used to create robust building units part of a hierarchical activity recognition model, in where the recognition of high-level activities is based on the recognition results of other simpler activity instances. The motivation is to let the simpler activity instances - which are easier to identify- be first, and subsequently use them as building units for recognizing higher-level ones. For instance, a high-level activity like fighting may be recognized by detecting a sequence of several pushing and kicking interactions. Thus, in hierarchical approaches, a high-level activity is represented in terms of sub-instances which themselves might be decomposable until atomicity (last level of division of an activity) is obtained. As a result, sub-instances could serve as observations or entries for a higher-level activity recognizer.

Even though, this work doesnt focus on the hierarchical model per se, it is important to continue and extend this work for connecting it with a generic framework. I consider that the hierarchical activity recognition paradigm not only makes the recognition tasks computationally tractable and conceptually understandable, but also scalable and reusable by reducing the redundancy and utilizing common acknowledge or recognized sub-activity instances multiple times.

Second, more work on creating collaborative general-purpose datasets should be done. This is important because these datasets could uncover which is the distinctive variability among different activities and places providing a collection of representative training examples for activity recognition systems. If collaborative datasets are combined with a well-defined activity classification scheme, the research community could better investigate the problems of variability and subject- dependent recognition. In fact, through out the multiple arguments presented in this thesis, I advocate for making these datasets public and well-defined in terms of their activity taxonomy description since we believe that this will result on faster development and advancement of the algorithms used in the field.

Third, other important aspects to be improved in future work are the creation and use of unambiguous and well-designed experimental protocols and appropriate evaluation metrics. These aspects are crucial for reproducing proposed approaches and making solid comparisons across different recognition methods, which is only possible when they are tested with similar or comparable conditions. They can make a significant impact on moving system prototypes into wide-scale practical deployment. Currently, as it was discussed in chapters 2 and 7, such type of real-world deployment is quite challenging.

However, to conduct an unambiguous and well-designed experiment or study in activity recognition is more difficult than it might be thought at first. For instance, there are several issues faced such as: to maintain a balance between ergonomics and unobtrusiveness of the sensors versus ease-of-use and performance of the system; to allocate the time and resources required to prepare, conduct, and maintain the experiment; to cover the cost for participants, staff running the experiment, the equipment, and data or phone subscriptions used by participants.

Chapter 9

Appendix

9.1 SVM Model Confusion Matrices

SVM 379 features, 20 classes, polynomial kernel degree=1

Time taken to build model: 458.78 seconds

=== cross-validation ===

Correctly Classified Instances	10287	93.4078 %
Incorrectly Classified Instances	726	6.5922 %
Total Number of Instances	11013	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = unknown
0	1005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = sitting_slouching
0	0	431	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = sitting_at_desk
0	0	0	529	0	0	1	2	2	0	0	0	0	1	0	3	0	0	0	0	0	d = standing_hands
0	0	0	0	1468	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = treadmill_3mph_0
0	0	0	0	0	1557	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = airdyne_30_rpm
0	0	0	4	0	0	1299	16	48	37	5	2	0	0	0	3	0	0	0	3	1	g = folding_stacking
0	0	0	3	3	0	22	1139	20	17	2	0	0	3	0	1	0	0	0	8	3	h = vacuuming
0	0	0	14	0	0	81	28	674	5	1	3	0	2	0	2	0	0	0	2	2	i = washing_windows
0	0	0	0	0	0	67	21	8	1135	1	2	0	0	1	2	0	0	0	1	1	j = sweeping_mopping
0	0	0	4	0	0	11	4	16	4	152	3	0	0	1	0	0	0	0	26	4	k = stretching
0	0	0	5	0	0	5	0	5	0	2	55	0	1	0	0	0	0	0	0	2	l = swinging_arms
0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	m = pushups_from_knees
0	0	0	15	0	0	4	5	2	0	0	1	0	76	0	0	0	0	0	0	0	n = deep_knee_bends
0	0	0	1	0	0	1	8	6	0	1	0	0	0	180	0	0	0	0	0	0	o = stepping_platform
0	0	0	19	0	0	16	2	0	0	0	0	0	0	0	42	0	0	0	1	0	p = arm_curls_weights
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	3	0	0	q = situps_or_crunches
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	r = lying_down
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	s = picking_up
0	0	0	2	0	0	3	19	9	1	15	0	0	0	0	2	0	0	0	295	1	t = stretching
0	0	0	6	0	0	7	14	8	1	6	0	0	0	0	0	0	0	0	0	40	u = swinging_arms_t

Figure 9-1

SVM 64 features, 20 classes, polynomial kernel degree=2

Time taken to build model: 97.17 seconds

=== cross-validation ===

Correctly Classified Instances	10712	97.2669 %
Incorrectly Classified Instances	301	2.7331 %
Total Number of Instances	11013	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = unknown
0	1005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = sitting_slouching
0	0	431	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = sitting_at_desk_
0	0	0	538	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = standing_hands_
0	0	0	0	1468	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = treadmill_3mph_0
0	0	0	0	0	1557	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = airdyne_30_rpm
0	0	0	0	0	0	1356	20	17	18	4	1	0	0	0	2	0	0	0	0	0	g = folding_stacking_l
0	0	0	0	6	0	16	1185	3	6	1	0	0	0	0	2	0	0	0	0	2	h = vacuuming_moving_c
0	0	0	0	0	0	37	19	755	3	0	0	0	0	0	0	0	0	0	0	0	i = washing_windows
0	0	0	0	1	0	27	3	4	1201	1	1	0	0	0	0	0	0	0	0	1	j = sweeping_mopping
0	0	0	0	0	0	3	5	13	3	198	0	0	1	2	0	0	0	0	0	0	k = stretching
0	0	0	0	0	0	1	1	5	0	0	63	0	5	0	0	0	0	0	0	0	l = swinging_arms
0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	m = pushups_from_knees
0	0	0	1	0	0	0	1	6	0	0	0	0	95	0	0	0	0	0	0	0	n = deep_knee_bends
0	0	0	0	0	0	0	1	0	0	1	0	0	0	195	0	0	0	0	0	0	o =stepping_on_platform
0	0	0	0	0	0	1	3	0	0	0	0	0	0	0	75	0	0	0	0	1	p = arm_curls _hand_w
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	0	q = situps_or_crunches
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	r = lying_down
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	s = picking_up
0	0	0	0	0	0	0	16	0	1	0	0	0	0	0	0	0	0	0	330	0	t = stretching_standing
0	0	0	0	2	0	3	21	0	0	1	4	0	0	0	4	0	0	0	0	47	u = swinging_arms_turn

