

ADVERTIMENT. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (www.tesisenxarxa.net) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (www.tesisenred.net) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (www.tesisenxarxa.net) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author

Smartphone-Based Human Activity Recognition



Jorge Luis Reyes Ortiz

Supervisors:

Xavier Parra
Davide Anguita

Centre d'Estudis Tecnològics per a l'atenció a la Dependència i la Vida Autònoma -
CETpD

Universitat Politècnica de Catalunya

and

Dipartimento di Ingegneria Navale Elettrica Elettronica e delle Telecomunicazioni -
DITEN

Università degli Studi di Genova

A thesis submitted for the degree of

Doctor of Philosophy in Interactive and Cognitive Environments

Wednesday 2nd July, 2014



ICE PhD Acknowledgements

This PhD Thesis has been developed in the framework of, and according to, the rules of the Erasmus Mundus Joint Doctorate on Interactive and Cognitive Environments EMJD ICE [FPA n° 2010-0012] with the cooperation of the following Universities:



Alpen-Adria-Universität Klagenfurt – AAU



Queen Mary, University of London – QML



Technische Universiteit Eindhoven – TU/e



UNIVERSITÀ DEGLI STUDI
DI GENOVA

Università degli Studi di Genova – UNIGE



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Universitat Politècnica Catalunya – UPC

According to ICE regulations, the Italian PhD title has also been awarded by the Università degli Studi di Genova.

... a mi familia, el motor de mi vida.

Acknowledgements

First of all, I would like to thank my supervisors Xavier Parra and Davide Anguita for their continuous and invaluable support throughout the course of this work. Their constructive ideas have been really helpful in order to achieve the objectives of my research.

Infinite thanks to the group of researchers who have surrounded me during the last three and a half years. They have provided me with significant insights not only on the technical areas, but also about life. Luca Oneto, Albert Samà, Alessandro Ghio, Boris Takač, Sophia Bano, Isah Lawal, and Daniel Rodriguez are just a few of them. Moreover, I would like to thank my research centers: the CETpD and the Smartlab, and all their members which have welcome me and made me feel like if I was at home.

I also want to express my gratitude to my close friends: Lorena, David, Emmanuel, Jordi, Sandhya, Craig, Carlos, Kreš, and Maira which have been always there backing me up. Finally and most importantly, I want to thank my family: my father Jorge, my mother Elsa, and my sisters Alba and Diana. Their unconditional support has been immense even though they are so far away.

Smartphone-Based Human Activity Recognition

Abstract

Human Activity Recognition (HAR) is a multidisciplinary research field that aims to gather data regarding people's behavior and their interaction with the environment in order to deliver valuable context-aware information. It has nowadays contributed to develop human-centered areas of study such as Ambient Intelligence and Ambient Assisted Living, which concentrate on the improvement of people's Quality of Life.

The first stage to accomplish HAR requires to make observations from ambient or wearable sensor technologies. However, in the second case, the search for pervasive, unobtrusive, low-powered, and low-cost devices for achieving this challenging task still has not been fully addressed. In this thesis, we explore the use of smartphones as an alternative approach for performing the identification of physical activities. These self-contained devices, which are widely available in the market, are provided with embedded sensors, powerful computing capabilities and wireless communication technologies that make them highly suitable for this application.

This work presents a series of contributions regarding the development of HAR systems with smartphones. In the first place we propose a fully operational system that recognizes in real-time six physical activities while also takes into account the effects of postural transitions that may occur between them. For achieving this, we cover some research topics from signal processing and feature selection of inertial data, to Machine Learning approaches for classification. We employ two sensors (the accelerometer and the gyroscope) for collecting inertial data. Their raw signals are the input of the system and are conditioned through filtering in order to reduce noise and allow the extraction of informative activity features. We also emphasize on the study of Support Vector Machines (SVMs), which are one of the state-of-the-art Machine Learning techniques for classification, and reformulate various of the standard multiclass linear and non-linear methods to find the best trade off between recognition performance, computational costs and energy requirements, which are essential aspects in battery-operated devices such as smartphones. In particular, we propose two multiclass SVMs for activity classification: one linear algorithm which allows to control over dimensionality reduction and system accuracy; and also a non-linear hardware-friendly algorithm that only uses fixed-point arithmetic in the prediction phase and enables a model complexity reduction while maintaining the system performance.

The efficiency of the proposed system is verified through extensive experimentation over a HAR dataset which we have generated and made publicly available. It is composed of inertial data collected from a group of 30 participants which performed a set of common daily activities while carrying a smartphone as a wearable device.

The results achieved in this research show that it is possible to perform HAR in real-time with a precision near 97% with smartphones. In this way, we can employ the proposed methodology in several higher-level applications that require HAR such as ambulatory monitoring of the

disabled and the elderly during periods above five days without the need of a battery recharge. Moreover, the proposed algorithms can be adapted to other commercial wearable devices recently introduced in the market (e.g. smartwatches, phablets, and glasses). This will open up new opportunities for developing practical and innovative HAR applications.

Reconocimiento de Actividades Humanas Basada en Smartphones

Resumen

El Reconocimiento de Actividades Humanas (RAH) es un campo de investigación multidisciplinario que busca recopilar información sobre el comportamiento de las personas y su interacción con el entorno con el propósito de ofrecer información contextual de alta significancia sobre las acciones que ellas realizan. Recientemente, el RAH ha contribuido en el desarrollo de áreas de estudio enfocadas a la mejora de la calidad de vida del hombre tales como: la inteligencia ambiental (Ambient Intelligence) y la vida cotidiana asistida por el entorno para personas dependientes (Ambient Assisted Living).

El primer paso para conseguir el RAH consiste en realizar observaciones mediante el uso de sensores fijos localizados en el ambiente, o bien portátiles incorporados de forma vestible en el cuerpo humano. Sin embargo, para el segundo caso, aún se dificulta encontrar dispositivos poco invasivos, de bajo consumo energético, que permitan ser llevados a cualquier lugar, y de bajo costo. En esta tesis, nosotros exploramos el uso de teléfonos móviles inteligentes (Smartphones) como una alternativa para el RAH. Estos dispositivos, de uso cotidiano y fácilmente asequibles en el mercado, están dotados de sensores embebidos, potentes capacidades de cómputo y diversas tecnologías de comunicación inalámbrica que los hacen apropiados para esta aplicación.

Nuestro trabajo presenta una serie de contribuciones en relación al desarrollo de sistemas para el RAH con Smartphones. En primera instancia proponemos un sistema que permite la detección de seis actividades físicas en tiempo real y que, además, tiene en cuenta las transiciones posturales que puedan ocurrir entre ellas. Con este fin, hemos contribuido en distintos ámbitos que van desde el procesamiento de señales y la selección de características, hasta algoritmos de Aprendizaje Automático (AA). Nosotros utilizamos dos sensores inerciales (el acelerómetro y el giroscopio) para la captura de las señales de movimiento de los usuarios. Estas han de ser procesadas a través de técnicas de filtrado para la reducción de ruido, segmentación y obtención de características relevantes en la detección de actividad. También hacemos énfasis en el estudio de Máquinas de soporte vectorial (MSV) que son uno de los algoritmos de AA más usados en la actualidad. Para ello reformulamos varios de sus métodos estándar (lineales y no lineales) con el propósito de encontrar la mejor combinación de variables que garanticen un buen desempeño del sistema en cuanto a precisión, coste computacional y requerimientos de energía, los cuales son aspectos esenciales en dispositivos portátiles con suministro de energía mediante baterías. En concreto, proponemos dos MSV multiclase para la clasificación de actividad: un algoritmo lineal que permite el balance entre la reducción de la dimensionalidad y la precisión del sistema; y asimismo presentamos un algoritmo no lineal conveniente para dispositivos con limitaciones de hardware que solo utiliza aritmética de punto fijo en la fase de predicción y que permite reducir la complejidad del modelo de aprendizaje mientras mantiene el rendimiento del sistema.

La eficacia del sistema propuesto es verificada a través de una experimentación extensiva sobre la base de datos RAH que hemos generado y hecho pública en la red. Esta contiene la información inercial obtenida de un grupo de 30 participantes que realizaron una serie de actividades de la vida cotidiana en un ambiente controlado mientras tenían sujeto a su cintura un smartphone que capturaba su movimiento.

Los resultados obtenidos en esta investigación demuestran que es posible realizar el RAH en tiempo real con una precisión cercana al 97% con smartphones. De esta manera, podemos emplear la metodología propuesta en aplicaciones de alto nivel que requieran el RAH tales como monitorizaciones ambulatorias para personas dependientes (ej. ancianos o discapacitados) durante periodos mayores a cinco días sin la necesidad de recarga de baterías. Adicionalmente, los algoritmos propuestos se pueden adaptar a otros dispositivos vestibles recientemente introducidos en el mercado (ej. relojes y gafas inteligentes, y phablets). Esto permitirá abrir nuevas oportunidades para el desarrollo de aplicaciones para el RAH más prácticas e innovadoras.

Contents

Abstract	ix
Resumen	xi
Publications	xvii
Mathematical Symbols and Notation	xix
1 Introduction	1
1.1 Motivation	1
1.2 Main Contributions	2
1.3 Thesis Outline	3
I Human Activity Recognition Essentials	5
2 Background	7
2.1 Introduction	7
2.2 Ambient Intelligence and Assisted Living	7
2.2.1 AmI Systems	8
2.2.2 AmI Applications	9
2.2.3 Ambient Assisted Living	10
2.3 Sensing Activity	11
2.3.1 Ambient Sensors	11
2.3.2 Wearable Sensors	12
2.4 Smartphones: The Evolution of the Traditional Mobile Phone	14
2.4.1 Characteristics and Selection Criteria	15
2.4.2 Smartphones as Wearable Sensors	16
2.4.3 Opportunistic Sensing	17
2.5 Machine Learning	18
2.5.1 Taxonomy of ML Algorithms	18
2.5.2 Machine Learning Approaches	20
2.5.3 Support Vector Machines	21
2.5.4 Performance Evaluation	27
2.6 Summary	29

3	State of the Art	31
3.1	Introduction	31
3.2	Human Activity Recognition Overview	32
3.2.1	Human Activities	33
3.2.2	Sensing and Data Collection	34
3.2.3	Feature Selection and Extraction	35
3.2.4	Machine Learning	35
3.3	Related work in HAR Systems	36
3.3.1	Human Activity Type	38
3.3.2	Sensor type and Smartphones	40
3.3.3	Machine Learning	41
3.3.4	Offline and Online	42
3.4	Summary	43
II	Data Collection and Offline Activity Recognition	45
4	Human Activity Dataset Generation	47
4.1	Introduction	47
4.2	Experimental Data Collection	47
4.2.1	Smartphone Selection	48
4.2.2	HAR Protocol	51
4.2.3	Data recording	51
4.3	HAR Data Processing	54
4.3.1	Labeling	54
4.3.2	Signal Processing	55
4.3.3	Feature Mapping and Dataset Generation	58
4.4	Results	61
4.4.1	Dataset Validation	61
4.4.2	HAR Competition	63
4.5	Summary	64
5	Hardware-Friendly Activity Recognition with Fixed-Point Arithmetic	65
5.1	Introduction	65
5.2	The Hardware-Friendly Multiclass SVM	66
5.2.1	The Binary Hardware-Friendly Formulation	66
5.2.2	Generalization to Multiclass SVM with probability estimates	68
5.3	HF-SVM and Statistical Learning Theory	69
5.4	Results	71
5.4.1	System performance using Fixed-Point Arithmetic	72
5.4.2	Processing Time and Battery Consumption	74
5.5	Summary	76

III	Online Activity Recognition with Smartphones	77
6	Linear SVM Models for Online Activity Recognition	79
6.1	Introduction	79
6.2	L1-Norm and L2-Norm SVMs for Activity Recognition	80
6.3	L1-L2 SVM Algorithm	82
6.3.1	L1-L2 SVM Formulation	83
6.3.2	Extended Algorithm for solving L1-L2 SVMs	83
6.4	Results	84
6.4.1	Linear vs. Non-Linear SVMs	86
6.4.2	Selection of subsets of features	87
6.4.3	Dimensionality reduction with L1-SVM	88
6.4.4	L1-L2 SVM with HAR data	90
6.5	Summary	94
7	Online Recognition with Postural Transition Awareness	97
7.1	Introduction	97
7.2	Postural Transitions in HAR Systems	98
7.3	HAR with Postural Transitions Awareness	99
7.3.1	Signal Conditioning and Feature Extraction	100
7.3.2	Implementation of the SVM Feed Forward Phase	102
7.3.3	Temporal Activity Filtering	102
7.4	PTA-HAR Experiments	106
7.4.1	The HAR Dataset with Postural Transitions	106
7.4.2	Online System Error Estimation and Performance Evaluation	107
7.4.3	HARApp: The Android App for HAR	109
7.5	Results	110
7.5.1	System Error Evaluation	110
7.5.2	Activity Classification Performance	111
7.6	Summary	113
8	Conclusions	115
8.1	Achievements	115
8.2	Future Work	116
	Bibliography	119
	Glossary	133

Publications

Journal Papers

- Jorge-Luis Reyes-Ortiz, Luca Oneto, Samà Albert, Alessandro Ghio, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Personal and Ubiquitous Computing*, (Waiting for Acceptance):–, 2014c
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic. *Journal of Universal Computer Science*, 19:1295–1314, 2013b

Conference Papers

- Jorge-Luis Reyes-Ortiz, Luca Oneto, Samà Albert, Alessandro Ghio, Xavier Parra, and Davide Anguita. Human activity recognition on smartphones with awareness of basic activities and postural transitions. In *International Conference on Artificial Neural Networks*, 2014b
- Davide Anguita, Alessandro Ghio, Luca Oneto, Jorge-Luis Reyes-Ortiz, and Sandro Ridella. A novel procedure for training l1-l2 support vector machine classifiers. In *International Conference on Artificial Neural Networks*, 2013e
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. Training computationally efficient smartphone-based human activity recognition models. In *International Conference on Artificial Neural Networks*, 2013d
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier. Parra, and Jorge-Luis. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013c
- Jorge-Luis Reyes-Ortiz, Alessandro Ghio, Davide Anguita, Xavier Parra, Joan Cabestany, and A Catal. Human activity and motion disorder recognition: Towards smarter interactive cognitive environments. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013b

- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient Assisted Living and Home Care*, 2012b

Miscellaneous

- Jorge-Luis Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Recognition of basic activities and postural transitions using smartphones data set. <http://www.har.smartlab.ws>, 2014a
- Jorge-Luis Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human activity recognition using smartphones data set. <http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>, 2013a
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. Human activity recognition on smartphones for mobile context awareness. In *Neural Information Processing Systems. Workshop on Machine Learning Approaches to Mobile Context Awareness*, 2012c

Mathematical Symbols and Notation

Table 1: Most commonly used mathematical symbols in this thesis

Symbol	Description
$\mathbf{x} \in \mathbb{R}^d$	Input sample of d dimensions
$X \in \mathbb{R}^{n \times d}$	Set of n vector samples of dimension d
$x_{i,j}$	j -th feature of the i -th sample
$y \in \{-1, 1\}$	output target in binary classification problems
$y \in \{1, \dots, m\}$	output target in m -class problems with $m \in \mathbb{N}_1 > 2$
$\mathbf{y} \in \mathbb{Z}^n$	Vector of n output targets y_i
$f_c(\mathbf{x}) \in \mathbb{R}$	SVM output of a sample for class c
$\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$	SVM output vector for a m -class problem
$p_c(\mathbf{x}) \in [0, 1] \subset \mathbb{R}$	SVM probability output
$\mathbf{p}(\mathbf{x}) \in \mathbb{R}^m$	SVM probability output vector
$c^* \in \{1, \dots, m\}$	SVM predicted class
$\mathbf{a}(t) \in \mathbb{R}^3$	Time-domain triaxial body acceleration signal
$\boldsymbol{\omega}(t) \in \mathbb{R}^3$	Time-domain triaxial angular velocity signal
$\mathbf{g}(t) \in \mathbb{R}^3$	Time-domain gravity signal
T	Window sampling period in seconds
N	Number of samples in an activity window
k	number of bits for the fixed-point number representation

Table 2: Notation of the presented SVM Machine Learning algorithms

SVM Algorithm				
Name	Binary	Multiclass	Norm	$K(x_i, x_j)$
<i>Linear</i>				
L1 Manhattan	L1-SVM	MC-L1-SVM	$\ \mathbf{w}\ _1$	$\mathbf{x}^T \mathbf{x}$
L2 Euclidean	L2-SVM	MC-L2-SVM	$\ \mathbf{w}\ _2$	$\mathbf{x}^T \mathbf{x}$
L1-L2	L1-L2-SVM	MC-L1-L2-SVM	$\ \mathbf{w}\ _1, \ \mathbf{w}\ _2$	$\mathbf{x}^T \mathbf{x}$
<i>Non-Linear</i>				
Gaussian	GK-SVM	MC-GK-SVM	$\ \mathbf{w}\ _2$	$e^{-\gamma \ \mathbf{x}_i - \mathbf{x}_j\ _2^2}$
Laplacian	LK-SVM	MC-LK-SVM	$\ \mathbf{w}\ _2$	$e^{-\gamma \ \mathbf{x}_i - \mathbf{x}_j\ _1}$
Hardware-Friendly	HF-SVM	MC-HF-SVM	$\ \mathbf{w}\ _2$	$2^{-\gamma \ \mathbf{x}_i - \mathbf{x}_j\ _1}$

Table 3: Notation of the Proposed HAR systems

HAR System						
Name	Notation	ML Algorithm	Activities	Dataset	Online	Chap.
Hardware Friendly	HF-HAR	MC-HF-SVM	6BAs	\mathcal{D}_1	-	5
Linear	L-HAR	MC-L1-SVM MC-L2-SVM MC-L1-L2-SVM	6BAs	$\mathcal{D}_2, \mathcal{D}_{2T}$	✓	6
Postural Transition Aware	PTA-HAR	MC-L1-SVM	6BAs, 6BAs+PTs	\mathcal{D}_{3T}	✓	7

Chapter 1

Introduction

1.1 Motivation

In the last years, recent advances in computing and sensing technologies have contributed with novel ideas aiming to solve people’s needs in a wide range of situations: from basic daily living essentials such as personal care, feeding and mobility, to more complex social issues including health, education and security. Automated systems are an example of the materialization of these ideas. They exploit information gathered from users and their environment in order to produce an appropriate action [Campbell et al., 2008]. In this thesis, we are interested in exploring the development of systems that promote the improvement of people’s Quality of Life (QoL) through the *recognition of human activities*, especially, that of individuals with any type of limitation (e.g. the disabled and the elderly) and lack of general well-being.