Figure 9-2

SVM 42 features, 20 classes, polynomial kernel degree=2

Time taken to build model: 1948.01 seconds

```
=== cross-validation ==
Correctly Classified Instances       10303               93.5542 %
Incorrectly Classified Instances       710                6.4458 %
Total Number of Instances            11013
```

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = unknown
0	1758	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = sitting_in_
0	0	731	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = sitting_at_desk
0	0	0	1031	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = standing_
0	0	0	1	2428	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	e = treadmill_3mph_0
0	0	0	0	0	2651	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = airdyne_30_rpm
0	0	0	0	0	0	2205	118	41	66	6	1	0	0	0	4	0	0	0	0	0	g = folding_
0	0	0	23	35	0	104	1719	15	13	2	0	0	1	5	4	0	0	0	6	1	h = vacuuming_
0	0	0	4	0	0	139	100	1183	9	1	0	0	0	2	0	0	0	0	0	0	i = washing_windows
0	0	0	0	1	0	106	4	17	1930	2	3	0	0	0	2	0	0	0	3	0	j = sweeping_mopping
0	0	0	0	2	0	12	29	11	17	279	0	0	9	4	0	0	0	0	1	0	k = stretching
0	0	0	0	0	0	1	12	27	2	0	76	0	1	0	0	0	0	0	0	0	l = swinging_arms_s
0	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0	0	0	m = pushups_from_knees
0	0	0	0	0	0	9	29	15	0	0	0	0	125	0	0	0	0	0	0	0	n = deep_knee_
0	0	0	0	3	1	2	15	0	0	0	0	0	0	303	0	0	0	0	0	0	o =stepping_on_platform
0	0	0	1	0	0	3	39	0	3	0	0	0	0	0	87	0	0	0	0	0	p = arm_curls_
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	2	0	0	0	q = situps_or_crunches
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	137	0	0	0	r = lying_down
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	126	0	0	s = picking_up
0	0	0	4	1	0	3	24	0	3	1	0	0	0	0	0	0	0	0	571	0	t = stretching
0	0	0	6	10	0	3	44	4	2	0	0	0	0	3	0	0	0	0	0	60	u = swinging_arms_p

Figure 9-3

SVM 18 features, 20 classes, polynomial kernel degree=2

Time taken to build model: 161.28 seconds

=== cross-validation ===

Correctly Classified Instances	8802	79.9237 %
Incorrectly Classified Instances	2211	20.0763 %
Total Number of Instances	11013	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = unknown
0	1005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = sitting_slouching
0	0	431	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = sitting_at_desk
0	0	0	520	0	0	0	4	7	7	0	0	0	0	0	0	0	0	0	0	0	d = standing_hands
0	0	0	0	1467	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	e = treadmill_3mph_0
0	0	0	0	0	1557	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = airdyne_30_rpm
0	0	0	4	0	0	1148	107	32	114	13	0	0	0	0	0	0	0	0	0	0	g = folding_laundry
0	0	0	50	16	0	64	922	52	111	3	0	0	0	3	0	0	0	0	0	0	h = vacuuming
0	0	0	13	0	0	275	42	392	92	0	0	0	0	0	0	0	0	0	0	0	i = washing_windows
0	0	0	74	0	0	231	186	72	669	6	0	0	0	1	0	0	0	0	0	0	j = sweeping_mopping
0	0	0	59	0	0	5	14	13	17	117	0	0	0	0	0	0	0	0	0	0	k = stretching
0	0	0	0	0	0	26	8	34	7	0	0	0	0	0	0	0	0	0	0	0	l = swinging_arms
0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	m = pushups_from_knees
0	0	0	16	0	0	0	34	10	28	0	0	0	15	0	0	0	0	0	0	0	n = deep_knee_bends
0	1	0	2	96	1	0	19	0	16	0	0	0	0	62	0	0	0	0	0	0	o = stepping_platform
0	0	0	0	0	0	0	17	15	48	0	0	0	0	0	0	0	0	0	0	0	p = arm_curls_weights
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	10	0	0	q = situps_or_crunches
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	76	0	0	0	r = lying_down
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	s = picking_up
0	0	0	0	1	0	0	35	0	30	0	0	0	0	0	0	0	0	0	281	0	t = stretching
0	0	0	2	4	0	0	51	0	7	0	0	0	0	1	0	0	0	0	0	17	u = swinging_arms

Figure 9-4

SVM 18 features, 20 classes, polynomial kernel degree=3

Time taken to build model: 194.63 seconds

=== cross-validation ===

Correctly Classified Instances	9250	83.9916 %
Incorrectly Classified Instances	1763	16.0084 %
Total Number of Instances	11013	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a = unknown
0	1005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = sitting_slouching
0	0	431	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = sitting_at_desk
0	0	0	523	0	0	0	1	13	1	0	0	0	0	0	0	0	0	0	0	0	d = standing_hands
0	0	0	0	1467	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	e = treadmill_3mph_0
0	0	0	0	0	1557	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = airdyne_30_rpm
0	0	0	3	0	0	1127	97	69	112	10	0	0	0	0	0	0	0	0	0	0	g = folding_stacking
0	0	0	22	13	0	50	995	35	97	7	0	0	0	2	0	0	0	0	0	0	h = vacuuming_moving
0	0	0	8	0	0	185	34	542	45	0	0	0	0	0	0	0	0	0	0	0	i = washing_windows
0	0	0	45	0	0	205	143	92	740	13	0	0	0	1	0	0	0	0	0	0	j = sweeping_mopping
0	0	0	42	0	0	7	3	13	6	154	0	0	0	0	0	0	0	0	0	0	k = stretching
0	0	0	0	0	0	14	8	46	7	0	0	0	0	0	0	0	0	0	0	0	l = swinging_arms
0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	m = pushups_from_knees
0	0	0	17	0	0	1	15	27	12	1	0	0	30	0	0	0	0	0	0	0	n = deep_knee_bends_
0	1	0	2	49	1	0	18	0	14	1	0	0	0	111	0	0	0	0	0	0	o = stepping_platform
0	0	0	0	0	0	0	17	58	5	0	0	0	0	0	0	0	0	0	0	0	p = arm_curls_h_weights
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	0	q = situps_or_crunches
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	79	0	0	0	r = lying_down
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	s = picking_up
0	0	0	0	0	0	0	12	0	7	2	0	0	0	0	0	0	0	0	326	0	t = stretching_s_up
0	0	0	0	3	0	0	36	1	9	0	0	0	1	2	0	0	0	0	0	30	u = swinging_arms_p

Figure 9-5

Bibliography

[1] D. B. Rubin A. P. Dempster, N. M. Laird. Maximum likelihood from incomplete data via the EM algorithm.

[2] Saeed Abdullah, Nicholas D. Lane, and Tanzeem Choudhury. Towards Population Scale Activity Recognition : A Framework for Handling Data Diversity. In *AAAI: Twenty-Sixth AAAI Conference on Artificial Intelligence*, page 10, 2012.

[3] D Abowd, A K Dey, R Orr, and J Brotherton. Context-awareness in wearable and ubiquitous computing. *Virtual Reality*, 3(3):200–211, 1998.

[4] Actigraph. http://www.actigraphcorp.com/.

[5] John G. Adair. The Hawthorne effect: A reconsideration of the methodological artifact. *Journal of Applied Psychology*, 69(2):334–345, 1984.

[6] J.K. Aggarwal and Q. Cai. Human motion analysis: a review. In *Proceedings IEEE Nonrigid and Articulated Motion Workshop*, pages 90–102. IEEE Comput. Soc, June 1997.

[7] Nadav Aharony, Wei Pan, Cory Ip, Inas Khayal, and Alex Pentland. Social fMRI: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing*, 7(6):643–659, December 2011.

[8] B E Ainsworth, W L Haskell, M C Whitt, M L Irwin, A M Swartz, S J Strath, W L O'Brien, D R Bassett, K H Schmitz, P O Emplaincourt, D R Jacobs, and A S Leon. Compendium of physical activities: an update of activity codes and MET intensities. *Medicine and science in sports and exercise*, 32(9 Suppl):S498–504, September 2000.

[9] Barbara E Ainsworth, William L Haskell, Stephen D Herrmann, Nathanael Meckes, David R Bassett, Catrine Tudor-Locke, Jennifer L Greer, Jesse Vezina, Melicia C Whitt-Glover, and Arthur S Leon. 2011 Compendium of Physical Activities: a second update of codes and MET values. *Medicine and science in sports and exercise*, 43(8):1575–81, August 2011.

[10] BE Ainsworth, WL Haskell, and MC Whitt. Compendium of physical activities: an update of activity codes and MET intensities. ... *in sports and exercise*, 2000.

[11] Fahd Albinali, Stephen Intille, William Haskell, and Mary Rosenberger. Using wearable activity type detection to improve physical activity energy expenditure estimation. In *Proceedings of the 12th ACM international conference on Ubiquitous computing - Ubicomp '10*, page 311, New York, New York, USA, September 2010. ACM Press.

[12] Amazon. Amazon.

[13] Peter D. Hoff Assaf P. Oron. Kruskal-Wallis and Friedman type tests for nested effects in hierarchical designs 1.

[14] AutomateIt App. AutomateIt: http://automateitapp.com/.

[15] Ling Bao and StephenS. Intille. Activity Recognition from User-Annotated Acceleration Data. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing SE - 1*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg, 2004.

[16] L. F. Barrett and D. J. Barrett. An Introduction to Computerized Experience Sampling in Psychology. *Social Science Computer Review*, 19(2):175–185, May 2001.

[17] Ari Y. Benbasat and Joseph A. Paradiso. A framework for the automated generation of power-efficient classifiers for embedded sensor nodes. In *Proceedings of the 5th international conference on Embedded networked sensor systems - SenSys '07*, page 219, New York, New York, USA, November 2007. ACM Press.

[18] Martin Berchtold, Matthias Budde, Dawud Gordon, Hedda R. Schmidtke, and Michael Beigl. ActiServ: Activity Recognition Service for mobile phones. In *International Symposium on Wearable Computers (ISWC) 2010*, pages 1–8. IEEE, October 2010.

[19] Jit Biswas, Matthias Baumgarten, Andrei Tolstikov, AungAungPhyo Wai, Chris Nugent, Liming Chen, and Mark Donnelly. Requirements for the Deployment of Sensor Based Recognition Systems for Ambient Assistive Living. In Yeunsook Lee, Z.Zenn Bien, Mounir Mokhtari, JeongTai Kim, Mignon Park, Jongbae Kim, Heyoung Lee, and Ismail Khalil, editors, *Aging Friendly Technology for Health and Independence SE - 27*, volume 6159 of *Lecture Notes in Computer Science*, pages 218–221. Springer Berlin Heidelberg, 2010.

[20] Ulf Blanke and Bernt Schiele. Remember and transfer what you have learned - recognizing composite activities based on activity spotting. In *International Symposium on Wearable Computers (ISWC) 2010*, pages 1–8. IEEE, October 2010.

[21] BM-Innovations. BM innovations GmbH development and product website.

[22] Bodymedia. Sensewear: http://sensewear.bodymedia.com/.

[23] Marko Borazio and Kristof Van Laerhoven. Improving activity recognition without sensor data. In *Proceedings of the 4th Augmented Human International Conference on - AH '13*, pages 108–115, New York, New York, USA, March 2013. ACM Press.

[24] Oliver Brdiczka, James L Crowley, and Patrick Reignier. Learning situation models in a smart home. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 39(1):56–63, February 2009.