Understanding people’s actions and their interaction with the environment is a key element for the development of the aforementioned intelligent systems. Human Activity Recognition (HAR) is a research field that specifically deals with this issue through the integration of sensing and reasoning, in order to deliver context-aware data that can be employed to provide personalized support in many applications [Chen et al., 2012a]. As a simple example, imagine a *smart home* equipped with ambient sensors able to detect people’s presence and the activation of household appliances. It is possible to infer the activities performed by its residents based on the sensors signals along with other relevant aspects such as time of the day and date (e.g. a person in the kitchen during morning time while a coffee machine is on suggests that person is making breakfast). Consequently, the collected HAR information can be exploited to anticipate future people requirements and become responsive to them (e.g. by automatically pre-heating the coffee machine, controlling room lighting and temperature, etc.).

In the HAR framework, there are still several issues that need to be addressed, some of which are: obtrusiveness of current wearable sensors; lack of fully pervasive systems able to reach users at any location any time; privacy concerns regarding invasive and continuous monitoring of activities (e.g. by using video cameras); difficulty of performing HAR in real-time; battery limitations of wearable devices; and dealing with content extraction from sparse multisensor

data.

In this thesis, we explore the feasibility of using *smartphones* to perform the automatic recognition of physical activities while also addressing some of these current HAR limitations. These modern devices, which are a new generation of mobile phones, are provided with enhanced computing capabilities and embedded sensors that make them highly suitable for their application in HAR. We aim to provide a technological tool that can be employed as an activity information resource to other systems for better decision making (e.g. for the remote monitoring of elderly patients who live alone or without permanent caretaking and the measurement of their personal autonomy).

The development of a smartphone-based HAR system introduces new challenges linked to the incorporation of the recognition system on everyday use devices. In this work, we propose various approaches in order to recognize, in real-time, human activity using the smartphone inertial sensors (*accelerometer* and *gyroscope*) in conjunction with the implementation of supervised Machine Learning (ML) algorithms (specifically Support Vector Machine (SVM)) on the device. We focus particularly in the study of a group of six locomotion Basic Activities (BAs): *standing*, *sitting*, *lying-down*, *walking*, *walking downstairs* and *walking upstairs*; and also examine six Postural Transitions (PTs) that occur between its static postures: *stand-to-sit*, *sit-to-stand*, *sit-to-lie*, *lie-to-sit*, *stand-to-lie* and *lie-to-stand*. Sometimes these PTs are disregarded in HAR systems [Lara and Labrador, 2012a]. But they are significant for certain applications where their incidence is high and overall duration is comparable to other activities. For example, in the design of activity monitoring systems for the disabled during rehabilitation practices. The study of these PTs is also a central element of our research.

This thesis addresses the following questions: i) How can we perform HAR using existing smartphones? ii) How to exploit inertial sensors (accelerometer and gyroscope) to develop smartphone-based HAR systems? iii) Which ML algorithms are suitable for efficient HAR implementations in battery-limited smartphones? iv) How to achieve real-time HAR using smartphones?

1.2 Main Contributions

The main contributions of this thesis are presented as follows:

- We propose a smartphone-based HAR system for the online recognition of human activities from inertial data (PTA-HAR). It consists of the combination of four elements: the device-embedded motion sensors, a signal processing unit for feature extraction, a multiclass linear SVM algorithm and an activity filtering module for dealing with recurrent PTs. We demonstrate the system operation in real-time and show the improvements achieved when considering the detection of postural transitions (Chapter 7).
- We propose an ML algorithm for activity classification based on a one-vs-all (OVA) SVM with L1- and L2-Norm regularization. Its advantage relies in its faster prediction when

compared against non-linear SVM approaches while also allowing to fine tune the trade-off between dimensionality reduction and classification accuracy. At the same time, we present a novel unified approach for training the three possible instances of this algorithm: L1-SVM, L2-SVM, L1-L2-SVM. Its significance relies on its flexibility and suitability to be implemented with well-known solvers (Chapter 6).

- We propose the first hardware-friendly SVM based on fixed-point arithmetic for the prediction of human activities (MC-HF-SVM). This approach allows to vary the fixed-point number representation (number of bits) to control over model accuracy and complexity, leading to improvements in terms of recognition accuracy, speed and battery energy sparing with respect to conventional floating-point based formulations (Chapter 5).
- We have generated and made publicly available a HAR dataset [Reyes-Ortiz et al., 2013a] composed of trials performed with a group of 30 participants which performed a set of common daily activities while carrying a smartphone as a wearable device. The dataset provides a collection of 10299 labeled activity instances which include the raw inertial data from the smartphone accelerometer and gyroscope along with the extracted activity features (Chapter 4).

1.3 Thesis Outline

This thesis focuses on the design and implementation of smartphone-based HAR systems. It has been divided in three main parts. Part I covers essential aspects on the topic of HAR. First, a series of fundamental concepts required to contextualize our research problem are portrayed. These are followed by an introduction to HAR accompanied with its most influential works which are contrasted and discussed. Part II concentrates on data collection and offline HAR. It presents the procedure for the generation of the HAR dataset required for this research and also introduces the first offline HAR method based on fixed-point arithmetic. Finally, Part III pays particular attention in the description of the methods developed for the implementation of the online HAR systems and concludes with a summarization of all the achievements of this work.

The remaining thesis chapters are briefly described here:

- **Chapter 2** describes the main ideas about the areas of study relevant to the development of HAR systems in order to develop a global perspective of our research problem. These areas are divided in two groups: regarding our framework context (Ambient Intelligence (AmI), Ambient Assisted Living (AAL)) and implementation mechanisms (sensors, smartphones and ML with emphasis on SVMs).
- **Chapter 3** examines the current state of the art on the subject of HAR. It starts with a general introduction regarding the HAR pipeline and then focuses on various already implemented HAR systems relevant to our research. It also highlights particular aspects

of these systems such as sensing technologies, types of activities, ML approaches and real-time computing.

- **Chapter 4** presents the collection of stages required for the acquisition of the experimental HAR data used in this thesis. It includes aspects such as smartphone selection, trials with volunteers, signal conditioning and feature selection. It also describes the procedures concerning the dataset validation; internally through experimentation and externally via a HAR contest in which other research groups were encouraged to propose novel solutions to the recognition problem.
- **Chapter 5** explains the proposed hardware-friendly HAR system (HF-HAR) and its core ML algorithm based on fixed point arithmetic (MC-HF-SVM). Initially motivated by current limitations of mobile devices regarding battery life, this method aims to predict activities relying on a modified SVM formulation with adjustable fixed-point number representation. In this way, we show through experiments how system accuracy levels can be maintained while increasing the speed during the prediction of activities. This is explained under the light of Statistical Learning Theory (SLT) which shows how this implementation brings advantages with respect to the generalization ability of the algorithm.
- **Chapter 6** studies linear SVM algorithms and its application to an online system for the recognition of activities on smartphones (L-HAR). The algorithms differ on the norm of their formulation's regularization term (whether it is the L1-, L2- or L1-L2-Norm). They allow to control over dimensionality reduction and classification accuracy while increasing the prediction speed when compared with kernelized SVM algorithms. Moreover, this chapter presents a novel approach for training these classifiers (EX-SMO) with minimal effort using well-known solvers. To conclude, the benefits of adding smartphones gyroscope into the recognition system are presented along with another feature selection mechanisms that use subsets of features in the time and frequency domain.
- **Chapter 7** introduces an online HAR system for the classification of human activities using smartphones which deals with recurring postural transitions while sequences of activities are carried out (PTA-HAR). For its implementation, the linear SVMs presented in Chapter 6 are combined with temporal filters that use activity probability estimations within a limited time window. The benefits of these approaches are presented through experimentation with the HAR dataset and compared against the previously presented HAR systems.
- **Chapter 8** summarizes the accomplishments of this work and also proposes future research directions

Part I

Human Activity Recognition Essentials

Chapter 2

Background

2.1 Introduction

In this chapter, we explore four areas or study directly related with HAR which are essential to develop a global perspective of our research problem. They explain the context where our proposed HAR systems can be applied to and also the mechanisms through which it is possible to implement them.

Firstly, we introduce AmI and AAL as the frameworks where we focus our research on HAR (Section 2.2). In particular, we target the detection of physical activities to provide context information for daily living applications, predominantly, for people with motor impairments or activity limitations. Following this, we concentrate on the requirements to implement HAR systems. From the hardware aspect, we give an overview of the sensors used for the recognition of activities in Section 2.3 and also present the smartphone as it is our selected device for HAR (Section 2.4). Then, from the software and mathematical standpoint, we give an overview of ML with emphasis on SVMs (Section 2.5). Finally, we summarize the chapter in Section 2.6.

2.2 Ambient Intelligence and Assisted Living

AmI is the paradigm that explores environments that behave intelligently and adaptively based on the actions of the people that are part of it. These environments are aimed to respond, in a cooperative manner, to their needs which are identified through the retrieval of context information gathered by means of the available technology. This process stimulates the generation of a dynamic human-machine interaction with the purpose of providing a better QoL to people, expressed, for instance, via more efficient ways of solving tasks, faster assistance, and greater comfort [Aarts and Wichert, 2009]. In the last decade of the 20th century, the term Ubiquitous Computing (UbiComp) made its appearance [Issarny et al., 2005]. It considered the introduction of unobtrusive computing devices in the environment. Following this, AmI emerged when advances in network technologies (e.g. internet, wireless communications) were incorporated in the original idea [Abowd and Mynatt, 2000; Cook and Das, 2012].

The integration of new technologies in our everyday life is evolving very rapidly. The miniaturization, greater functionality, and lower manufacture costs of electronic devices and sensors are making a transformation in the way humans and machines interact. These technological changes are allowing the development of AmI services to seamlessly face user needs [Cook and Das, 2012]. Moreover, dealing with a growing population, the lack of sufficient human resources in some areas (e.g. caretakers, security forces, medical staff), and limited budgets are also important aspects which encourage people, communities and countries to make use of AmI for their advantage.

AmI can benefit all population sectors. However, some of them, such as people with limitations (e.g. the disabled and the elderly), have become a priority and need immediate attention. In consequence, there is an area of AmI which is specifically focused on the development of intelligent systems to assist these people: Ambient Assisted Living [Kleinberger et al., 2007]. This area is developing very rapidly in accordance to the current population explosion statistics of these age groups and their growing demands [Aarts and Wichert, 2009; Eur, 2011].

In this section we explore the principles and characteristics of these two areas of study and explore how they interact with each other. Finally, we connect these concepts with the development of HAR systems.

2.2.1 AmI Systems

AmI is still in its early stages but current technologies are facilitating its development and the introduction of new systems [Kleinberger et al., 2007]. These AmI systems exhibit common characteristics some of which are here described:

- They are *unobtrusive*: hardly or not perceived by its users. This avoids disturbances limiting users during the execution of their normal activities. For example, using smart wearable accessories (e.g. smartphones, watches and glasses) and devices embedded in clothes, nearby objects and furniture.
- They *communicate* between nodes. This involves fast and robust communication mechanisms, such as wireless technologies, to connect different devices (nodes) when the transmission of information over a distance is required.
- They are *ubiquitous*: available at any location, at any time. This characteristic considers the continuous delivery of services, through the use of either fixed infrastructure, mobile equipment or their combination, regardless of possible limitations linked to time and location.
- They are *context-aware*. They capture the state of the user and its environment by exploiting heterogeneous sensors in order to respond accordingly to diverse and changing conditions.

- They intelligently *anticipate* to our needs based on previous information (e.g. activity patterns, past events and their solutions). This is generally done through the use of ML techniques that take past experience to forecast the required actions for similar situations in the future.
- They allow *seamless human-machine interaction*. They are provided with mechanisms to naturally communicate with its users (e.g. by interpreting body motion, gestures, location and physiological signals). Ideally, without the user having to interact through custom-made devices.
- They are *adaptive*: able to learn and respond to new or unusual situations. This characteristic allows systems to become more robust and able to react against unexpected conditions from the users and the environment.

2.2.2 AmI Applications

AmI has been already employed in a wide range of applications. Below, three significant areas where it has been already applied are described:

- *Smart Homes*. Home automation is changing the way households and their residents interact. Novel households are arranged with multimodal technologies involving environmental sensors, user interfaces, computing devices and actuators; which aim to guarantee supervision and fast response to people in order to improve their QoL. Moreover, they are incorporating centralized controls for most of their basic services such as heating, lighting, security, entertainment, and, in general, all the available electronic appliances [García-Vázquez et al., 2010; José et al., 2010; Silva et al., 2012].
- *Vital Signs Monitoring for Healthcare*. Wearable body sensors [Yang and Yacoub, 2006] offer an alternative for monitoring people’s physiological and vital signs. These include, for example, measurements of heart rate, blood pressure, oxygen saturation, body motion, skin temperature and conductivity. AmI systems are capable of processing these data to supervise the health condition of a target group (e.g. the elderly). Moreover, this information can be communicated and distributed in real-time to interested parties such as family members and emergency services when urgent assistance is required.
- *Unifying heterogeneous operation interfaces*. It is nowadays common to see environments (e.g. living rooms, offices, conference rooms) composed of computing resources and electronic equipment for all kinds of purposes. These include: TVs, beamers, sound systems, lighting control systems, personal computers and portable devices such as tablets and smartphones. Even though they share the same space, their control mechanisms and interfaces are rather different and there is generally no unique approach to operate them. AmI contributes to unify the way these devices interact with people by creating more nat-

Area	2000-2005	2045-2050
Entire World	65.4	75.1
More Developed regions	75.6	82.1
Less developed regions	63.4	74.0

Table 2.1: Life expectancy at birth comparison between more and less developed world regions (in years). Data from [United Nations, 2006].

ural and simpler approaches to interface them. It also allows the automatic incorporation of new devices in a more intuitive way.

2.2.3 Ambient Assisted Living

In developed countries, it has been observed in the recent years a large growth in the life expectancy of their inhabitants. This trend is expected to expand to the whole world in the forthcoming years [United Nations, 2006] (Refer to Table 2.1). These figures show the proportion of elderly people is rapidly increasing and updating the distribution of age population groups. These changes have a dramatic impact in the way resources and roles need to be assigned to the various issues that affect the world.

Recent benchmarks in the population structure by main age groups in Europe have showed that by 2060 the elderly (namely people over 65 years) will be near 30% of its population as opposed to a 17% by 2010 [Eur, 2011]. These numbers represent an alarming growth of more than 70% of this age group which brings new challenges to deal with.

People are living longer, primarily due to the enormous advances in medicine such as the development of new drugs, treatments and equipment. These important achievements of humanity are driving us to find alternatives to cope with the large demand from the expanding aging population. One of the basic needs of the elderly arises from their places of residence. Many of them live in isolated conditions and some are distant from the city centers. This makes more difficult their access to services such as health and social care facilities. We can alleviate these issues through the exploitation of current technologies. For example, by developing automatic mechanisms for remote monitoring, diagnosis and assistance of people. AAL plays an important role in this matter as it is specifically focused on finding solutions to provide independent and healthy living to this population sector. This task can be supported by the adoption of technologies developed for AmI as these two areas are closely related.

AAL is not only limited to the elderly but also to people with any type of limitation (e.g. blindness, deafness, and physical disabilities). About 15% of the world population have some type of disability [WHO, 2011]. This figure shows that there is a large sector to which AAL can also bring numerous benefits.

AmI and AAL systems need to capture information from the environment and from the users. In this research, we have centered our efforts in finding mechanisms for obtaining activity information from the user by means of the analysis of their body motion. In particular, through the use of the nowadays widely used smartphones.

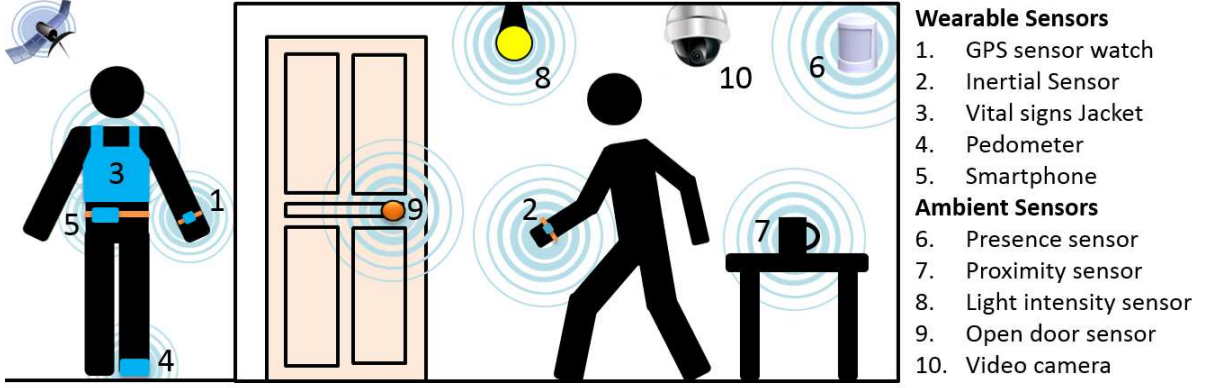


Figure 2.1: Examples of ambient and wearable devices.

The extraction of human activity information is only a small part of this large process of developing AmI systems. Therefore, we aim to provide a technological tool that can be employed as an input for applications in these two areas. This will provide systems with opportunities for better decision making not only to people with limitations but also to any kind of people which need to improve their well-being, health or comfort.

2.3 Sensing Activity

The right choice of sensors is one of the first elements to be taken into consideration for the design of HAR systems. A number of sensors have already been explored to extract activity-related information [Chen et al., 2012a; Lara and Labrador, 2012a]. They measure several attributes including *vital signs* (e.g. heart rate, body temperature, blood pressure), *environmental signals* (e.g. light intensity, temperature, sound levels), *motion* (e.g. acceleration, speed) and *positioning* (e.g. global and indoor location).

Based on the sensor placement with respect to the user, the sensing mechanisms are classified in: *ambient*, when sensors are in static locations of the environment and *wearable*, when they are worn or attached to the user's body. These two categories are detailed as follows. Besides, examples of sensors in these two categories are depicted in Figure 2.1.

2.3.1 Ambient Sensors

Ambient sensors, also known as *external* or *environmental* sensors, are the collection of devices found in the environment which measure physical properties regarding the surroundings, the users that are part of it, and their interaction. There is a wide range of ambient sensors such as microphones, video cameras, presence sensors, thermometers and depth sensors.

Many of these sensors have already been employed for HAR. For example, video cameras are employed in [Poppe, 2007] for markerless vision-based human motion analysis. Also in [Bian et al., 2005], microphones are used for sound source localization in a home environment for

communication activity inference. Even everyday use devices can also indicate the occurrence of certain activities such as the identification of switched on/off household appliances to infer home activities [Ogawa et al., 2002]. Moreover, in [Takač et al., 2013], a Microsoft Kinect’s depth sensor is used as an ambient sensor for position and orientation tracking for an indoor monitoring system for Parkinson’s disease (PD) patients. Although these examples can provide accurate context information about the agent’s motion and localization, they require a static infrastructure which limits its range of operation to a constrained space.

Additionally, video-based systems can be very effective for HAR but they are somewhat disadvantageous due to their demanding computations (e.g. for achieving real-time operation) and privacy concerns, for instance, when they are used in home environments as people are generally uncomfortable about being continuously monitored.

2.3.2 Wearable Sensors

Wearable sensors are used to gather signals directly from users. They are commonly attached to different body parts such as the waist, wrist, chest, legs and head [Ravi et al., 2005] but also fit to clothes and embedded in other accessories of regular use such as watches, glasses or mobile phones [Brezmes et al., 2009]. They contain a battery unit that provides the energy supply for continuous operation and, some of them, also have a wireless unit for sensor data transmission when it is externally needed or to interface them with other body-worn devices. Physiological and motion signals obtained from wearable sensors are highly informative for HAR. Skin temperature, heart rate, heat flux, conductivity, Global Positioning System (GPS) location and body motion are some examples of variables that can be measured with current wearable sensor technologies [Yang and Yacoub, 2006]. These can be convenient, for example, in healthcare applications where it is possible to exploit them for the continuous monitoring of patients and the detection of an emergent health condition.

Contrarily to ambient sensors, wearable sensors have advantages regarding privacy and area of operation. In the first case, users are less reluctant to use them in every location if there is no capture of images or video. Secondly, considering that these sensors are always carried by the user, they are *ubiquitous* and their location coverage is virtually unconstrained. They have also the advantage of being highly portable and do not require fixed equipment. On the other hand, wearable sensors have also brought new challenges: preserving battery life and minimizing obtrusiveness while being able to gather reliable context information from limited sensing. These sensors are sometimes uncomfortable for the common user (e.g. if they are fastened too tight or wired or if they need to be constantly repositioned after dressing) and cannot provide a long-term solution for activity monitoring without recharging them regularly.

In addition, *hybrid sensing* approaches, which combine wearable and ambient sensors from different sources, offer an alternative robust option for HAR. For instance, in [Tapia et al., 2006], a sensor rich environment has been set for the collection of signals from 72 environmental and body sensors aiming to evaluate complex activities in an indoor location. In this work, we

employ accelerometers and gyroscopes for the sensing human body motion. We describe their key features as follows.

2.3.2.1 Accelerometer

The accelerometer is an instrument that measures the experienced physical acceleration of an object. It has been employed for several applications in science, medicine, engineering and industry such as for measuring vibrations in machinery, acceleration in high-speed vehicles and moving loads on bridges. For what concerns to HAR, the accelerometer is one of the most widely used sensors for reading body motion signals [Mannini and Sabatini, 2010].

Its principle of operation generally consists of a seismic mass which is displaced in relation to the acceleration it is subjected to. The displacement can then be transduced into a measurable electrical signal. This phenomenon has been applied for the development of *Microelectromechanical Systems* (MEMS) sensors. Their technology allows to create nano-scale devices fabricated with semiconductors. They are advantageous against other sensor technologies because it is possible to produce them in large scale and with low manufacturing costs. Most common MEMS accelerometers work as a capacitive sensor composed of a cantilever beam with a proof mass whose deflection is correlated with the acceleration experienced by the sensor [Woodman, 2007; Yang and Hsu, 2010].

Acceleration magnitude and direction can be measured as a vector quantity by orthogonally arranging sensors in the three spatial dimensions. This can be also built on a single chip and it is now common to find triaxial accelerometers in several commercial electronic devices. This is the case of smartphones which we exploit in this research.

One of the problems of using accelerometers for detection of body motion is the effect of the gravitational field which is always present in the measurements and its magnitude ($g = 9.81m/s^2$) is relatively high. However, it can also be separated from body motion by filtering (see Section 4.3.2). Sensing the gravity vector can also help to determine orientation of an object with respect to the gravitational-force axis when triaxial accelerometers are used.

2.3.2.2 Gyroscope

A gyroscope is a sensing device for measuring *orientation* [Woodman, 2007]. It has been used in many applications such as inertial navigation systems, aerial vehicles for stability augmentation (e.g. in quadcopters) and recently, it has been introduced in electronic devices (e.g. smartphones, game consoles) for enhancing user interfaces and gaming experience. For HAR, this sensor has been employed in various applications such as for the detection of activities (e.g. walking, stair climbing) and transitions between postures (e.g. from standing to sitting) [Coley et al., 2005; Najafi et al., 2002].

Gyroscopes have also been produced with MEMS technologies. However, sensors of this type can only measure orientation indirectly. They estimate *angular speed* instead which can then be integrated in time in order to obtain orientation. However, it is required first to have a reference

Hardware	Sensors	Communications	Services
CPU	Camera	Wi-Fi	SMS
Battery	Accelerometer	GPRS	MMS
Internal Memory	Gyroscope	EDGE	Email
Touchscreen	Compass	Bluetooth	Internet
SD Card	Proximity	USB	Third-party apps
SIM Card	GPS	2G, 3G, 4G	Radio
Speakers	Barometer	NFC	Gaming
	Light Sensor		
	Microphone		

Table 2.2: Common features of current smartphone models in the market

initial angular position to achieve this. These sensors are also highly prone to noise which can cause measurement drift from integration.

Several MEMS gyroscopes obey the tuning fork principle in which a couple of test masses are driven to resonance [Yazdi et al., 1998]. Then, their displacement with respect to their plane of oscillation can be measured into a signal related to the angular speed of the sensor. Similarly to the accelerometer, it is possible to measure angular velocity (magnitude and direction) using arrangements of single-axis gyroscopes.

2.4 Smartphones: The Evolution of the Traditional Mobile Phone

Since the launch of the first mobile phones to the market by the end of the 1970s [Oliphant, 1999], a remarkable growth in the production of these devices has been observed over the years. Statistics have shown that by 2011, this technology reached nearly 80% of the world population [Ekholm and Fabre, 2011]. This trend is clearly showing that mobile devices will be accessible to virtually everyone in a matter of years.

Conventional mobile phones offer the basic telephony service which allows people to communicate wirelessly from any location, provided it is within the available cellular network coverage. These devices also include other essential communication services such as Short Message Service (SMS), Multimedia Messaging Service (MMS) and email.

The user need for new services such as the instant access to information and more sophisticated communication alternatives has stimulated the emergence of smartphones. These devices, which are mainly a new generation of mobile phones, are provided with more computing capabilities, integrated hardware, services and wireless communications than the traditional mobile phones. Web access, multimedia, gaming, location-based services, image capture and video recording, are just a few examples of their available features. Figure 2.2 shows an smartphone and lists some of its features.

Moreover, Table 2.2 lists a set of features available in current smartphone models. Particular importance should be given to the substantial number of sensors in these devices, nine in this



Figure 2.2: Example of a commercial smartphone (SGSII) and some of its features.

case. They have been originally included to enhance human-computer interaction but soon it has been observed that they can bring a broad range of measuring possibilities for several applications [Lane et al., 2010]. Nowadays, they are playing an important role in the exploration of novel information retrieval approaches directly from the users. This characteristic, known as *opportunistic sensing* (Section 2.4.3), along with the high availability of these mass-marketed devices which nearly cover the entire population, seems an interesting area to be exploited for its application in HAR.

2.4.1 Characteristics and Selection Criteria

Various essential aspects are required for the selection of an appropriate smartphone for HAR applications. These include:

- Hardware specifications, regarding memory capacity, processor speed, and power consumption able to cope with the algorithm requirements. Other aspects, such as size and weight, need to be considered as they could contribute to the device's obtrusiveness during use.
- Mobile Operating System (OS), which is in charge of the device's resource management and controls the operation of apps. Moreover, other characteristics such as open source model availability, user interface and developing interface should be taken into account.
- Sensors available, this aspect examines the type of sensors embedded on the device (and

possible connection with others), their specifications and access possibilities through the OS.

First mobile phones used embedded systems to control their operation. In contrast, smartphones now employ mobile OSs for this purpose. Their systems have similar characteristics to OSs in PCs such as being responsible for providing common services to apps and programs, and the management of resources and hardware. However, OSs for smartphones are also highly oriented towards energy efficiency due to the device's mobile nature and battery limitations.

Furthermore, the development of a new generation of portable devices (e.g. tablets, laptops, etc.) is helping to vanish the line from fixed to mobile computing. This extends the possibility of transferring smartphone tools easily, not only to different OSs but also devices. As an example, Microsoft Windows latest OS's platform (Windows 8) has hybrid characteristics that allows its operation either in personal computers and tablets [Honeycutt, 2012].

The smartphone operation is not limited only to its OS, it should also be accompanied with an appropriate hardware. The computational power of high-end smartphones is nowadays not far from the one in standard personal computers regarding processor speeds and communication services. Nevertheless, substantial improvements are needed in some aspects. Particularly and most importantly in battery life. It is, in general, very short, in the order of hours rather than days as it used to be with first mobile phones. Memory management and shared resources are other aspects that affect the device performance and also need special attention. Meanwhile, we overcome these issues by developing light and energy efficient applications. These aspects are then taken into consideration for the design of our HAR systems.

2.4.2 Smartphones as Wearable Sensors

Wearable technologies comprises all the devices that are *body-worn* and allow to gather and process information from the users and their interaction with the environment. In this research, we use smartphones as a wearable device given they are now provided with numerous internal sensors, some of which can be exploited for motion sensing and are thus appropriate for the identification of human activities. Table 2.3 shows a list of these sensors. We selected two of them: the *accelerometer* and the *gyroscope*. They provide information about the user's linear acceleration and angular velocity respectively when used as a wearable sensors, and are not highly affected by external factors such as bad indoor signal reception in GPSs or electromagnetic noise in compasses. However, accelerometers measurements are always influenced by the gravity component in the detection of the body motion acceleration (See Section 2.3.2.1). Similarly, we work with the *acceleration* and *angular velocity* signals which are directly read from these sensors and avoid their integration to obtain either *position* or *orientation* information given the known drift due to noise usually found in this type of inertial sensors (Section 2.3.2.2).

Accelerometers have been included in smartphones since 2007 [Lane et al., 2010], while gyroscopes were introduced more recently (2010) in mid- and high-end devices and they have demonstrated to improve the recognition performance in HAR systems when used in combination

Table 2.3: Smartphone sensors with potential use in motion detection.

Sensor	Measures/Captures	Advantages/Disadvantages
Accelerometer	Linear acceleration	Gravity component is always present in the measurements. 3D position estimation requires double integration therefore it is susceptible to a large position drift.
Gyroscope	Angular Velocity	Angle orientation is estimated through integration of the angular velocity and prone to drift due to signal noise.
Camera	Images & Video	Data extraction from images is computationally expensive and battery inefficient. It brings concerns regarding user's privacy.
Compass	Magnetic field	Measures orientation respect to the magnetic north but it is susceptible to electromagnetic noise.
GPS	Geolocation	Directly measures global 3D positioning but it does not work indoors due to low signal strength from satellites.
Barometer	Atmospheric pressure	Can deliver altitude coordinates and helps to rapidly acquire GPS location. Its precision is low and can be affected by atmospheric factors (e.g. air currents and changing weather).

with accelerometers such as in [Anguita et al., 2013d; Wu et al., 2012]. As these sensors are both based on MEMS technologies, they are small, affordable and therefore suitable for commercial use and integration in portable devices. Noise is however a limiting factor of these sensors which we deal with through a few signal processing solutions (Refer to Section 4.3.2).

Smartphones are one of the portable devices users spend more interaction time with while performing their usual Activities of Daily Living (ADL). This aspect combined with their characteristics and the possibility of collecting and distributing usage data make them a exploitable tool for HAR, for example, when used as wearable sensors. However, disadvantages related to these sensors (Section 2.3.2) such as obtrusiveness and user-awareness are reduced to some extent taking into account that smartphones are becoming part of our environment and their users are familiarized with them.

2.4.3 Opportunistic Sensing

Our environment is now a highly instrumented infrastructure composed of all kinds of devices and sensors which can provide, to some extent, information about it and its interaction with users. When these elements are exploited as a sensing resource and its use is *different* from its original function, we refer to the term *Opportunistic Sensing*. For example: the detection of the activation of light switches at home can be a good indicator of the location of its occupants and can contribute to infer their performed activities. Even more, this can be improved if the gathered information is combined with the one coming from household appliances such as microwaves, blenders, computers, phones, etc. (e.g. if the lights and the TV of the living room are on, there is a high chance that somebody is watching TV) [Ogawa et al., 2002]. Clearly,

advantage is being taken from these existing elements even though neither the switches nor the appliances were firstly designed for this purpose.

Similarly, smartphones are equipped with a varied range of sensors which can be exploited for new opportunistic sensing applications. The Nokia N95 mobile phone for example, was launched in 2009 and it was one of the pioneers in introducing motion sensors [Campbell and Choudhury, 2012]. Its embedded accelerometer was originally used only for photo orientation and video stabilization but a subsequent software upgrade provided developers control over the sensor through an Application Programming Interface (API) and allowed them to take advantage of it for novel applications. Even microphones that were primarily designed to make and receive telephone calls, is now exploited for several mobile applications such as in voice recognition as interface between the user and the device, and also to measure *social isolation* based on the duration of ambient conversations [Lane et al., 2012].

Also, the combination of accelerometers and gyroscopes have brought many advantages regarding spatial positioning and its use on smartphones. The sensing of 6-axis motion have been already exploited for a richer gaming experience. For example, in first person shooter games, players can explore three dimensional locations in a more realistic manner by tilting or twisting their devices instead of using the touchscreen or other phone controls.

2.5 Machine Learning

ML is the area of study concerned about the design, development and evaluation of systems capable to *learn from data*. In many common situations where we need, for instance, to complete a particular task, or perhaps to make some prediction regarding a given issue, it is possible to find solutions by the inspection and analysis of previous observations with similar characteristics to the addressed problem. In other words, ML systems are capable of predicting future actions based on past experiences [Bishop, 2006; Murphy, 2012].

Data are used as the input of the learning process and their representation is fundamental for the performance of the ML systems. They must describe any specific situation to better predict future data in a meaningful way. The property that allows to correctly predict unseen samples, is known as *generalization* and it is highly desirable in any learning machine as it is directly related to its performance.