[25] Michael Buettner, Richa Prasad, Matthai Philipose, and David Wetherall. Recognizing daily activities with RFID-based sensors. In *Proceedings of the 11th international conference on Ubiquitous computing - Ubicomp '09*, page 51, New York, New York, USA, September 2009. ACM Press.

[26] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Computing Surveys*, 2013.

[27] Bureau of Labor Statistics. ATUS: American Time Use Survey.

[28] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition.

[29] Michelle Nicole Burns, Mark Begale, Jennifer Duffecy, Darren Gergle, Chris J Karr, Emily Giangrande, and David C Mohr. Harnessing context sensing to develop a mobile intervention for depression. *Journal of medical Internet research*, 13(3):e55, January 2011.

[30] Johannes B. J. Bussmann, Ulrich W. Ebner-Priemer, and Jochen Fahrenberg. Ambulatory activity monitoring: Progress in measurement of activity, posture, and specific motion patterns in daily life. *European Psychologist*, 14(2):142, 2009.

[31] Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Hong Lu, Xiao Zheng, Mirco Musolesi, Kristóf Fodor, and Gahng-Seop Ahn. The Rise of People-Centric Sensing. *IEEE Internet Computing*, 12(4):12–21, July 2008.

[32] Claudette Cédras and Mubarak Shah. Motion-based recognition a survey. *Image and Vision Computing*, 13(2):129–155, 1995.

[33] Center for Behavioral Intervention Technologies. Purple Robot.

[34] Center for Time Use Research. MTUS: Multinational Time Use Study.

[35] Liming Chen, Jesse Hoey, CD Nugent, DJ Cook, and Zhiwen Yu. Sensor-based activity recognition. 42(6):790–808, 2012.

[36] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Haehnel, Beverly Harrison, Bruce Hemingway, Jeffrey Hightower, Predrag "Pedja" Klasnja, Karl Koscher, Anthony LaMarca, James A. Landay, Louis LeGrand, Jonathan Lester, Ali Rahimi, Adam Rea, and Danny Wyatt. The Mobile Sensing Platform: An Embedded Activity Recognition System. *IEEE Pervasive Computing*, 7(2):32–41, April 2008.

[37] Tanzeem Choudhury, Aaron Quigley, Thomas Strang, and Koji Suginuma, editors. *Location and Context Awareness*, volume 5561 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[38] Ian Cleland, Basel Kikhia, Chris Nugent, Andrey Boytsov, Josef Hallberg, Kå re Synnes, Sally McClean, and Dewar Finlay. Optimal placement of accelerometers for the detection of everyday activities. *Sensors (Basel, Switzerland)*, 13(7):9183–200, January 2013.

[39] Nello Cristianini Colin Campbell. Simple Learning Algorithms for Training Support Vector Machines.

[40] S. Consolvo and M. Walker. Using the experience sampling method to evaluate ubicomp applications. *IEEE Pervasive Computing*, 2(2):24–31, April 2003.

[41] Sunny Consolvo, Ryan Libby, Ian Smith, James A. Landay, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, and Louis LeGrand. Activity sensing in the wild: a field trial of ubifit garden. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, pages 1797–1806, New York, New York, USA, April 2008. ACM Press.

[42] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, September 1995.

[43] Mihaly Csikszentmihalyi and Reed Larson. Validity and reliability of the experience-sampling method. *The Journal of nervous and mental disease*, 175(9):526–536, 1987.

[44] Dennis Decoste and Bernhard Schölkopf. Training Invariant Support Vector Machines. *Machine Learning*, 46(1-3):161–190, March 2002.

[45] Dan Ding, Rory A Cooper, Paul F Pasquina, and Lavinia Fici-Pasquina. Sensor technology for smart homes. *Maturitas*, 69(2):131–6, June 2011.

[46] Trinh Minh Tri Do and Daniel Gatica-Perez. GroupUs: Smartphone Proximity Data and Human Interaction Type Mining. In *2011 15th Annual International Symposium on Wearable Computers*, pages 21–28. IEEE, June 2011.

[47] N. Eagle and A. Pentland. Social Serendipity: Mobilizing Social Software. *IEEE Pervasive Computing*, 4(2):28–34, April 2005.

[48] Federico Girosi Edgar Osuna, Robert Freund. An Improved Training Algorithm for Support Vector Machines.

[49] Eegosports. eegosports — ANT Neuro.

[50] Louis LeGrand Ali Rahimi Adam Rea Gaetano Borriello Bruce Hemingway Karl Koscher James A. Landay Jonathan Lester Danny Wyatt Dirk Haehnel et al. Tanzeem Choudhury, Anthony LaMarca. The Mobile Sensing Platform: An Embedded Activity Recognition System.

[51] ETH Activity Recognition Challenge. Activity Recognition Challenge - Dataset — Opportunity.

[52] ETH Opportunity Datasets. Opportunity Dataset - Publications — Opportunity.

[53] Evans. Paco, 2011.

[54] Exposure Biology Program. The Genes, Environment and Health Initiative.

[55] Hossein Falaki, Ratul Mahajan, and Deborah Estrin. SystemSens. In *Proceedings of the sixth international workshop on MobiArch - MobiArch '11*, page 25, New York, New York, USA, June 2011. ACM Press.

[56] Davide Figo, Pedro C. Diniz, Diogo R. Ferreira, and João M. P. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, March 2010.

[57] K.P. Fishkin, M. Philipose, and A. Rea. Hands-On RFID: Wireless Wearables for Detecting Use of Objects. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 38–43. IEEE, 2005.

[58] Fitbit. http://www.fitbit.com/.

[59] FitnessKeeper Inc. RunKeeper: http://runkeeper.com/.

[60] Fitocracy. https://www.fitocracy.com/.

[61] Kilian Forster, Daniel Roggen, and Gerhard Troster. Unsupervised Classifier Self-Calibration through Repeated Context Occurences: Is there Robustness against Sensor Displacement to Gain? In *2009 International Symposium on Wearable Computers*, pages 77–84. IEEE, September 2009.