2.5.1 Taxonomy of ML Algorithms

ML algorithms have been categorized according to the type of input used for training and its expected outcome. In this section, we describe the most relevant categories.

2.5.1.1 Supervised Learning

In this type of learning, input data are usually composed of a pair of elements, namely the *input vector* (\mathbf{x}) together with its *target* (y) [Bishop, 2006]. This can be better clarified with

an example: assume a system that learns handwritten numbers from 0 to 9. The *input vectors* would be the set of images of all the numbers (usually several samples per each one) and the *target vector* the actual labels that correspond to each sample.

If the output of desired system is categorical (only a set of discrete classes are considered), then it is a classification problem such as the example presented above. Otherwise, if the output data are continuous variables, such as in temperature forecasting or stock market prediction, then the system is considered a regression.

When the learning is performed gradually, for instance, by adding one new sample and its target at a time to the model, we refer to *Online Machine Learning* [Shalev-Shwartz, 2011]. This supervised approach have the advantage of making the model adaptive and flexible in accordance to the new inputs. This type of learning is required in applications with high output variability and where a stream of new samples is available and can be progressively added to the model for learning. This is the case of online web ranking and stock market prediction applications.

This algorithm type is the most commonly used for ML and it is also the one used in our research. However it is not useful to solve all kind of problems. In fact, one of its disadvantages is that in some applications it is not always possible to have target information for all the available input samples. As a result, other techniques can cope with these situations such as unsupervised and semi-supervised learning which are described as follows.

2.5.1.2 Unsupervised Learning

In an unsupervised learning problem, the training data consists of only *input vectors* without their associated *targets*. It aims to find certain similarities or discover distinguishable structure within the input data (e.g. *clustering*). It can also be used for *density estimation* to describe the distribution of the data in its space. Moreover, this learning approach can be exploited for *data visualization* using dimensionality reduction methods which allow to better project high-dimensional data into smaller spaces (e.g. 2D and 3D).

Unsupervised learning approaches have been already applied in several areas such as in medical imaging where 3D Positron Emission Tomography (PET) scans use cluster analysis to find dissimilarities between different organs and types of tissue to be able to correctly segment the scanned area [George et al., 2011]. It has also been applied in the automatic grouping of similar shopping items (e.g. books, movies, music), particularly in recommender systems for online stores that aim to predict the user preferences based on products similarities and previous purchases.

2.5.1.3 Semi-supervised learning

This learning approach combines labeled and unlabeled data for learning. Therefore, it takes aspects from both supervised and unsupervised approaches. In general, small amounts of labeled data are integrated with a large number of unlabeled samples for learning. For example, it is useful for datasets where it is not always possible to have a label for each sample. Evidence

have shown that semi-supervised learning can greatly improve the learning performance when compared with supervised learning which does not take into account unlabeled data. This is feasible if considerations such as the data smoothness assumption apply [Chapelle et al., 2006]. Vast digital image collections on the internet for content retrieval are an application example where this type of learning can be exploited. Not all the images have an associated targets and it would be humanly impossible to perform this labeling manually [Fergus et al., 2009].

2.5.1.4 Reinforcement Learning

This learning approach is oriented on finding an appropriate set of actions to solve a particular problem. This is done with the purpose of maximizing a reward. Optimal solutions are not found through learning a model given a set of *input-target* pairs. Instead, they are found by trial and error [Sutton and Barto, 1998]. A common example assumes an environment that interacts with an agent which can take actions that allow him to shift between different states. The best actions are guided by the evaluation of a reward function related to each problem.

Humanoid Robotics is one of the research areas where Reinforcement Learning (RL) has shown its great potential [Vijayakumar et al., 2003]. Although there are still many difficulties to solve RL problems when the complexity of the problem increases (e.g when the dimensionality is too high or states are continuous), there have been successful real life samples that have demonstrated its applicability such as the *cart-pole* which automatically controls an inverted pendulum [Doya, 2000].

2.5.2 Machine Learning Approaches

Several ML modeling approaches have been developed throughout the years in order to solve different tasks such as classification, regression and clustering [Bishop, 2006]. Some of them are based on deterministic models which aim to find fixed causal relationships between events. Other approaches, on the other hand, are probabilistic and assume occurring events are generated from a probability distribution. Combinations of these approaches have also been explored such as in [Franc et al., 2011].

In the following list, the most popular ML algorithms are briefly described. Then, in the next section we make particular focus on SVMs as they are the central ML algorithm employed in this thesis.

- **Decision Tree (DT):** is a predictive model based on decision trees which makes choices from a set of hierarchical rules related to the input data. Different versions have been proposed such as ID3 and C5.4 [Quinlan, 1986, 1993]. It is a common approach for classification particularly because the resulting models are easily interpretable by humans (due to its intrinsic tree structure).
- **Random Forest (RF):** is an ML meta-classifier which is built using an ensemble of DTs. The predicted class is chosen as the most frequently occurring amongst the output of each

DT [Breiman, 2001].

- **k-Nearest Neighbors (k-NN)**: this deterministic learning approach exploits similarity measures between data for classification and regression tasks. Given a new sample, the approach finds the k closest samples from a training set to decide the prediction outcome with their values (e.g. by using majority rule in classification or averaging in regression) [Altman, 1992]. Its main disadvantage relies on the size of its model as it is data-dependent and makes it unfeasible in large datasets. There are, however, versions which consider data reduction techniques for alleviating this issue.
- **Naive Bayes (NB)**: is a popular probabilistic classifier based on Bayes's theorem that predicts the class of a given sample by assuming an underlying probability model of the data and making strong independence assumptions between its features. Even though its formulation is quite simple, it has shown to perform well in various applications [Bishop, 2006]. For example, when data are assumed to be Gaussian-distributed, it is possible to learn the model only by calculating the mean and variance of the input data.
- **Artificial Neural Networks (ANN)**: is an ML approach with a biological inspiration. It simulates how the brain and its nervous system, composed of interconnected neurons, is able to learn from experience and capture the underlying structure of the data. Neurons are set in a layered structure and have associated weights which are able to adapt based on the training data and the network output through a cost function. This approach has shown to perform well in many applications (e.g. [LeCun et al., 1989]), including non-linear problems. Its main disadvantage relies in the need of a large dataset for its training stage. **Multilayer Perceptron (MLP)** is a popular ANN model that maps the input through multiple layers of neurons in a fully connected directed graph until reaching the output.
- **Logistic Regression (LR)** is a probabilistic algorithm used for solving classification and regression problems. It estimates the probability of a given sample of belonging to a particular class. This is achieved through the use of a logistic function which is modeled by fitting the training data generally using maximum likelihood estimation. [Landwehr et al., 2005].

2.5.3 Support Vector Machines

A Support vector machine is one of the most commonly used supervised ML algorithms. It was initially proposed by Vladimir Vapnik and his colleagues in [Boser et al., 1992; Cortes and Vapnik, 1995] with the aim of solving linear and non-linear binary classification problems. Afterward, this algorithm has been adapted for its application in *multiclass classification* and *regression analysis* [Drucker et al., 1997; Weston and Watkins, 1998].

The SVM for classification is a deterministic approach that aims to find the hyperplanes that best separate the data into classes. These subspaces are the ones that provide the largest

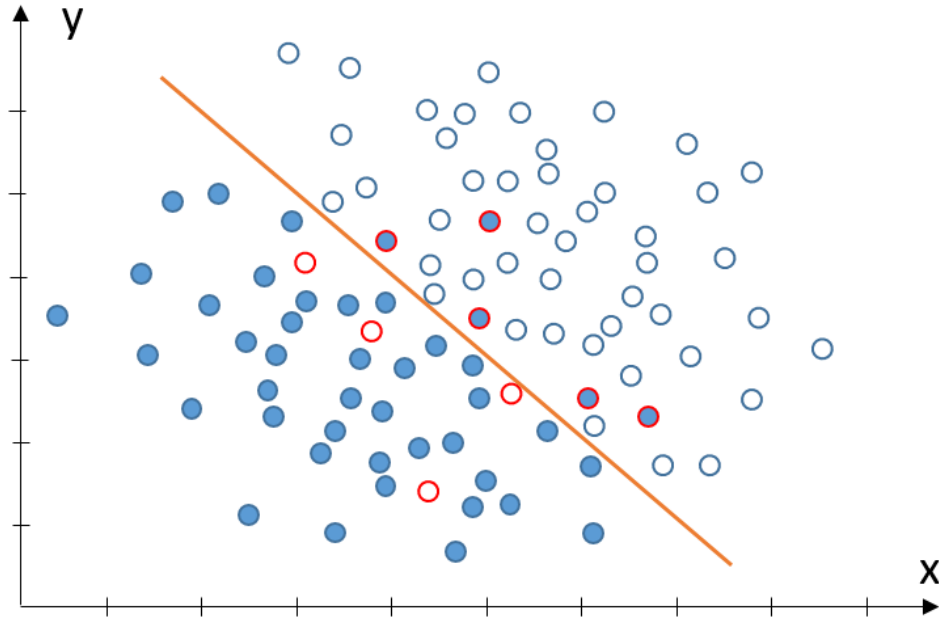


Figure 2.3: Example of a binary classification problem in a two-dimensional space. The line represents a possible solution to the problem and the dots with red outline are the misclassified dataset samples.

margin separation from the classes of the training data with the intention of providing a model with low generalization error for its use with unseen data samples.

SVMs are the basis for the classification of activities in this work. For this reason we now introduce them, starting from the binary SVM model which is its simplest representation, to the extended case that allows the classification of more than two classes: the multiclass SVM. This algorithm will be further revised throughout the development of this research to tackle specific requirements for our application in aspects such as kernel type, arithmetic used and algorithm output type.

2.5.3.1 The binary HF-SVM model

In a binary classification problem only two classes are involved and it aims to predict the class that a given new sample belongs to. As a mode of illustration, Figure 2.3 shows an example in a two-dimensional space. Two classes represented by *white* and *dark* dots are depicted. The solution to this classification problem is to find a way to separate the elements in two groups in order to reduce the number of misclassifications to a minimum. In this example, a simple approach involves tracing a line to divide the space in two regions (such as the one depicted in the figure). This approach, however, produces some classification errors (the ones highlighted in red) because these groups are partially overlapped. A non-linear approach is therefore most likely to better solve this particular data configuration.

More formally, consider a dataset composed of n samples. Each one corresponds to an ordered pair $(\mathbf{x}_i, y_i) \forall i \in \{1, \dots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ are the input vectors and $y_i = \pm 1$ are their target values representing one of the two possible classes. We want to solve this problem by finding a linear hyperplane which separates the data in two groups. The set of linear classifiers which are possible solutions to the problem is of the form:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b, \quad (2.1)$$

where $\mathbf{w} \in \mathbb{R}^d$ is a vector orthogonal to the hyperplane (commonly known as weights), and the scalar $b \in \mathbb{R}$ is the bias. A standard linear SVM aims to find the \mathbf{w} and b values that allow the optimal separation of the data in order to provide the largest margin between the classes and the lowest error rate of the available data. This can be learned by solving Convex Constrained Quadratic Programming (CCQP) minimization problem formulated as:

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \mathbf{1}_n^T \boldsymbol{\xi} \\ & y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in \{1, \dots, n\}, \end{aligned} \quad (2.2)$$

where $\boldsymbol{\xi} \in \mathbb{R}^n$ is the vector of slack variables associated with each sample and represent a measure of error ($\xi_i = 0$ when the sample is correctly classified, $0 < \xi_i < 1$ when is also correctly classified but lies within the margin, and $\xi_i \geq 1$ when it is misclassified). C is the hyperparameter which allows to trade-off the contribution of the margin¹ and the error terms. In this thesis we will use the notation \mathbf{a}_b to represent a vector of dimension b of the scalar a . In this way, $\mathbf{1}_n^T \boldsymbol{\xi}$ can be seen as an upper bound of the number of errors.

This formulation is called the *primal* problem. Moreover, it can be reformulated and solved more easily by using the Lagrange multipliers such as presented in [Oneto and Greco, 2010] in order to obtain the *dual* formulation. The solution starts with the Lagrangian of the primal formulation (Equation (2.2)) where two sets of multipliers are introduced. These are the ones linked to the first and second constrains: $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{\mu} \in \mathbb{R}^n$ respectively. This is presented as:

$$\mathcal{L}_p(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \mathbf{1}_n^T \boldsymbol{\xi} - \sum_{i=1}^n \alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^n \mu_i \xi_i, \quad (2.3)$$

Following this, with can obtain the Karush-Kuhn-Tucker (KKT) conditions for the Wolfe dual problem [Karush, 1939; Kuhn et al., 1951]. They include the partial derivatives of the Lagrangian (\mathcal{L}_p) with respect to \mathbf{w} , b , and $\boldsymbol{\xi}$; and slackness conditions for $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$.

¹The $\|\cdot\|$ symbol represents the Euclidean norm unless stated otherwise.

$$\frac{\partial \mathcal{L}_p}{\partial w_j} = 0 \rightarrow w_j = \sum_{i=1}^n \alpha_i y_i x_{i,j}, \quad \forall j = 1, \dots, d \quad (2.4)$$

$$\frac{\partial \mathcal{L}_p}{\partial b} = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = 0, \quad (2.5)$$

$$\frac{\partial \mathcal{L}_p}{\partial \xi_i} = 0 \rightarrow C - \alpha_i - \mu_i = 0, \quad \forall i = 1, \dots, n \quad (2.6)$$

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, n \quad (2.7)$$

$$\alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) = 0, \quad \forall i = 1, \dots, n \quad (2.8)$$

$$\mu_i \xi_i = 0, \quad \forall i = 1, \dots, n \quad (2.9)$$

$$(C - \alpha_i) \xi_i = 0, \quad \forall i = 1, \dots, n \quad (2.10)$$

$$\alpha_i, \xi_i, \mu_i \geq 0, \quad \forall i = 1, \dots, n \quad (2.11)$$

The dual form is obtained as:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^n \alpha_i \\ & 0 \leq \alpha_i \leq C, \quad \forall i = 1, \dots, n \\ & \sum_{i=1}^n y_i \alpha_i = 0, \end{aligned} \quad (2.12)$$

which can be further simplified in matrix form as:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{1}_n^T \boldsymbol{\alpha} \\ & 0 \leq \alpha_i \leq C \quad \forall i \in \{1, \dots, n\}, \quad \mathbf{y}^T \boldsymbol{\alpha} = 0, \end{aligned} \quad (2.13)$$

where Q is the kernel matrix and is a symmetric positive semidefinite $n \times n$ matrix where $q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$.

$K(\cdot, \cdot)$ is the kernel function which helps to solve linear and non-linear problems. This is possible through an *implicit* mapping of the inputs known as the *Kernel trick*. The idea was first introduced in [Aizerman et al., 1964] and it assumes the existence of a function $\phi(\mathbf{x})$ which maps a sample from its original space to a *feature space* (\mathcal{H}) which is higher-dimensional (with potentially infinite dimensions) and it is defined as:

$$\phi(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathcal{H} \quad (2.14)$$

In a binary classification problem, if samples are mapped into \mathcal{H} , they are more likely to be linearly separable. Therefore, the use of a linear SVM to solve it appears to be practical.

Moreover, from its formulation, we can also notice that it is not necessary to explicitly compute the $\phi(\mathbf{x})$ function but rather the inner product between two elements. This is known as the *Kernel Function* and its formulation is:

$$K(\mathbf{a}, \mathbf{b}) = \phi(\mathbf{a}) \cdot \phi(\mathbf{b}) \quad (2.15)$$

Its only requirement is that it must satisfy the Mercer's theorem [Cristianini and Shawe-Taylor, 2000] which represents a symmetric positive-definite function on a square by means of a sum of a convergent sequence of product functions.

In order to solve the CCQP problem, many efficient algorithms have been previously proposed [Shawe-Taylor and Sun, 2011], such as, for example, the well-known Sequential Minimal Optimization (SMO) [Keerthi et al., 2001; Platt, 1998].

Once the α_i coefficients are found, new patterns can be classified by applying the SVM Feed-Forward Phase (FFP) which is given by the following general formulation:

$$f(\mathbf{x}) = \sum_{i=1}^n y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \quad (2.16)$$

where b is the bias term and can be estimated by taking into account the KKT conditions that were formulated to obtain the dual problem as described in [Oneto and Greco, 2010]. They consist of the derivatives and the slackness conditions of the Lagrangian of the primal formulation. We can obtain b from the following condition:

$$\alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0 \quad (2.17)$$

and assuming that we have a data sample whose $\alpha_i > 0$, we find that:

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) = 1 \quad (2.18)$$

From Equation (2.16) it is also possible to realize that only the samples which affect the classification of new samples are those with $\alpha_i > 0$. They are denominated *support vectors*. Geometrically, these points usually lie close to the margin bounds between the two classes.

2.5.3.2 Extension to the Multiclass SVM

It is possible to generalize binary ML models to solve problems with more than two classes. This process is known as *multiclass* or *multinomial* classification. Figure 2.4 shows a simple example of a set of elements from 3 different classes in a space of two dimensions, each one represented with a different color (red, green and blue). Also, separating hyperplanes are chosen as a possible solution to this problem.