[62] Jon Froehlich, Mike Y. Chen, Sunny Consolvo, Beverly Harrison, and James A. Landay. MyExperience. In *Proceedings of the 5th international conference on Mobile systems, applications and services - MobiSys '07*, page 57, New York, New York, USA, June 2007. ACM Press.

[63] D.M Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[64] Hans W. Gellersen, Albercht Schmidt, and Michael Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mobile Networks and Applications*, 7(5):341–351, October 2002.

[65] Harris Georgiou, Michael Mavroforakis, and Sergios Theodoridis. *Artificial Neural Networks ICANN 2006*, volume 4131 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, September 2006.

[66] James Goh, editor. *The 15th International Conference on Biomedical Engineering*, volume 43 of *IFMBE Proceedings*. Springer International Publishing, Cham, 2014.

[67] Marta C González, César A Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–82, June 2008.

[68] Lara González-Villanueva, Stefano Cagnoni, and Luca Ascari. Design of a wearable sensing system for human motion monitoring in physical rehabilitation. *Sensors (Basel, Switzerland)*, 13(6):7735–55, January 2013.

[69] Google and Google Play Services. Activity Recognition API.

[70] Tao Gu, Wu Zhanqing, Tao Xianping, Hung Keng Pung, and Lu Jian. epSICAR: An Emerging Patterns based approach to sequential, interleaved and Concurrent Activity Recognition. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–9. IEEE, March 2009.

[71] William Haskell, Denise Shine, Abey Mukkananchery, Mary Rosenberg, Stephen Intille, Fahd Albinali, Selene Mota, and Jason Nawyn. SWAP Study.

[72] Joel M. Hektner and Mihaly Csikszentmihalyi. The experience sampling method: Measuring the context and content of lives. In *Handbook of environmental psychology*, chapter 8, pages 233–243. 2002.

[73] Joyce Ho and Stephen S. Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, page 909, New York, New York, USA, April 2005. ACM Press.

[74] Tâm Huynh, Mario Fritz, and Bernt Schiele. Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, page 10, New York, New York, USA, September 2008. ACM Press.

[75] Eibe Frank Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques.* The Morgan Kaufmann Series in Data Management Systems, 3rd edition.

[76] Stephen S Intille, Fahd Albinali, Selene Mota, Benjamin Kuris, Pilar Botana, and William L Haskell. Design of a wearable physical activity monitoring system using mobile phones and accelerometers. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2011:3636–9, January 2011.

[77] Stephen S. Intille, Jason Nawyn, Beth Logan, and Gregory D. Abowd. Developing shared home behavior datasets to advance HCI and ubiquitous computing research. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09*, page 4763, New York, New York, USA, April 2009. ACM Press.

[78] Stephen S. Intille, John Rondoni, Charles Kukla, Isabel Ancona, and Ling Bao. A context-aware experience sampling tool. In *CHI '03 extended abstracts on Human factors in computing systems - CHI '03*, page 972, New York, New York, USA, April 2003. ACM Press.

[79] StephenS. Intille, Kent Larson, EmmanuelMunguia Tapia, JenniferS. Beaudin, Pallavi Kaushik, Jason Nawyn, and Randy Rockinson. Using a Live-In Laboratory for Ubiquitous Computing Research. In KennethP. Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley, editors, *Pervasive Computing SE - 22*, volume 3968 of *Lecture Notes in Computer Science*, pages 349–365. Springer Berlin Heidelberg, 2006.

[80] Jawbone. Jawbone Band: https://jawbone.com/.

[81] Thorsten Joachims. Making large-scale support vector machine learning practical. pages 169–184, February 1999.

[82] Thorsten Joachims. Estimating the Generalization Performance of an SVM Efficiently. pages 431–438, June 2000.

[83] Daniel Kahneman, Alan B Krueger, David A Schkade, Norbert Schwarz, and Arthur A Stone. A survey method for characterizing daily life experience: the day reconstruction method. *Science (New York, N.Y.)*, 306(5702):1776–80, December 2004.

[84] Ashish Kapoor and Eric Horvitz. Principles of Lifelong Learning for Predictive User Modeling. In Cristina Conati, Kathleen McCoy, and Georgios Paliouras, editors, *User Modeling 2007 SE -*

7, volume 4511 of *Lecture Notes in Computer Science*, pages 37–46. Springer Berlin Heidelberg, 2007.

[85] Ashish Kapoor and Eric Horvitz. Experience sampling for building predictive user models: a comparative study. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, page 657, New York, New York, USA, April 2008. ACM Press.

[86] S. Katz, T. D. Downs, H. R. Cash, and R. C. Grotz. Progress in Development of the Index of ADL. *The Gerontologist*, 10(1 Part 1):20–30, March 1970.

[87] Nicky Kern, Bernt Schiele, Holger Junker, Paul Lukowicz, and Gerhard Tröster. Wearable sensing to annotate meeting recordings. *Personal and Ubiquitous Computing*, 7(5):263–274, 2003.

[88] Julie A. Kientz, Shwetak N. Patel, Brian Jones, Ed Price, Elizabeth D. Mynatt, and Gregory D. Abowd. The Georgia Tech aware home. In *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08*, page 3675, New York, New York, USA, April 2008. ACM Press.

[89] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143, August 1995.

[90] Stefanos D. Kollias, Andreas Stafylopatis, Wodzisaw Duch, and Erkki Oja, editors. *Artificial Neural Networks ICANN 2006*, volume 4131 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[91] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research*, 32(8):951–970, July 2013.

[92] Ulrich H.-G. Kreß el. Pairwise classification and support vector machines. pages 255–268, February 1999.

[93] Coco Krumme, Alejandro Llorente, Manuel Cebrian, Alex Sandy Pentland, and Esteban Moro. The predictability of consumer visitation patterns. *Scientific reports*, 3:1645, January 2013.

[94] Kristof Laerhoven. Porcupines.

[95] Kristof Van Laerhoven. Porcupine sensor.

[96] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous*

information management and communication - ICUIMC '08, page 208, New York, New York, USA, January 2008. ACM Press.