There are several methods that have been previously proposed for solving multiclass problems from binary formulations. [Hsu and Lin, 2002], but generally the two most commonly used are: OVA and one-vs-one (OVO). Their difference relies in the way they compare each class of interest

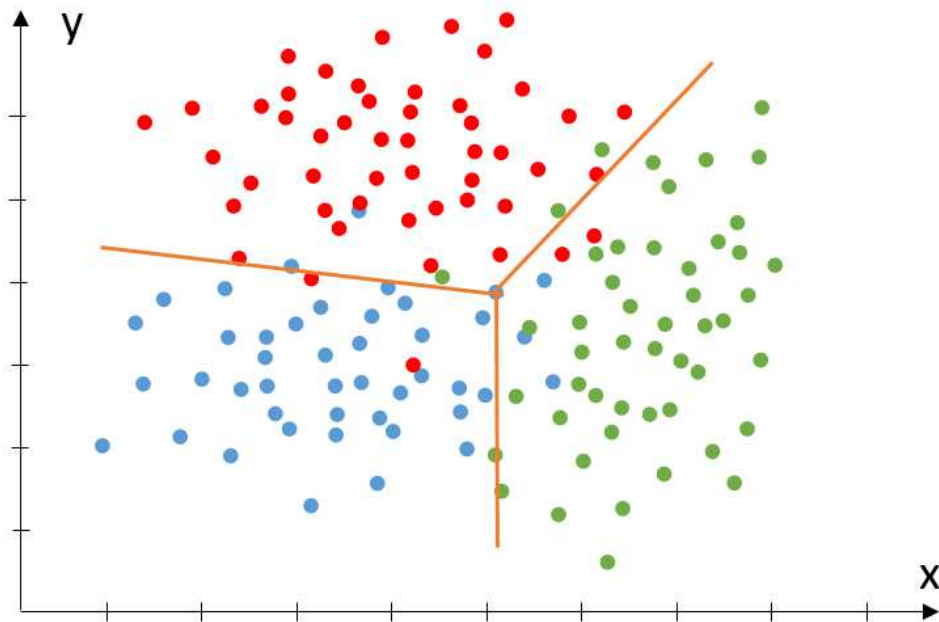


Figure 2.4: Example of a multiclass problem in a two-dimensional space. Three classes represented by the red, green and blue dots are separated by hyperplanes which provide a possible linear solution to the classification problem.

against the remaining ones: either *all together* for the first case and *one by one* for the latter. In this work we use OVA [Rifkin and Klautau, 2004] and take advantage of it because its output directly represents how likely is each class to match a new test sample against the rest.

The OVA approach consists on constructing a set of m binary SVMs, each one associated to each existing class c . They are built from *positive* training samples coming from the class of interest (labeled as $+1$) and *negative* samples which contain the remaining samples (labeled as -1). Once the SVMs are learned, it is possible to compare them to determine which class is the most likely to represent a test sample. The output of the FFP for every class ($f_c(\mathbf{x})$) is either positive or negative and its sign represents if the new sample is either classified as a given class or not. Ideally, for a given sample in a multiclass problem, only one of the binary classifiers should be positive. Therefore the classification of a new sample can be then formulated as a winner-take-all arbiter which selects the label c^* corresponding to the class with the maximum value of the SVM:

$$c^* = \arg \max_c f_c(\mathbf{x}) \quad (2.19)$$

In the following chapters, we will deal with better approaches to compare the output of the SVMs from multiple classes such as the use of probability estimates (refer to Section 5.2.2).

Actual Class	Predicted Class	
	a	b
	a	True Positives False Negatives
	b	False Positives True Negatives

Table 2.4: The four fundamental numbers for estimating statistical performance measures of a classifier

2.5.4 Performance Evaluation

The evaluation of ML algorithms is predominantly made through the statistical analysis of the models using the available experimental data. The most common method is the *confusion matrix* which allows representing the algorithm performance by clearly identifying the types of errors (false positives and negatives) and correctly predicted samples over the test data. From it, various metrics can also be extracted such as model accuracy, sensitivity, specificity, precision and F1-Score [Bulling et al., 2014; Lara and Labrador, 2012a]. In addition, other comparative qualitative indicators, such as the number of available activities, prediction speed and memory consumption, can support the selection of HAR algorithms.

2.5.4.1 Confusion Matrix

A common method to visualize the performance of an ML algorithm is through the *confusion matrix* \mathcal{C} , also called *contingency table*. Assuming there are m classes available, a typical confusion matrix consists of a squared matrix of size $m \times m$ where misclassifications are visible outside the diagonal. For example, if we consider a dataset composed of n patterns, where each one corresponds to an ordered pair of extracted features and their corresponding activity label $(\mathbf{x}_i, y_i) \forall i \in \{1, \dots, n\}$, $\mathbf{x}_i \in \mathbb{R}^d$, and $y_i \in \{1, \dots, m\}$, we can obtain \mathcal{C} when provided with the predicted labels $f_c(\mathbf{x})$.

Rows represent the actual activities and columns the predicted ones. Therefore, each matrix cell $\mathcal{C}_{i,j}$ shows the number of instances of activity i that were predicted as activity j . It is clear then that all the values within the matrix diagonal are correct predictions and classification errors otherwise. From this matrix, we can visualize four different helpful values used to estimate the various statistical measures regarding the system performance. These are visible in the simplified confusion matrix of two classes (a and b) from Table 2.4. We take these values assuming class a is the class of interest or positive condition.

- *True Positives (TP)*: actual samples of class a correctly predicted as class a
- *True Negatives (TN)*: actual samples of class b correctly predicted as class b
- *False Positives (FP)*: actual samples of class b incorrectly predicted as class a
- *False Negatives (FN)*: actual samples of class a incorrectly predicted as class b

2.5.4.2 Accuracy, Sensitivity and Specificity

The *accuracy* can be obtained as the proportion of the true results (TP and TN) with respect to the total number of instances in the population:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.20)$$

This measure gives an indication of how good the overall performance of a classifier is. Moreover, we can also use the *error* measure for expressing the opposite. It denotes the deviation of the measurement from the truth and it can be obtained in terms of the accuracy:

$$error = 1 - accuracy \quad (2.21)$$

Sensitivity, also known as the *true positive rate* or *recall*, is a measure of how good is the classifier to correctly predict actual positives samples. Its formulation is:

$$sensitivity = \frac{TP}{TP + FN} \quad (2.22)$$

In contrast, the *specificity* measure, also called the *true negative rate*, shows the ability to correctly predict actual negative samples. It is formulated as:

$$specificity = \frac{TN}{TN + FP} \quad (2.23)$$

The measures above are the ones we mostly use throughout this work. Nevertheless there are also some statistical measures that are widely used such as *precision*, also known as *positive predictive value*, which is the rate of TP with respect to all the predicted positives, and F_1 -score (or *F-measure*) which is a general measure of the classifier's accuracy that combines *precision* and *sensitivity*. These two can be estimated in the following way:

$$precision = \frac{TP}{TP + FP} \quad (2.24)$$

$$F_1\text{-score} = 2 \cdot \frac{precision \cdot sensitivity}{precision + sensitivity} \quad (2.25)$$

2.5.4.3 Qualitative Criteria

In order to select the most appropriate classification system for a particular application, it is also interesting to take into account a set of various qualitative criteria in addition to the statistical measures. These aspects can help to make important trade-off decisions during this selection process. Some of them are described as follows:

- *Online capability*: tells whether the system is able to perform activity recognition in real-time.

- *Recognition time*: is the time delay associated with the activity prediction process. For instance, the length of the time window related to each prediction and its CPU processing time.
- *Battery consumption*: quantifies the energy expenditure of portable devices and how this affects their continuous operation time.
- *Memory and CPU usage*: considers limitations regarding memory and CPU requirements. It is critical for instance in systems that share resources with others applications (e.g. in smartphones, personal computers).
- *Obstrusivity*: evaluates how comfortable is the system for the user (e.g. sensors location and weight, presence of wires, fitting time, etc.)
- *User's privacy*: examines whether the system protects the accessed personal data of its users from external sources.
- *Number and type of classes* the system is able to classify.
- *Number and type of sensors* required for the recognition of activities.
- *Modular design*: indicates whether or not the system allows its integration with others or the adaptation of new sensors and devices.

2.6 Summary

In this chapter we described the background required to contextualize the problem of HAR. We first covered the fields of application our research work is targeting to: AAL and AmI which are human-centered areas aiming to improve people's QoL through the use of smart technologies. Then, we explored three implementation considerations that constitute the main building blocks involved in the development of our proposed HAR systems. These are: Sensing devices, with focus on inertial sensors (accelerometers and gyroscope), for the detection of physical activity; smartphones, the recent everyday use device with strong computing capabilities now exploited as a novel service provider (HAR in our case); and the intelligence behind our systems provided by SVMs.

Chapter 3

State of the Art

3.1 Introduction

The rise of ubiquitous computing systems in our environment is engendering a strong need for novel approaches to human-computer interaction. HAR systems can effectively contribute to develop these approaches due to the unobtrusive and pervasive ways of interaction they provide [Schmidt et al. \[1999\]](#). Human intervention is needed in many systems for decision making, usually by means of interaction through traditional devices such as keyboards, remote controls, switches, or touchscreens [\[Weiser, 1993\]](#). These mechanisms of interaction are becoming intractable considering the amount of devices we are exposed to every day.

Moreover, we are now facing a new challenge as a result of the easy access to vast amounts of information coming from multiple sources. They can contribute to reduce our demanding interaction with machines to a minimum. This is possible especially because we are always (involuntarily or not) providing feedback to the environment through our behavior and actions. Novel systems could therefore become more cognitive and fundamentally transform our ways of interaction with them [\[Cook and Das, 2012\]](#). For instance, in the medical field, monitoring physiological signals in humans, such as an increase in our average daily heart rate or body temperature, can indicate an emerging health condition. This detection can be possible even without the patient being aware of the situation or the need of a check up visit to the doctor.

HAR is an active research field in which methods for understanding human behavior are developed by interpreting attributes derived from various sources [\[Karantonis et al., 2006; Lukowicz et al., 2004; Poppe, 2007\]](#) (e.g. by sensing motion, location, physiological signals, weather and temperature etc.). It aims to identify the actions carried out by a person given a set of observations of him/herself and the surrounding environment. HAR has provided substantial contributions in human-centered areas of study such as AmI, Pervasive Computing (PerComp) and AAL. These areas make use of HAR systems as an underlying technological tool which gathers behavioral information from people about their actions and environment during the course of their daily life and delivers context-aware data that can help to provide valuable services, products and technologies aiming to improve people's QoL.

In this chapter we address the subject of HAR supported by its most relevant works in the literature. In Section 3.2, we start with a general description of this field including aspects such as the standard recognition process, studied activities, common approaches for experimental data collection, and performance evaluation. Subsequently, we emphasize on the state of the art research on HAR systems, some of which are described and compared against our approaches in Section 3.3. Finally, we conclude the state of the art in Section 3.4.

3.2 Human Activity Recognition Overview

HAR is an essential component for the development of systems for allowing smarter interactive cognitive environments. As a mode of illustration, in a simplified view of the human information processing pipeline which consists of four stages (*sensing*, *data analysis*, *decision making* and *taking action*) [Gandetto et al., 2003; Parasuraman et al., 2000], HAR belongs to the first and second stages. It contributes to acquire the necessary information regarding the user activity. This is then combined with the perceived environmental data in order to obtain a complete state representation of the world and its individuals before continuing to subsequent stages. For instance, a system for the management of accidents in the elderly (e.g. falls [Lord, 2007]) would require initially the detection of a potential event from the user’s wearable sensors. Then this detection needs to be fused with environmental information which can help to confirm that what has occurred is not a false negative (e.g. if the sensor felt on the floor). Afterward, it is possible to make decisions about what to do and if necessary take action by calling the emergency services for immediate assistance. In this section we introduce the main concepts behind HAR and their application into real world problems with particular focus on the fields of AAL and AmI.

A general representation of the principal components of a HAR process is depicted in Figure 3.1. Many of the HAR approaches found in literature, follow a regular structure with slight variations based on their application, sensors, and selected ML algorithms. The diagram is valid to supervised, semi-supervised and incremental learning approaches [Karantonis et al., 2006; Stikic et al., 2011; Wang et al., 2012] (Refer to Section 2.5.1). They differ on the type of input (labeled or unlabeled) and if the learned model updates when new samples are added into the system (notice the *Feedback* dotted line on the graph).

From the four main blocks of a HAR system (Figure 3.1), *sensing* is responsible of gathering the sensor data from the available sources and process them (See section Section 2.3). Generally, signal conditioning (e.g. reducing noise, digitizing, amplifying) is always required for adapting the sensed signals to the application requirements. In the second place, the *feature extraction* process is in charge of obtaining meaningful features that describe the data and allow a better representation and understanding of the studied phenomena. The extracted features turn into the input of the ML algorithm, either for *learning* the model or for the activity *prediction* of novel samples when the model already exist.

Moreover, traditional HAR systems usually operate in a feed-forward basis thus *learning* is performed offline only once and there is no further feedback into the system. This is useful in

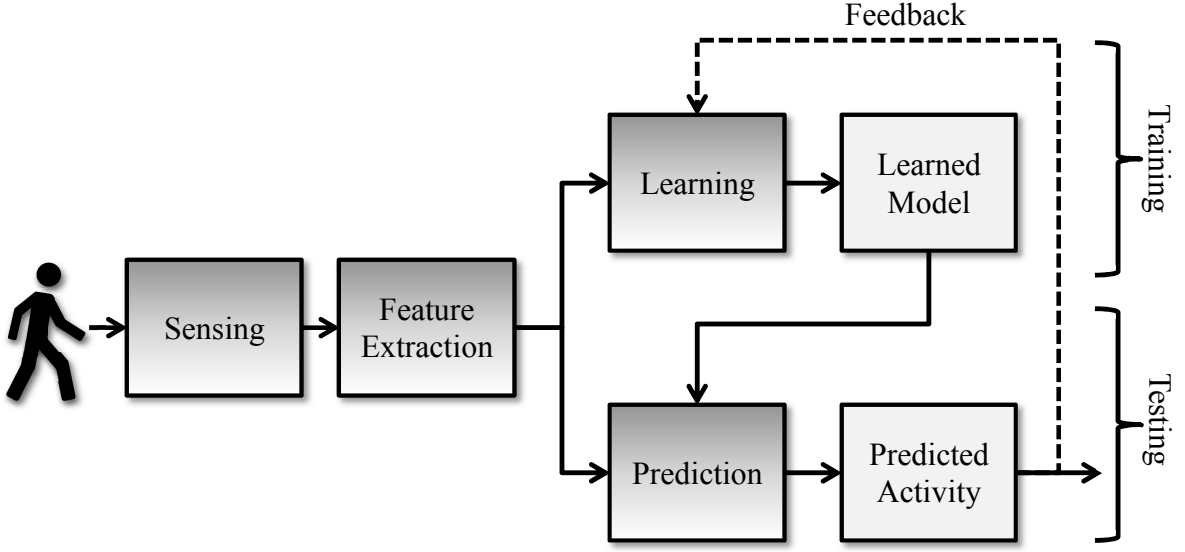


Figure 3.1: The Human Activity Recognition Process Pipeline with its four main blocks.

Duration/Complexity	Activity Type	Examples
Short Events	Gestures Transitions	waving hands, nodding head, laughing stand-to-sit, lie-to-sit
Basic Activities (BAs)	Static Dynamic	standing, sitting, reading walking, running, cycling
Complex Activities (CAs)	Multi-activity Multi-user	cooking, assembling furniture, weight training talking, ballroom dancing, hugging

Table 3.1: Classification of activities by duration and complexity.

cases where the data distribution does not change over time or the system is subject-independent and robust against high input variability. Otherwise adaptive methods such as incremental online or transfer learning [Zheng et al., 2009] are advised but conditioned with an increase in the computational load into the process. In relation to the analysis of high level activities, which are combinations of simple activities (e.g. assembling furniture or fixing a car [Amft et al., 2007]), there is a limited amount of work that has been done and it is still an open research field.

3.2.1 Human Activities

A HAR system is dependent on the set of activities to recognize as they can directly affect the way systems are designed and implemented. For this reason, the classification of activities in categories simplifies the selection of appropriate mechanisms to recognize activities. Here, we categorize them with respect to their duration and complexity, and activity type.

Table 3.1 shows the categorization of activities with respect to their duration and complexity. There are three main groups: *Short events* are simple activities with a defined small duration. These are divided in two types. First, *Body gestures* which are visible motions mainly used as

Application	Examples
Daily living	Watching TV, ironing, eating, showering, cleaning
Locomotion	Walking, riding, standing, laying down, falling
Sports/Fitness	Jumping, weight lifting, climbing, swimming
Communication / Connectivity	Phone calling, texting, talking, signing
Security/Surveillance	Loitering, chasing, supervising, stalking

Table 3.2: Classification of activities by type.

a mechanism for nonverbal communication and second, *transitions* which are the events that connect the execution of two different activities. For example: when a person is seated and then stands up, the sit-to-stand (SiSt) PT occurs. The second group is composed of *Basic Activities*. They are mostly characterized either by a continuous or a cyclic action. Their duration is variable but in general they take much longer than *short events* (in the order of minutes). Two activity types are derived from this class: *Static* which involve Static Postures (SPs) such as *sitting* and *standing* and *dynamic* which are activities that have periodicity such as *walking* and *running* (here we will refer to them as *Ambulation Activities (AAs)*). Finally, the third group is the one of *Complex Activities (CAs)*. This comprises the rest of activities which are generally a sequence of BAs and *short events (multi-activity)* or that involve more than one subject (*multi-user*) such as ballroom dancing which requires the participation and coordination of two people while performing several movements.