[97] Nicholas D. Lane, Ye Xu, Hong Lu, Andrew T. Campbell, Tanzeem Choudhury, and Shane B. Eisenman. Exploiting Social Networks for Large-Scale Human Behavior Modeling. *IEEE Pervasive Computing*, 10(4):45–53, April 2011.

[98] Nicholas D. Lane, Ye Xu, Hong Lu, Shaohan Hu, Tanzeem Choudhury, Andrew T. Campbell, and Feng Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the 13th international conference on Ubiquitous computing - UbiComp '11*, page 355, New York, New York, USA, September 2011. ACM Press.

[99] Óscar D. Lara, Alfredo J. Pérez, Miguel A. Labrador, and José D. Posada. Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and Mobile Computing*, 8(5):717–729, October 2012.

[100] Young-Dong Lee and Wan-Young Chung. Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring. *Sensors and Actuators B: Chemical*, 140(2):390–395, 2009.

[101] Thomas Leung, Yang Song, and John Zhang. Handling label noise in video classification via multiple instance learning. In *2011 International Conference on Computer Vision*, pages 2056–2063. IEEE, November 2011.

[102] Michael Lew, Nicu Sebe, Thomas S. Huang, and Erwin M. Bakker, editors. *HumanComputer Interaction*, volume 4796 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[103] Xuchun Li, Lei Wang, and Eric Sung. AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, 21(5):785–795, August 2008.

[104] Libelium. Libelium: Detecting iPhone and Android Smartphones by WiFi and Bluetooth [Cellular - Mobile - Hand Phone Detection] — Libelium.

[105] Karen Kay-Lynn Liu. *A personal, mobile system for understanding stress and interruptions.* Phd diss., Massachusetts Institute of Technology, 2004.

[106] Shaopeng Liu, Robert X Gao, Dinesh John, John Staudenmayer, and Patty S Freedson. SVM-based multi-sensor fusion for free-living physical activity assessment. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, 2011:3188–91, January 2011.

[107] Beth Logan, Jennifer Healey, Matthai Philipose, EmmanuelMunguia Tapia, and Stephen Intille. A Long-Term Evaluation of Sensing Modalities for Activity Recognition. In John Krumm, GregoryD. Abowd, Aruna Seneviratne, and Thomas Strang, editors, *UbiComp 2007: Ubiquitous Computing SE - 28*, volume 4717 of *Lecture Notes in Computer Science*, pages 483–500. Springer Berlin Heidelberg, 2007.

[108] Brent Longstaff, Sasank Reddy, and Deborah Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *Proceedings of the 4th International ICST Conference on Pervasive Computing Technologies for Healthcare*, pages 1–7. IEEE, 2010.

[109] P. Lukowicz, S. Pentland, and a. Ferscha. From Context Awareness to Socially Aware Computing. *IEEE Pervasive Computing*, 11(1):32–41, 2012.

[110] Andrea Mannini and Angelo Maria Sabatini. On-line classification of human activity and estimation of walk-run speed from acceleration data using support vector machines. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, 2011:3302–5, January 2011.

[111] J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, volume 2, pages 747–752. IEEE, 2001.

[112] K. Mase. Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, 1(3):24–32, July 2002.

[113] Winter Mason and Siddharth Suri. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, 44(1):1–23, 2012.

[114] U. Maurer, A. Smailagic, D.P. Siewiorek, and M. Deisher. Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions. In *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pages 113–116. IEEE.

[115] Marci Meingast, Tanya Roosta, and Shankar Sastry. Security and privacy issues with health care information technology. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 1:5453–8, January 2006.

[116] Microsoft Research Asia. GeoLife GPS Trajectories, 2012.

[117] Misfit Wearables. Shine Activity Monitor: http://www.misfitwearables.com/.

[118] Thomas B. Moeslund, Adrian Hilton, Volker Krüger, Pascal Fua, and Remi Ronfard. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[119] Yves-Alexandre Montjoye, Jordi Quoidbach, Florent Robic, and Alex(Sandy) Pentland. Predicting Personality Using Novel Mobile Phone-Based Metrics. In ArielM. Greenberg, WilliamG. Kennedy, and NathanD. Bos, editors, *Social Computing, Behavioral-Cultural Modeling and Prediction SE - 6*, volume 7812 of *Lecture Notes in Computer Science*, pages 48–55. Springer Berlin Heidelberg, 2013.

[120] Michael C. Mozer. The neural network house: An environment that adapts to its inhabitants. In *Proc. AAAI Spring Symp. Intelligent Environments*, pages 110–114. 1998.

[121] 1978 Munguia Tapia, Emmanuel. Using machine learning for real-time activity recognition and estimation of energy expenditure, 2008.

[122] Kevin Murphy. Matlab Toolbox for Hidden Markov Models.

[123] Nadav Aharony, Wei Pan, Cory Ip, Inas Khayal and Alex Pentland. FUNF.

[124] Netflix. Netflix Movie Recommender.

[125] The-Minh Nguyen, Takahiro Kawamura, Yasuyuki Tahara, and Akihiko Ohsuga. Capturing Users Buying Activity at Akihabara Electric Town from Twitter. In Jeng-Shyang Pan, Shyi-Ming Chen, and NgocThanh Nguyen, editors, *Computational Collective Intelligence. Technologies and Applications SE - 18*, volume 6422 of *Lecture Notes in Computer Science*, pages 163–171. Springer Berlin Heidelberg, 2010.

[126] The Minh Nguyen, Takahiro Kawamura, Yasuyuki Tahara, and Akihiko Ohsuga. Self-supervised capturing of users' activities from weblogs. *International Journal of Intelligent Information and Database Systems*, 6(1):61, January 2012.

[127] Nike. Fuelband: http://www.nike.com/us/en_us/c/nikeplus-fuelband.

[128] Preetham Mysore Michael L. Littman Nishkam Ravi, Nikhil D. Activity recognition from accelerometer data.

[129] Greg Norcie. Ethical and Practical Considerations For Compensation of Crowdsourced Research Participants Introduction. In *CHI WS on Ethics Logs and VideoTape: Ethics in Large Scale Trials & User Generated Content*, 2011.

[130] Georg Ogris, Thomas Stiefmeier, Paul Lukowicz, and Gerhard Troster. Using a complex multi-modal on-body sensor system for activity spotting. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 55–62. IEEE, 2008.