There are currently many applications of public interest where HAR is greatly needed. For instance: in healthcare, for tailoring medication of disabled patients with motor problems through the continuous monitoring of *locomotion* and *daily living* activities using inertial sensors, and in security, for the surveillance of public places and crime prevention through the detection of *security-related* activities using video cameras [Avci et al., 2010; Lin et al., 2008]. In this manner it is possible to see a clear categorization of activities according to their type. These groups of activities are, in general, relevant to various HAR applications. Table 3.2 depicts different types of activities which have been employed in previous studies. Other proposed taxonomies of activity type are presented in [Bruno et al., 2012; Lara and Labrador, 2012a].

3.2.2 Sensing and Data Collection

The definition of the experimental set up for data acquisition is also an important aspect in HAR. Depending on how the subject is observed in its habitat with or without any manipulation by the observer. *Naturalistic environments* are ideal for experimentation but in many cases it is not feasible to exploit them. Therefore, controlled experiments can be carried out in laboratory conditions aiming to simulate natural settings (*semi-naturalistic environments*). Otherwise, fully *controlled environments* are the last resource for data acquisition although the performance of the developed method/system with this approach is uncertain until verified in real situations.

Failures in the design of HAR systems can be due to the lack of real life considerations such as unaccounted activities or target users, noise, sensor calibration and positioning, etc. This latter is for instance highly linked to the system performance as presented in [Atallah

et al., 2010; Maurer et al., 2006] where different sensor locations were evaluated for determining the ideal positions for performing HAR through the use of wearable accelerometers. Another final consideration about the experimentation process is the number of individuals selected as generally larger number of people involving various age groups and physical conditions are preferred. This is also directly related with the performance and generalization capability of the system in the presence of new users.

3.2.3 Feature Selection and Extraction

In an ML problem, *feature selection* refers to the process of selecting a significant set of features to largely impact the discrimination ability of a learning algorithm. *Feature extraction*, on the other hand, is an approach to diminish the dimensionality of an available set of features by performing inter-feature transformations in order to obtain a new dimensionally reduced representation without largely sacrificing relevant information from the original set. The *curse of dimensionality*, which describes the difficulty in understanding and dealing with high-dimensional data, is certainly linked with these two reduction mechanisms as they can alleviate the problems that may arise when working in high-dimensional spaces.

Feature selection and extraction also allows to reduce the training times and increase the generalization performance in ML problems. They, however, differ on that the interpretability of models in which feature selection is employed is much clearer. In this case, features are distinct between each other and not merged such as in feature-extraction-based approaches.

Depending on the application, the features required for the extraction of relevant information may vary. In the particular case of HAR, a reduced representation of the sensor data can be used as the input of a recognition algorithm. This is attained by estimating various measures from the sensor signals in different domains (e.g. in time and frequency). Nonetheless, other time-frequency function representations such as the wavelet transforms are also applicable. Once obtained, they can be further reduced using feature selection (e.g. exhaustive search, or wrappers, filters and embedded methods [Guyon and Elisseeff, 2003]) and extraction approaches (e.g. using Principal Component Analysis (PCA) [Bishop, 2006]), or a combination of both. In Section 4.3.3 the features selected for the development of our HAR dataset are presented. Moreover, in Section 6.4 we describe the feature selection approaches used in this work.

3.2.4 Machine Learning

Several ML approaches have been developed throughout the years for HAR. It has mostly been targeted through supervised learning algorithms although semi-supervised and unsupervised methods have also been proposed [Stikic et al., 2008; Wyatt et al., 2005].

Frequentist and Bayesian models have been well covered throughout HAR literature. They involve *rule-based* models such as DT and RF [Coley et al., 2005; Ermes et al., 2008], *geometric* approaches including k-NN, ANN and SVMs [He and Jin, 2009; Khan et al., 2010a; Maurer et al., 2006], and *probabilistic* classification methods as for example NB classifiers, and Hidden

Markov Models (HMM) [Tapia et al., 2007; Zhu and Sheng, 2009].

Many of these ML approaches have demonstrated comparable performance in different works (e.g. [Mannini and Sabatini, 2010]) though suggesting that the effectiveness and right selection of the algorithms can be linked to other aspects such as data structure and application [Wolpert and Macready, 1997]. Other relevant aspects for ML algorithm selection include: energy consumption, memory requirements, interpretability and computational complexity, etc. As a matter of example, decision trees could be preferred when the model interpretability is required and SVMs for high performance applications. In Section 3.3.3 we provide more details regarding HAR systems which have employed different ML algorithms.

3.3 Related work in HAR Systems

Several approaches have been previously proposed in literature for the recognition of human activities. They cover diverse application domains such as healthcare, smart homes, UbiComp, AAL, surveillance and security [Cedras and Shah, 1995; Choudhury et al., 2008; Poppe, 2010; Turaga et al., 2008]. In general, these proposed HAR systems are able to sense, monitor, and learn from our actions in order to provide useful information which can be used to decide better about our future needs or behavior [Cook and Das, 2012].

These approaches can be categorized according to many different criteria. Some of the most relevant are: by *sensor type*, which is reliant on the class of signals measured to extract activity information (e.g. inertial, vision-based and physiological [Lara and Labrador, 2012a]); by *sensor location*, which depends on the position of the sensor with respect to the user. Namely external sensing when sensors are located in fixed positions in the environment and wearable sensing when they are body-attached [Yang and Yacoub, 2006]; by *human activity type*, which groups the systems with respect to the activities they are able to identify (e.g. locomotion and ADLs); by *modeling principle*, which can be data- or knowledge-driven depending on whether the HAR models are built given pre-existing datasets or from the exploitation of prior knowledge regarding a particular domain [Chen et al., 2012a,b]; by *learning approach*, in relation to the type of algorithms used for learning such as supervised, semi-supervised or unsupervised methods [Kwapisz et al., 2011; Stikic et al., 2011; Wyatt et al., 2005].

The work presented in [Bao and Intille, 2004] was pioneer in developing a method for the detection of a set of activities of daily living using five body-worn accelerometers and employing well-known ML classifiers (e.g. decision trees, naive Bayes, and nearest neighbor). They developed a method for the detection of a large set (20) of ADL. Their approach was performed offline and, considering the large amount of activities included, it achieved promising results regarding the possibility of extracting activity information from accelerometer signals. They also suggested the potential advantages of developing online systems.

Moreover, other wearable systems have particularly grabbed the attention of the HAR research community [Lee and Mase, 2002; Lukowicz et al., 2004; Mantyjarvi et al., 2001] due to the ease of obtaining activity information (e.g. body motion, temperature and heart rate)

directly from the user, unobtrusively and virtually at any location without the need of fixed infrastructure. For example, [Ravi et al., 2005], as opposed to [Bao and Intille, 2004], used only one body-worn triaxial accelerometer in the pelvic region, and evaluated a set of base-level and meta-level classifiers (e.g. boosting and bagging) for improving the recognition performance. Furthermore, in [Lukowicz et al., 2004], an approach to recognized workshop activities was proposed. It used accelerometers in combination with microphones strategically located on the user’s dominant arm. They analyzed the intensity of the acoustic signals and correlated them with the inertial data in order to infer user activities. This approach, which achieved accuracy levels of 84.4%, showed that the fusion of different sensors can greatly improve the classification accuracy of their system.

More recently in [Kwapisz et al., 2011] six human activities such as jogging, walking and sitting were classified using a smartphone-embedded accelerometer carried in the pocket in an attempt to simplify the recognition process with a more pervasive, practical and unobtrusive approach. Other approaches have also been proposed targeting specific applications: for example, from the medical standpoint, monitoring systems have been presented for the detection of different attributes in elder PD patients such as gait parameters, motion disorders and falls using on-body accelerometer [Herrlich et al., 2011; Sama et al., 2012].

In Table 3.3, a summary of the most influential offline HAR works is presented. It highlights the most important aspects about them such as type of sensor used, activities identified and system accuracy levels. In some of these works, system accuracy was not provided, but instead, measures such as sensitivity and specificity. For comparative purposes, we assumed in these cases a balanced dataset with the same number of positive and negative samples in order to approximate this measure as $accuracy = \frac{1}{2} (sensitivity + specificity)$. In the same manner, we present in Table 3.4 the most relevant online HAR systems. In the tables, it can be noticed the large diversity of sensors, types of activities, number of subjects in the experiment and learning approaches that has been employed to achieve the recognition of activities. These differences clearly produce variations in the algorithms performance. For example, the number of people involved in an experiment is an important factor during learning: the higher this number is, the more realistic representation of the overall population (or target group) is obtained. This can therefore increase the generalization ability of a recognition system in order to correctly classify the activities of new unseen persons (e.g. the work in [Lukowicz et al., 2004] performed experiments with a single individual. Even though it achieved a high accuracy it is possible that their system performance would degrade when tried on new people). Likewise, the system accuracy decreases when more activities are added into the system for classification.

The HAR systems proposed in this thesis are included at the bottom of the Tables 3.3 and 3.4. They are identified by their corresponding system name. We use smartphones as a wearable device located on the waist which contains two inertial sensors (the accelerometer and gyroscope) for the extraction of activity information. With regard to human activities, we classify 6 different BAs and also consider the effects 6 PTs. Our learning approach is based on supervised SVMs which is a data-driven model as it is built from the HAR dataset. Lastly, we explore both online

HAR System	Sensor			Activities		P*	Machine Learning Algorithm	Computing Device	Acc.
	No.	Type	Location	No.	Type				
[Bao and Intille, 2004]	5	2D-acc	Thigh, ankle, arm, wrist, hip	20	ADL	20	NB, DT, k-NN	PC	84%
[Ravi et al., 2005]	1	3D-acc	Pelvic region	8	Locomotion, ADL	2	SVM, NB, DT, k-NN, bagging, boosting, stacking, plurality voting	PDA, PC	73.3%-99.8%
[Lukowicz et al., 2004]	5	3D-acc, Microphone	Arm, chest	8	Workshop	1	Intensity Analysis (IA), LDA, HMM, majority voting: (IA+LDA)	Laptop, PC	95.5% - 100%
[Kwapisz et al., 2011]	1	3D-acc	Pocket	6	Locomotion	29	DT, LR, ANN	Smartphone, PC	91.7%
[Mannini and Sabatini, 2010]	5	2D-acc	Thigh, ankle, arm, wrist, hip	7	Locomotion	13	NB, DT, k-NN, ANN, GMM, cHMM	PC	92.2%-98.5%
[Lee and Mase, 2002]	2	2D-acc 1D-gyro	Waist	5	Locomotion	8	Threshold-based	PC	92.9%-95.9%
[Allen et al., 2006]	1	3D-acc	Waist	8	Locomotion, PTs	6	GMM, threshold-based	PC	71.1%-91.3%
[Najafi et al., 2003]	2	1D-acc, 1D-gyro	Chest	6	Locomotion, PTs	11	Threshold-based	PC	94.3%
[Salarian et al., 2007]	3	2D-acc, 1D-gyro	Trunk, shanks	6	Locomotion, PTs	20	Threshold-based, Fuzzy Logic	Data logger, PC	95.4%
[Rodríguez-Martín et al., 2013b]	1	3D-acc	Waist	9	Locomotion, PTs	31	Threshold-based, SVM	Microcontroller, PC	90.5%
[Wu et al., 2012]	2	3D-acc, 3D-gyro	Arm, pockets	13	Locomotion	16	k-NN	Smartphone, PC	90.2%
[Altun and Barshan, 2010]	5	3D-acc, 3D-gyro, 3D-Mag	Chest, arms, legs	19	ADL	8	BDM, LSM, k-NN, DTW, SVM, ANN	PC	75.8%-99.1%
[Lee and Mase, 2002]	3	2D-acc, 1D-gyro, 1D-Mag	Waist, pocket	5	Locomotion	8	Threshold-based	PDA, PC	92.8%-19.91%
[Lara et al., 2012]	6	3D-acc, vital signs	Chest	5	Locomotion	8	NB, DT, MLP, LR, bagging, boosting	Smartphone, PC	92.0%
[Nham et al., 2008]	1	3D-acc	Pocket	4	Locomotion	4	GDA, SVM	Smartphone, PC	93.88%
HF-HAR	1	3D-acc	Waist	6	Locomotion	30	MC-HF-SVM	Smartphone, PC	89.0%

*P: Number of participants

Table 3.3: Summary of existing offline HAR systems.

and offline approaches for the recognition of activities.

Various surveys regarding HAR systems have been presented in the literature covering general approaches [Chen et al., 2012a], or more specific ones such as focusing on wearable sensors [Lara and Labrador, 2012a], on-body accelerometers [Mannini and Sabatini, 2010], smart environments [Cook and Das, 2007]. In the following subsections, we focus on some of these HAR systems attributes in order to analyze them independently against our proposed approaches.

3.3.1 Human Activity Type

Human activities can be categorized based on complexity and area of application as described in Section 3.2.1. Related HAR works have directed efforts in these two directions. Most of them, however, have focused on the study of locomotion activities as it can be seen in Table 3.3 and Table 3.4. Within this group, we have found that static postures such as sitting and standing are commonly investigated along with some dynamic activities such as walking and climbing stairs. In [Karantonis et al., 2006], for instance, the user posture is detected using a waist-mounted accelerometer and it is then used for early stage decisions in a human classifier of dynamic movements (e.g. walking, running, cycling). Moreover, [Allen et al., 2006] developed a classifier which combines 3 static postures and 5 movements using a rule-based heuristic system and a Gaussian Mixture Models (GMM).

HAR System	Sensor			Activities		P*	Machine Learning Algorithm	Computing Device	Acc.
	No.	Type	Location	No.	Type				
[Karantonis et al., 2006]	1	3D-acc	Waist	10	Locomotion, falls		Threshold-based	Microcontroller, PC	90.8%
[Sama et al., 2012]	1	3D-acc	Waist	3	PD motor symptoms	20	SVM	Microcontroller	94%
[Brezmes et al., 2009]	1	3D-acc	Pockets	6	Locomotion, falls	N/A	k-NN	Smartphone	80%
[Bruno et al., 2013]	1	3D-acc	Wrist	8	ADL, PTs, AAs	16	GMM, GMR	PC	68%
[Maurer et al., 2006]	4	2D-acc, microphone, light sensor, thermometer	Wrist	6	Locomotion	6	DT, k-NN, NB	Watch	92.8%
[Lara and Labrador, 2012b]	7	3D-acc, vital signs, GPS	Chest	3	Locomotion	8	DT	Smartphone	92.64%
[Ermes et al., 2008]	3	3D-acc	Chest, wrist, ankle	5	Locomotion	3	DT	PDA	94%
[Riboni and Bettini, 2011]	3	3D-acc, GPS	Wrist, pocket	10	Locomotion, ADL	6	NB, DT, LR, SVM	Smartphone, server	93%
[Tapia et al., 2007]	6	3D-acc, heart rate	Wrist, arm, ankle, thigh, chest, hip	30	Gymnasium	21	NB, DT	Laptop	58.6%-97.6%
L-HAR	2	3D-acc, 3D-gyro	Waist	6	Locomotion	30	MC-L1-L2-SVM	Smartphone	96.4%
PTA-HAR	2	3D-acc, 3D-gyro	Waist	7	Locomotion, PTs	30	MC-L1-SVM	Smartphone	96.7%

*P: Number of participants

Table 3.4: Summary of existing online HAR systems.

Considering the classification of activities based on complexity, most of the works have been concentrated on short events and basic activities. For example, in [Ravi et al., 2005], 8 basic activities related to locomotion and ADL are classified using four different learning algorithms. Additionally, transitions and basic activities are classified simultaneously in [Najafi et al., 2003]. Not much work has been done on complex activities or activity sequences. Primarily because of the difficulty of developing automatic sensor signal segmentation into activities and the high computational cost involved which takes into account past events. In [Lukowicz et al., 2004], a method for continuous recognition of activities based on large decision windows after small window classification is proposed. It is a first step in the study of more complex sets of activities. They apply Linear Discriminant Analysis (LDA) and HMM over audio and inertial signals for developing their classification model.

Some systems have focused on the detection of specific events that occur to people. For example, in healthcare, falls are commonly studied activities. Their detection is essential in order to assist people with any type of limitation such as the elderly (e.g. [Li et al., 2009]). Other activities are directly related to particular illnesses. In the case of PD patients, symptoms such as bradykinesia, freezing of gait, and on/off states are also relevant and therefore their detection have been previously investigated such as in [Sama et al., 2012], [Bachlin et al., 2010] and [Takač et al., 2013].

Most of the HAR systems do not consider PTs as part of their activity set, even if these frequently occur during the transitions from one activity to another (Refer to Section 7.1 for further information). Among the works which have studied PTs we have the following: [Khan et al., 2010b], which chose 3 SPs, 4 AAs and 7 PTs for their study. Furthermore in [Najafi et al., 2003], a comprehensive study of signals occurring during PTs for healthy and elderly

people is used for the classification of activities. In [Salarian et al., 2007], the detection of sit-to-stand and stand-to-sit transitions were crucial for the distinction between standing and sitting. This was achieved through a fuzzy logic classifier which required past and future transition information for this task. Finally in [Rodríguez-Martín et al., 2013b], a hierarchical structure of classifiers was employed to distinguish stand-to-sit and sit-to-stand PTs in patients with PD through measurements from a triaxial accelerometer located on the waist. Lastly, in the medical field, the detection of PTs is sometimes necessary such as in the work of [Mellone et al., 2012] where the *Timed Up and Go* test, used for the assessment of balance and mobility in patients with motor problems, is automated using smartphones. The objective measure of this time required the detection of stand-to-sit and sit-to-stand transitions. In this thesis we first explore methods for the detection of 6 BAs linked to locomotion, and then we also propose an approach which considers PTs as their detection can also enhance the correct prediction of the first set of activities.