[131] Daniel Olguin and Alex Pentland. Human Activity Recognition : Accuracy across Common Locations for Wearable Sensors. In *10th Int. Symp. Wearable Comput. (Student Colloq.)*, pages 11–13, 2006.

[132] A. Pantelopoulos and N.G. Bourbakis. A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, January 2010.

[133] Sheng-Ying Pao, Selene Mota, Keywon Chung, and Alexander Reben. A need-driven design approach. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12*, page 829, New York, New York, USA, February 2012. ACM Press.

[134] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen. Activity Classification Using Realistic Data From Wearable Sensors. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):119–128, January 2006.

[135] Kurt Partridge and Philippe Golle. On using existing time-use study data for ubiquitous computing applications. In *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, page 144, New York, New York, USA, September 2008. ACM Press.

[136] Shyamal Patel, Chiara Mancinelli, Jennifer Healey, Marilyn Moy, and Paolo Bonato. Using Wearable Sensors to Monitor Physical Activities of Patients with COPD: A Comparison of Classifier Performance. In *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, pages 234–239. IEEE, June 2009.

[137] Daniel Peebles, Hong Lu, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. Community-Guided Learning : Exploiting Mobile Sensor Users to Model Human Behavior. In *AAAI*, pages 1600–1606, 2010.

[138] Adrian Perrig, John Stankovic, and David Wagner. Security in wireless sensor networks. *Communications of the ACM*, 47(6):53, June 2004.

[139] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring Activities from Interactions with Objects. *IEEE Pervasive Computing*, 3(4):50–57, October 2004.

[140] Philips Research. 365 days: Ambient Intelligence research in HomeLab, 2003.

[141] Rosalind W. Picard and Karen K. Liu. Relative subjective count and assessment of interruptive technologies applied to mobile monitoring of stress. *International Journal of Human-Computer Studies*, 65(4):361–375, 2007.

[142] John C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, February 1999.

[143] Quantified Self. The Quantified Self Web Page.

[144] C. Randell and H. Muller. Context awareness by analysing accelerometer data. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, pages 175–176. IEEE Comput. Soc.

[145] Pedro M R Reis, Felix Hebenstreit, Florian Gabsteiger, Vinzenz von Tscharner, and Matthias Lochmann. Methodological aspects of EEG and body dynamics measurements during motion. *Frontiers in human neuroscience*, 8:156, January 2014.

[146] Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '12*, page 1, New York, New York, USA, June 2012. ACM Press.

[147] Attila Reiss and Didier Stricker. Introducing a New Benchmarked Dataset for Activity Monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109. IEEE, June 2012.

[148] Attila Reiss, Didier Stricker, and Gustaf Hendeby. Confidence-based multiclass AdaBoost for physical activity monitoring. In *Proceedings of the 17th annual international symposium on International symposium on wearable computers - ISWC '13*, page 13, New York, New York, USA, September 2013. ACM Press.

[149] D. Roggen, Gerhard Troster, P. Lukowicz, A. Ferscha, Jose del R. Millan, and R. Chavarriaga. Opportunistic human activity and context recognition. *Computer*, 46(2):36–45, February 2013.

[150] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Forster, Gerhard Troster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and Jose del R. Millan. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pages 233–240. IEEE, June 2010.

[151] Natalia Romero, AgnieszkaMatysiak Szóstek, Maurits Kaptein, and Panos Markopoulos. Behaviours and preferences when coordinating mediated interruptions: Social and system influence. In LiamJ. Bannon, Ina Wagner, Carl Gutwin, RichardH.R. Harper, and Kjeld Schmidt, editors, *ECSCW 2007 SE - 19*, pages 351–370. Springer London, 2007.

[152] Mary E Rosenberger, William L Haskell, Fahd Albinali, Selene Mota, Jason Nawyn, and Stephen Intille. Estimating activity and sedentary behavior from an accelerometer on the hip or wrist. *Medicine and science in sports and exercise*, 45(5):964–75, May 2013.

[153] Stephanie Rosenthal, AnindK. Dey, and Manuela Veloso. Using Decision-Theoretic Experience Sampling to Build Personalized Mobile Phone Interruption Models. In Kent Lyons, Jeffrey Hightower, and ElaineM. Huang, editors, *Pervasive Computing SE - 11*, volume 6696 of *Lecture Notes in Computer Science*, pages 170–187. Springer Berlin Heidelberg, 2011.

[154] Steven L. Salzberg. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, 16(3):235–240, September 1994.

[155] A. Schmidt and K. van Laerhoven. How to build smart appliances? *IEEE Personal Communications*, 8(4):66–71, August 2001.

[156] Albrecht Schmidt, Michael Beigl, and Hans-W Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

[157] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, July 1998.

[158] Nicu Sebe, Michael Lew, and Thomas S. Huang, editors. *Computer Vision in Human-Computer Interaction*, volume 3766 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[159] Shimmer. Wearable Sensor Technology — Shimmer — Wearable Wireless Sensing Technology and Solutions.

[160] Dorra Ben Ayed Mezghanni Siwar Zribi Boujelbene. Improving SVM by Modifying Kernel Functions for Speaker Identification Task. *JDCTA*, 4:100 – 105, 2010.

[161] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.

[162] Maja Stikic, Tam Huynh, Kristof Van Laerhoven, and Bernt Schiele. ADL recognition based on the combination of RFID and accelerometer sensing. In *2008 Second International Conference on Pervasive Computing Technologies for Healthcare*, pages 258–263. IEEE, January 2008.

[163] Maja Stikic, Tam Huynh, Kristof Van Laerhoven, and Bernt Schiele. ADL recognition based on the combination of RFID and accelerometer sensing. In *2008 Second International Conference on Pervasive Computing Technologies for Healthcare*, pages 258–263. IEEE, January 2008.

[164] Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele. Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2521–2537, February 2011.