3.3.2 Sensor type and Smartphones

There is currently a wide range of sensors available that have been used for developing HAR systems. In this review, we highlight the ones exploited for wearable sensing. Inside this category, we find that sensors can come as a self-contained device, or take part of specific purpose hardware along with other sensors (e.g. [Sama et al., 2012]), or be interconnected as nodes forming a Body Sensor Network (BSN) [Yang and Yacoub, 2006]. Moreover, they can also be embedded elements within other portable devices, such as smartphones or Personal Digital Assistants (PDAs). In this section, we contrast the various wearable approaches used for gathering activity-related signals. Previous HAR works selected accelerometers as their preferred inertial sensor, e.g. [Allen et al., 2006; Ravi et al., 2005; Sama et al., 2012]. However, they are generally employed in cooperation with other sensors such as gyroscopes, microphones and vital signs sensors, in order to contribute with additional information in the recognition process [Lara et al., 2012; Lee and Mase, 2002; Lukowicz et al., 2004].

Previous work has used configurations of multiple sensors located in different body parts, generally ranging from 1 to 5. Although the use of numerous sensors could improve the performance of a recognition algorithm, it is unrealistic to expect that the general public will use them on a regular basis as their obtrusiveness is relatively high and it becomes tedious to wear them. For instance, in [Bao and Intille, 2004], where a set of five biaxial body-worn accelerometers was used, a drawback was evident regarding the number of sensors they needed to attach around the body. The same occurs with other works such as [Lukowicz et al., 2004; Mannini and Sabatini, 2010; Salarian et al., 2007]. The trend shows that in recent years, HAR approaches are aiming to reduce obtrusiveness, either by using less wearable sensors, by opportunistically gathering signals from commonly used devices (e.g. smartphones) or by even fusing sparse sensor data from environmental sensors when available [Bahle et al., 2013]. In our approaches (e.g. L-HAR, PTA-HAR), we only use a single device with only two embedded inertial sensors (accelerometer

and gyroscope) and explore how it is possible to achieve HAR with them.

Research efforts have also concentrated on exploiting smartphones for HAR. Some smartphone-based approaches have been already proposed in the literature [Berchtold et al., 2010; Brezmes et al., 2009; Kwapisz et al., 2011; Wu et al., 2012]. In [Kwapisz et al., 2011], for example, it was presented one of the first approaches to exploit an Android smartphone and its embedded triaxial accelerometer for HAR. Their approach was able to classify six locomotion activities over intervals of 10 seconds using an ANN while the device was carried in the pockets. In [Nham et al., 2008], the mode of transport (walking, running, cycling and driving) was predicted using accelerometer data from an iPhone through a signal Fourier analysis and an SVM classifier. In a similar way, Brezmes et al. in [Brezmes et al., 2009] implemented a real-time activity classifier to detect 6 different states on a Nokia mobile phone. Research on HAR with smartphone has mostly incorporated only accelerometers [Lara and Labrador, 2012a; Mannini and Sabatini, 2010]. This can be explained due to the fact that these embedded inertial sensors were the first to be introduced in the mobile phone market (2007) [Lane et al., 2010]. Gyroscopes, on the other hand, made a late appearance a few years later (2010), though less approaches have considered them for HAR (e.g. [Altun and Barshan, 2010; Lee and Mase, 2002]). Recently, in [Wu et al., 2012], a hybrid accelerometer and gyroscope approach was used for the classification of 9 activities using an iPhone 4. They showed insights of the benefits of adding gyroscope signals into the recognition system achieving improvements ranging from 3.1% to 13.4% in classification accuracy.

3.3.3 Machine Learning

Machine Learning approaches that have already been applied for the recognition of activities include: NB [Jatoba et al., 2008], HMM [Mannini and Sabatini, 2010], DT, and SVM [Maurer et al., 2006]. Some HAR works have also compared various ML methods to find the most suitable approach in their applications. However, they do not coincide with a specific best solution and have found instead heterogeneous approaches (e.g. DT, GMM, k-NN, etc.). This finding shows a possible dependency of the algorithm selection on the type of application and data used. Moreover, meta-classifiers, which predict based on the output of a series of base-level classifiers, has also been used in HAR systems. In [Ravi et al., 2005], for instance, plurality voting is preferred as the best approach for the classification of activities from the output of standard classifiers including SVM, NB, k-NN and DT. The use of these classifiers is therefore interesting in order to improve the overall recognition performance of a HAR system at the expense of carrying out more complex operations which can lead to limitations with respect to battery consumption, shared resources, and real-time operation.

Our approach exploits SVMs for the classification of activities similarly to other works which have successfully employed them [He and Jin, 2009; Khan et al., 2010a; Maurer et al., 2006]. Furthermore, SVMs have shown to be effective in heterogeneous types of recognition such as in handwritten characters [LeCun et al., 1995] and speech [Ganapathiraju et al., 2004]. In this

thesis, we show in Section 4.4.2 that they outperform other ML approaches when our HAR dataset is used. SVMs provide a good compromise between accuracy and training time while also count with a variety of publicly available learning tools for experimentation such as LIBSVM [Chang and Lin, 2011]. However, although characterized by several appealing attributes, one of its drawbacks consists in its naïve two-class nature, that makes generalization to multiclass problems (as in the typical case of HAR) not straightforward. Different approaches have been explored for targeting this issue [LeCun et al., 1995]. The two most commonly used methods are: OVA and OVO as seen in Section 2.5.3.2. We selected, in particular, the OVA method for our research.

Adaptive HAR systems have also been explored in the literature. Their purpose is to take classification algorithms and adapt them to a particular user in order to personalize it to the user characteristics making it more robust and accurate. Although, in this thesis we do not perform adaptation, some approaches that have already proposed them are: [Zhao et al., 2010], a cross-people AR technique is described. They transfer classification knowledge to a new user without data collection by adapting the activity labeled samples of an initial model using a binary DT model. Also in [Allen et al., 2006], a method for user adaptation based on GMM is proposed to compensate the problem of limited training data for each test subject.

3.3.4 Offline and Online

HAR system can also be classified into two main groups depending on the response time that systems take to perform activity classification [Lara and Labrador, 2012a]: *online* methods aim for real-time prediction of activities while, conversely, *offline* methods usually need extra processing time, use computationally demanding algorithms or simply they do not require real-time operation. In the first case, many of these approaches focus on the detection of a small group activities (generally SPs and AAs) because an increment in the number of activities or their complexity might not allow to achieve real-time operation.

Various previously proposed systems have required external computing support for allowing online recognition capability. This has been done by transmitting body motion data in real-time to a fixed device. For example, a real-time system for the detection of basic SPs and AAs was described in [Karantonis et al., 2006]. It used a waist-mounted wireless unit composed of a microcontroller and an accelerometer which transmitted processed inertial signals to a local computer for activity evaluation and display. Furthermore, an online approach was introduced in [Tapia et al., 2007]. It classified 30 physical gymnasium activities and their intensities using 3 accelerometers and a heart rate monitor. Signals were also wirelessly transmitted to a laptop for processing and classification. In [Ermes et al., 2008], motion bands with an embedded accelerometer attached to the user's wrist, ankle and chest transmitted via Bluetooth link the inertial signals to a PDA for the classification of fitness-related activities. Nowadays PDAs are nearly obsolete and are being replaced with smartphones [Liu et al., 2011].

Smartphones have simplified the online implementation of many applications as they inte-

grate sensing and computing capabilities, in contrast to the aforementioned approaches which had to rely on ad-hoc implementations to establish the recognition pipeline using different devices. First smartphone-based HAR attempts were *offline* and only recorded the inertial data using the device sensors or other linked devices and later processed the signals/data for activity classification. For example: in [Lara et al., 2012] the *Centinela* system was presented. It consisted of a chest unit composed of several sensors to measure acceleration data and vital signs (e.g. heart rate, breath amplitude, respiration rate) and a smartphone wirelessly connected via Bluetooth. Data was later processed and classified offline using different ML algorithms showing how vital signs can also add value to the system performance. Kwapisz et al. [Kwapisz et al., 2011], which was previously mentioned, also developed an offline smartphone-based HAR system for the classification of 2 SPs and 4 AAs. In the work proposed in [Nham et al., 2008], they are able to perform offline classification of four modes of transportation (biking, running, walking and driving) using an iPhone accelerometer. Similarly, in this thesis we propose the HF-HAR system which is able to perform offline the recognition of 6 activities from data gathered from smartphones using a hardware-friendly approach (Chapter 5). The ML method uses a modified multiclass SVM with fixed-point arithmetic prediction aiming to obtain a fast implementation more suitable for battery operated devices.

More recently, other contributions have proposed *online* smartphone-based HAR systems. A Nokia smartphone was used in [Brezmes et al., 2009] for the online recognition of 6 activities. In the same way, the work presented in [Kose et al., 2012] used an Android OS smartphone with embedded accelerometer for the classification of 4 activities. The training stage was also performed online through the option of collecting live data from the users by following a predefined activity protocol. [Lara and Labrador, 2012b] proposed an improved version of the work in [Lara et al., 2012] for the recognition of activities in real-time. To conclude, [Riboni and Bettini, 2011] proposed a HAR system which combined the smartphone internal accelerometer with an external one on the user's wrist for the classification of ADL. They proposed a context-aware framework based on ontologies, reasoning and statistical inferencing to solve scalability problems when the number of activities becomes large.

3.4 Summary

In this chapter we discussed essential aspects regarding the recognition of human activities in the light of the existing research literature. It included common strategies for the development and evaluation of HAR systems. Additionally, relevant up-to-date HAR approaches are highlighted and compared against the methods proposed in this thesis.

There are already several alternatives that deal with the problem of recognizing activities which have similarities between them in various aspects (e.g. the types and number of activities identified, sensors used, ML approaches and real-time prediction). Nevertheless, we explore in this thesis some unique aspects in this field which have not yet been fully covered in the literature. They include elements such as the evaluation of hardware-friendly approaches for

the recognition of activities using fixed-point arithmetic, the study of hybrid linear *SVM* models for their application in *HAR*, and the awareness of postural transitions in real-time recognition systems. These will be thoroughly covered in the forthcoming chapters.

Part II

Data Collection and Offline Activity Recognition

Chapter 4

Human Activity Dataset Generation

4.1 Introduction

Two of the vital elements required for developing our research in smartphone-based HAR are: experimental data collection and dataset generation. In this chapter, we describe a general overview of these elements organized in three main sections. Section 4.2 describes all the aspects linked to data gathering, first by selecting the appropriate smartphone for experimentation, and then by describing the protocol of trials with volunteers. Moreover, Section 4.3 deals with the raw sensor data in order to generate the dataset. For this task, we use signal conditioning techniques and select a suitable set of features for data characterization. In Section 4.4 preliminary results with the available data are presented. This includes data validation with different ML algorithms that confirm their usability and data publication in a centralized repository. We also describe an organized HAR competition with the obtained dataset in which people were encouraged to propose their own solutions to the recognition problem. Finally, the chapter is summarized in Section 4.5.

4.2 Experimental Data Collection

In the HAR research framework, some benchmark datasets have been released to the public domain providing experimental data with various inertial sensors. They provide a freely available source of data across different disciplines and researchers in the field. For example, the *Opportunity* Project [Roggen et al., 2010] has recorded a set of ADL in a sensor-rich environment using 72 environmental and body sensors. Similarly, other works have provided datasets such as [Tapia et al., 2006] and [Dernbach et al., 2012].

However, public smartphone inertial data for activity recognition is limited. For this reason, we chose to make our own collection of data and also to make it available online as a research resource. The data have been collected from one experiment which is thoroughly described in the following section.

4.2.1 Smartphone Selection

Finding a suitable smartphone to perform HAR involves the evaluation of the up-to-date devices in the market following the selection criteria listed in Section 2.4.1, in which hardware, sensors and software are the primary elements to be considered. Also, other aspects such as price, market of available brands and potential future distribution of applications were taken into account. However, by the time this selection process was done (2011), the smartphones with the required embedded sensors were very limited, as well as their specifications regarding speed and direct access to the sensors through their OS. This motivated the assessment of only the available high-end devices at that time. As it is known with the accelerated growth of current technologies, it was just a matter of two years to discover that now in almost every mid- and high-end smartphone in the market the required sensors and hardware with even higher specifications were available. This finding expands the range of applications of the developed approaches of this work to a greater group of devices.

4.2.1.1 Up-to-date Smartphones

For the selection of the smartphone regarding sensor type, we took as a reference point an already available inertial sensor, the 9X2 [Rodríguez-Martín, 2010], which had already demonstrated its applicability in the detection of various human motor disorders in people with disabilities (e.g. PD patients) [Rodríguez-Martín et al., 2013a,b]. It is internally composed of three triaxial sensors: an accelerometer, a gyroscope and a magnetometer. It collects inertial sensor data that can be stored on a microSD memory card or wirelessly transmitted to a terminal via Bluetooth. We expected to find smartphones with those minimum characteristics. Figure 4.2 shows an image of the sensor.

Table 4.1 presents a comparison of the 9X2 sensor with two high-end smartphone candidates. It includes their characteristics regarding hardware and software. The embedded accelerometer and gyroscope smartphone specifications outperform the ones from the reference inertial sensor regarding speed and operation ranges, making them potentially suitable for their application in HAR. However, even though the specified frequencies are provided by the devices datasheets, the net frequency of these inertial sensors is limited by the smartphones OS which regulates their operation speed and battery consumption. It is around 100Hz and it can vary depending on the OS load, but it is sufficient to properly capture human body motion according to [Karantonis et al., 2006] (refer to Section 4.3.2). Regarding memory and Central Processing Unit (CPU) characteristics, both smartphones share similar characteristics. As a result, the device selection was consequently subject to additional factors.

The selected smartphone for carrying out experiments on HAR was the SGSII. It is managed by the *Android OS* which is an open source platform with a publicly distributed development environment which includes a large set of APIs allowing access to the smartphones hardware and internal sensors, robust ML tools and its publication policies for apps are relatively simple to fulfill. Moreover, developing on this OS, also extends the utilization of HAR apps to a wider

Device	9X2	iPhone 4	I9100 Galaxy S II
Brand	CETpD	Apple	Samsung
CPU	Microchip dsPIC33	ARM Cortex-A8	ARM Cortex-A9
ROM Memory	80MHz	1 GHz	1.2GHz dual-core
RAM Memory	2GB	16GB/32GB	16GB/32GB
Operating System	16KB	510MB	1 GB
	N/A	iOS 4	Android OS v2.3
Accelerometer	STMicroelectronics	STMicroelectronics	Bosch Sensortec
	LIS3LV02DQ	LIS331DLH	SMB380
Max Frequency	640Hz	1KHz	1.5KHz
Operation Range	$\pm 2/6g$	$\pm 2/4/8g$	$\pm 2/4/8g$
Gyroscope	InvenSense	STMicroelectronics	STMicroelectronics
	IDG650, ISZ650	L3G4200D	L3G4200D
Max Frequency	140Hz	800Hz	800Hz
Operation Ranges	$\pm 440/2000^\circ s^{-1}$	$\pm 250/500/2000^\circ s^{-1}$	$\pm 250/500/2000^\circ s^{-1}$
Display	N/A	LED-Backlit IPS TFT	Super AMOLED Plus
External Card	Yes	No	Yes
Wireless	Bluetooth	3G,WLAN,Bluetooth	3G,WLAN,Bluetooth
Battery Type	Li-Ion 1000mAh	Li-Po 1420mAh	Li-Ion 1650 mAh
Battery Stand-by	Up to 18 h	Up to 300 h	Up to 610 h

Table 4.1: Smartphones and 9X2 sensor specifications

range of brands and devices (e.g. tablets, notebooks) which also work under *Android OS*, unlike *iOS* which only operates on devices of only one brand. Figure 4.1 depicts an image of the selected device.

Android OS has also a leading position in the market which is advantageous because it can also contribute with an easier distribution of final applications to a larger population sector. Smartphone market shares in 2013 [Gupta et al., 2013], for example, showed that the three top mobile OSs were: *Android OS*, *iOS* and *Windows Phone*. They together make up the vast majority of the phone market reaching a 96%. *Android OS* controls a remarkable 79% of the market share on its own.

4.2.1.2 App Development

All the smartphone apps of this work were built using a software solution for Android development (Android Development Tools (ADT) Bundle) which integrates a collection of various programs [Android, 2013]:

- Eclipse: is an integrated environment for the development of software projects with multi-language support.
- ADT plug-in: is the toolset for Eclipse designed to allow the development of Android Apps.
- Android Software Development Kit (SDK): provides the API libraries and developer tools required to build apps for Android.



Figure 4.1: SGSII Smartphone.