[165] Maja Stikic and Bernt Schiele. Activity Recognition from Sparsely Labeled Data Using Multi-Instance Learning. In Tanzeem Choudhury, Aaron Quigley, Thomas Strang, and Koji Suginuma, editors, *Location and Context Awareness SE - 10*, volume 5561 of *Lecture Notes in Computer Science*, pages 156–173. Springer Berlin Heidelberg, 2009.

[166] A. Reiss Stricker and D. PAMAP2 Physical Activity Monitoring Data Set.

[167] SupperBetter Labs. Superbetter: https://www.superbetter.com/.

[168] Sweden Statistics. HETUS: Harmonised European Time Use Survey - Start page.

[169] Emmanuel Munguia Tapia, Stephen S. Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 1–4. IEEE, October 2007.

[170] EmmanuelMunguia Tapia, Tanzeem Choudhury, and Matthai Philipose. Building Reliable Activity Models Using Hierarchical Shrinkage and Mined Ontology. In KennethP. Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley, editors, *Pervasive Computing SE - 2*, volume 3968 of *Lecture Notes in Computer Science*, pages 17–32. Springer Berlin Heidelberg, 2006.

[171] EmmanuelMunguia Tapia, StephenS. Intille, and Kent Larson. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing SE - 10*, volume 3001 of *Lecture Notes in Computer Science*, pages 158–175. Springer Berlin Heidelberg, 2004.

[172] EmmanuelMunguia Tapia, StephenS. Intille, Louis Lopez, and Kent Larson. The Design of a Portable Kit of Wireless Sensors for Naturalistic Data Collection. In KennethP. Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley, editors, *Pervasive Computing SE - 8*, volume 3968 of *Lecture Notes in Computer Science*, pages 117–134. Springer Berlin Heidelberg, 2006.

[173] Rosalind Picard Teresa Marrin. Analysis of Affective Musical Expression with the Conductor's Jacket. Teresa Mar.

[174] Yasuko Matsui Tomomi Matsui. A Survey of Algorithms for Calculating Power Indices of Weighted Majority Games.

[175] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, June 2011.

[176] TRIVISIO. Trivisio Prototyping GmbH development and product website.

[177] P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine Recognition of Human Activities: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, November 2008.

[178] UCI. The UC Irvine Machine Learning Repository.

[179] H. Bülthoff C. Burges V. Vapnik T. Vetter V. Blanz, B. Schölkopf. Comparison of view-based object recognition algorithms using realistic 3D models.

[180] K. Van Laerhoven and H.-W. Gellersen. Spine versus Porcupine: A Study in Distributed Wearable Activity Recognition. In *Eighth International Symposium on Wearable Computers*, volume 1, pages 142–149. IEEE.

[181] Kristof Van Laerhoven. Porcupines: fine grained activity monitoring in psychiatry using accelerometer sensors. *Annals of General Psychiatry*, 9(Suppl 1):S13, 2010.

[182] Kristof Van Laerhoven and Kofi Aidoo. Teaching Context to Applications. *Personal and Ubiquitous Computing*, 5(1):46–49, February 2001.

[183] Kristof Van Laerhoven, Marko Borazio, David Kilian, and Bernt Schiele. Sustained logging and discrimination of sleep postures with low-level, wrist-worn sensors. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 69–76. IEEE, September 2008.

[184] Kristof Van Laerhoven, David Kilian, and Bernt Schiele. Using rhythm awareness in long-term activity recognition. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 63–66. IEEE, 2008.

[185] Alexander Vezhnevets and Joachim M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3249–3256. IEEE, June 2010.

[186] VILIV. Viliv S5. Product Website.

[187] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[188] Ladd Wheeler and Harry T. Reis. Self-Recording of Everyday Life Events: Origins, Types, and Uses. *Journal of Personality*, 59(3):339–354, September 1991.

[189] Daniel H. Wilson, Danny Wyatt, and Matthai Philipose. Using context history for data collection in the home. *COGNITIVE SCIENCE RESEARCH PAPER - UNIVERSITY OF SUSSEX CSRP*, 577(17), 2005.

[190] Wockets Team. Data Collection Protocol for SWAP Study.

[191] Wockets Team. Wockets - Getting Started Web Page.

[192] Wockets Team. Wockets Hardware Design.

[193] Wockets Team. Wockets Software Architecture.

[194] Wockets Team. Wockets Source Code.

[195] Wockets Team. Wockets website at wikispaces.

[196] ChristopherR. Wren and EmmanuelMunguia Tapia. Toward Scalable Activity Recognition for Sensor Networks. In Mike Hazas, John Krumm, and Thomas Strang, editors, *Location- and Context-Awareness SE - 12*, volume 3987 of *Lecture Notes in Computer Science*, pages 168–185. Springer Berlin Heidelberg, 2006.

[197] Rongze Xia, Yi Han, Yan Jia, and Hu Li. Mining users' activity on large Twitter text data. In *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service - ICIMCS '13*, page 214, New York, New York, USA, August 2013. ACM Press.

[198] Thomas Hofmann Yasemin Altun, Ioannis Tsochantaridis. Hidden Markov Support Vector Machines.

[199] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking. *ACM Computing Surveys*, 38(4):13–es, December 2006.

[200] Zephyr Technology. Product Webpage.

[201] Zhengyou Zhang. Microsoft Kinect Sensor and Its Effect. *IEEE Multimedia*, 19(2):4–10, February 2012.

[202] Zhongtang Zhao, Yiqiang Chen, Junfa Liu, Zhiqi Shen, and Mingjie Liu. Cross-people mobile-phone based activity recognition. pages 2545–2550, July 2011.

[203] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th international conference on World wide web - WWW '09*, page 791, New York, New York, USA, April 2009. ACM Press.

[204] Andreas Zinnen, Kristof Laerhoven, and Bernt Schiele. Toward Recognition of Short and Non-repetitive Activities from Wearable Sensors. In Bernt Schiele, AnindK. Dey, Hans Gellersen, Boris Ruyter, Manfred Tscheligi, Reiner Wichert, Emile Aarts, and Alejandro Buchmann, editors, *Ambient Intelligence SE - 9*, volume 4794 of *Lecture Notes in Computer Science*, pages 142–158. Springer Berlin Heidelberg, 2007.