Figure 4.2: 9X2 inertial sensor

- **Android Native Development Kit (NDK)**: is the collection of tools that allows to implement apps using native-code languages such as C and C++.

The code was written using two languages, namely, *Java* and *C*. The former was employed for the development of the graphical user interface and app basic controls. *C* was reserved for the computationally expensive tasks such as accessing smartphone sensors, signal processing, running ML algorithms and storing data.

4.2.2 HAR Protocol

A set of experiments were carried out to obtain the HAR datasets. A group of 30 volunteers with ages ranging from 19 to 48 years were selected for this task. The mean age of all volunteers was 28.9 ± 6 years. Each person was instructed to follow a protocol of activities while wearing a waist-mounted SGSII smartphone. The experiment was planned in order to contain six BAs: three SPs (*standing*, *sitting*, *lying-down*) and three AAs (*walking*, *walking-downstairs* and *walking-upstairs*). Moreover, it was arranged with the intention of having also available all the possible PTs that occur between the three existing SPs. These are: *stand-to-sit* (StSi), *SiSt*, *sit-to-lie* (SiLi), *lie-to-sit* (LiSi), *stand-to-lie* (StLi), and *lie-to-stand* (LiSt). Figure 4.3 shows images of one of the experiment participants while carrying out the six BAs.

The protocol of activities is detailed in Table 4.2. It includes all the tasks in sequential order and the time expected to be spent on performing each of them. Each subject performed the protocol of activities twice: on the first trial the smartphone was fixed on the left side of the waist, approximately 45 degrees away from the anterior direction. On the second trial the belt with the smartphone was placed by the user himself as preferred. For the AAs, participants were told to remain standstill in between these tasks in order to facilitate the labeling of the ground truth. This also supported the repeatability of the test as in this way every activity was at least tried twice. Similarly, this period was required after the last SP (*lying-down*) in order to fully capture the LiSt PT.

The duration of the entire experiment was around 15 minutes per person including the setting up of the sensors and the repetition of the protocol. Although the tasks were carried out in laboratory conditions, volunteers were asked to perform freely the sequence of activities aiming to simulate a more naturalistic dataset. All the experiments were recorded on video with the consent of the participants in order to have a ground truth of the performed activities. It was done with the video camera of another smartphone at a frame rate of 30 Hz.

4.2.3 Data recording

MotoLog was the first Android app we developed for capturing and storing data from the smartphone inertial sensors and it was created for carrying out the HAR experiments. It also performs real-time visualization of the inertial signals (accelerometer, gyroscope and magnetometer) on the smartphone screen. The app also allows to visualize the experiment recordings offline. Figure 4.4 shows a screenshot of the app showing online and offline visualization modes.



Figure 4.3: Example of the performed activities during the collection of experimental data. From left to right and top to bottom: standing, sitting, lying-down, walking, walking-downstairs and walking-upstairs

No.	Static	Time (sec)	No.	Dynamic	Time (sec)
0	Start (Standing Pos)	0	8	Walk (1)	15
1	Stand (1)	15	9	Walk (2)	15
2	Sit (1)	15	10	Walk Downstairs (1)	12
3	Stand (2)	15	11	Walk Upstairs (2)	12
4	Lay Down (1)	15	12	Walk Downstairs (1)	12
5	Sit (2)	15	13	Walk Upstairs (2)	12
6	Lay Down (2)	15	14	Walk Downstairs (3)	12
7	Stand (3)	5	15	Walk Upstairs (3)	12
			16	Stop	0
				Total	192

Table 4.2: Protocol of activities for the HAR Experiment.

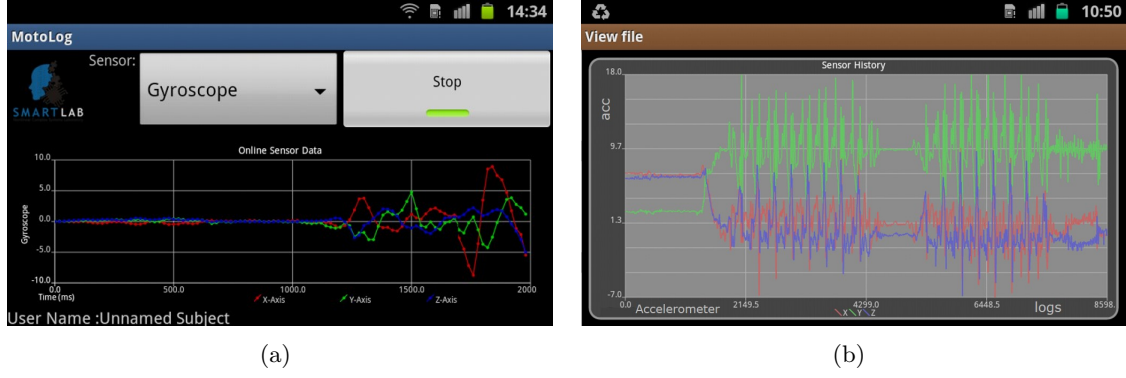


Figure 4.4: Screenshots of the smartphone *MotoLog* app for the recording of the inertial data from smartphone accelerometer, gyroscope and magnetometer. a) Preview of main screen and b) visualization of recorded files.



Figure 4.5: SGSII casing: (a) the smartphone case and belt used for the experiments and (b) arrows show the axis orientation of the inertial sensors.

For its basic operation, the app requests access to the three inertial sensors which periodically provide readings at a frequency equal or higher than a minimum specified (50 Hz in our case). A periodic task is then in charge of collecting these readings from the enabled sensors along with their timestamps. Subsequently, all the inertial data is handled as an output stream and stored in a log file. At the same time a visualization module performs the graphs that display the available data.

The smartphone was located on the users waist using a belt provided with a case as it is shown in Figure 4.5(a). The phone always faced the same direction with respect to the case in order to maintain the same orientation of the triaxial sensors coordinate axis (Figure 4.5(b)).

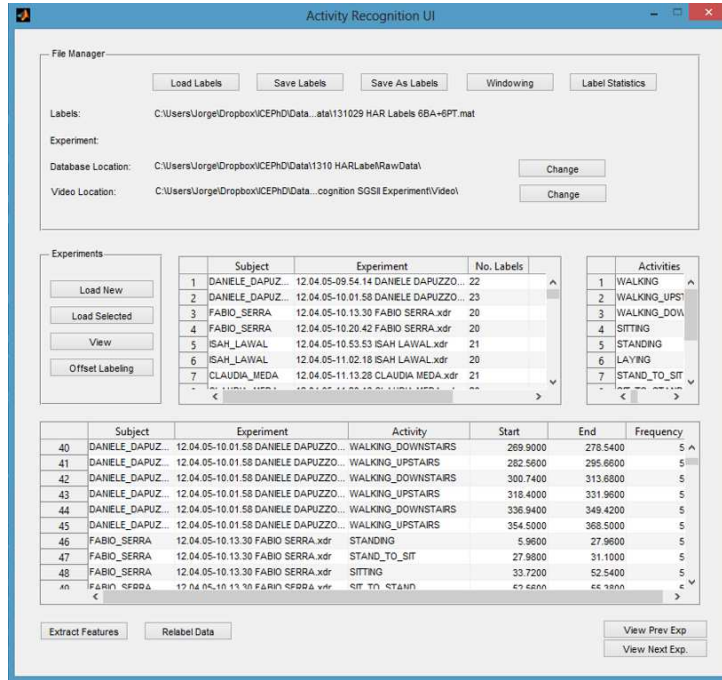


Figure 4.6: *ARGUI* application user interface for handling experiment inertial data and video

4.3 HAR Data Processing

Once data were collected from the experiment, log files and video footage were handled in order to generate the HAR datasets. We developed an application for this purpose (*ARGUI*). It was written in *Matlab* by means of its Graphical User Interface (GUI) tool. Figure 4.6 shows a screenshot of the main interface in which it is possible to visualize the collection of HAR experiments and the main functions available. The main steps involve: activity labeling, signal processing and feature mapping. These are all described in this section.

4.3.1 Labeling

Data were labeled manually and guided by the video footage recorded during the experiment execution. The first step consisted on synchronizing the video and inertial signals. This task was done aided by the user interface which allowed to visually inspect them. The occurrence of the first PT was used as a reference event to find the time shift between the video and inertial signals. A verification step was made by examining the last PT of each experiment. Figure 4.7 shows the labeling interface with an example of the synchronization process.

After synchronization, the start and end times were manually set for each performed activity (BAs and PTs) using the labeling application. All the labels for all the experiments were collected in a file (labels file) which was used as one of the inputs for the dataset generation process.

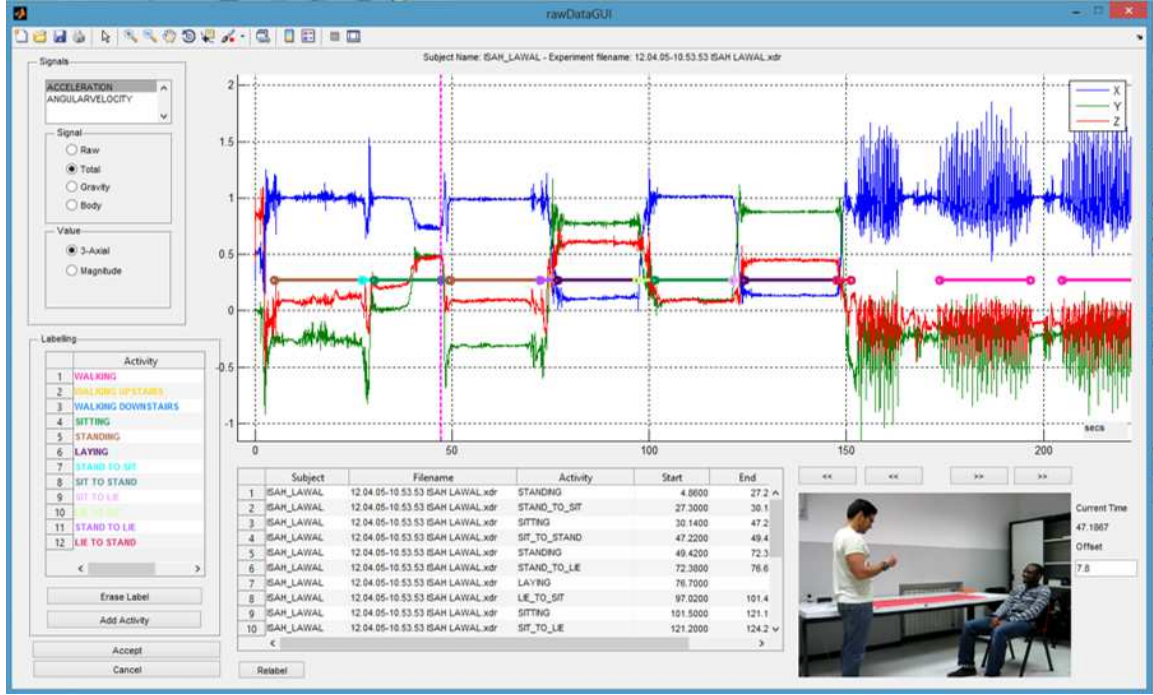


Figure 4.7: Labeling interface: visualization of inertial signals and video

4.3.2 Signal Processing

Raw sensor signals from the accelerometer ($\mathbf{a}_r(t)$) and the gyroscope ($\boldsymbol{\omega}_r(t)$) were preprocessed by the application of a series of filters for conditioning. First, *noise reduction* was achieved with a third-order median filter and then followed by a third-order low-pass Butterworth filter with a cutoff frequency of 20 Hz. This frequency threshold was selected from the work presented in [Karantonis et al., 2006] which states that the energy spectrum of the human body motion lies mainly within the range of 0 Hz to 15 Hz. From these processes, a clean triaxial total acceleration ($\mathbf{a}_r(t)$) and an angular velocity ($\boldsymbol{\omega}_t(t)$) signals were obtained. The application of these filters is represented by the $H_1()$ transfer function. The example presented in Figure 4.8 shows the raw triaxial acceleration while a subject is walking and then walking-upstairs.

The acceleration signal is further processed as it is interpreted as the combined effect of the gravitational force and the acceleration due to body motion. Therefore, assuming that the gravitational component only affects the lowest frequencies, it is possible to separate the body motion acceleration signal ($\mathbf{a}(t)$) through high-pass filtering $\mathbf{a}_r(t)$ with a cutoff frequency of 0.3 Hz. This threshold was calculated by varying the cutoff frequency from 0.0 to 1.0 Hz in small increments of 1/40 Hz and estimating the minimum square error of the filtered gravity signal minus the standard gravity constant (9.81 m/s^2) using the experimental data. This findings were similar to those in [Karantonis et al., 2006]. The segmentation of the body acceleration is represented with the transfer function $H_2()$. Finally, the gravity signal ($\mathbf{g}(t)$) can be found by

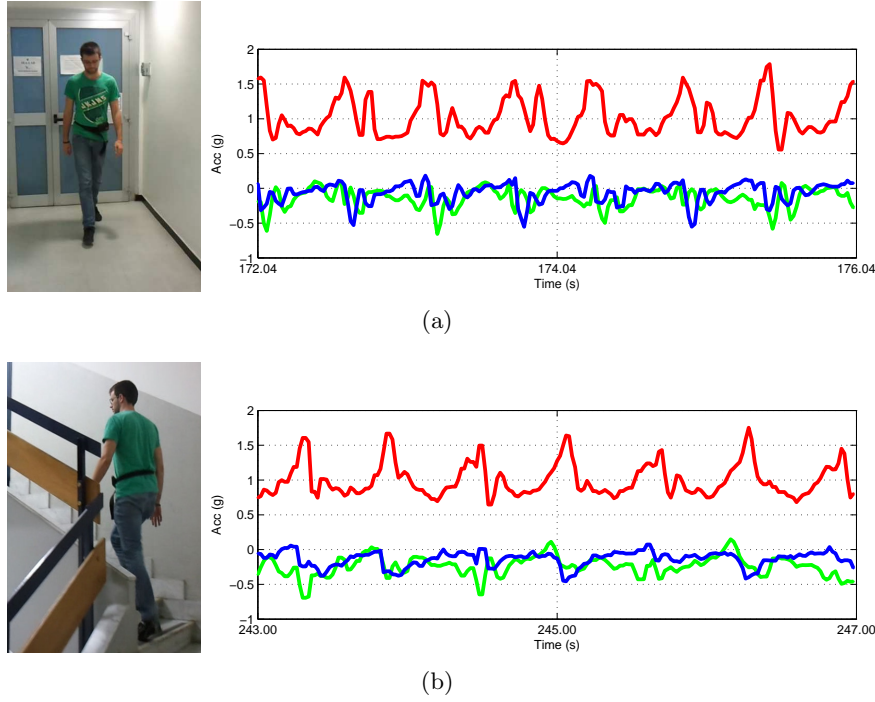


Figure 4.8: Example of inertial signals from the accelerometer while performing two activities: (a) walking, (b) walking upstairs.

using the total acceleration and body acceleration in the following way:

$$\mathbf{g}(t) = \mathbf{a}_\tau(t) - \mathbf{a}(t). \quad (4.1)$$

Other works have also separated body acceleration and gravity in a similar way such as in [Bruno et al., 2012].

Moreover, $\boldsymbol{\omega}_\tau(t)$ is also high-pass filtered in order to remove any DC bias affecting the gyroscope as it is one of the possible calibration errors that can be found in these sensors. After filtering, using the same frequency values as in $H_2()$, the $\boldsymbol{\omega}(t)$ signal was obtained.

The outcome after noise filtering and signal segmentation consists of three signals: $\mathbf{a}(t)$, $\mathbf{g}(t)$ and $\boldsymbol{\omega}(t)$. They are informative about the user's body motion, the person's orientation (e.g. helpful for the distinction of *lying-down* and *standing* states), and motion patterns people have for performing some activities (e.g. for recognizing AAs and PTs). An additional transformation is performed to $(\mathbf{a}(t)$ and $\boldsymbol{\omega}(t))$. This is the derivative with respect to time ($\mathbf{a}'(t)$ and $\boldsymbol{\omega}'(t)$) which has shown to be informative in order to extract relevant activity-related features and it has already been successfully used in some applications such as in the detection of ON/OFF states on PD patients [Samà et al., 2011]. Lastly, the magnitude (Euclidean norm) is also applied to the triaxial inertial signals in order to obtain $a_{mag}(t)$ and $\omega_{mag}(t)$. A compilation of the signal transformations is presented in Table 4.3.

Following these steps, the signals are segmented into window samples which are the activity unit of this work. In this manner, every window unequivocally has an associated activity.

Name	Symbol	Formulation
Total Acceleration	$\mathbf{a}_\tau(t)$	$H_1(\mathbf{a}_\tau(t))$
Body Acceleration	$\mathbf{a}(t)$	$H_2(\mathbf{a}_\tau(t))$
Gravity	$\mathbf{g}(t)$	$\mathbf{a}_\tau(t) - \mathbf{a}(t)$
Body Jerk	$\mathbf{a}'(t)$	$d(\mathbf{a}(t))/dt$
Body Acc Magnitude	$a_{mag}(t)$	$\ \mathbf{a}(t)\ $
Angular Speed	$\boldsymbol{\omega}(t)$	$H_2(H_1(\boldsymbol{\omega}_\tau(t)))$
Angular Acceleration	$\boldsymbol{\omega}'(t)$	$d(\boldsymbol{\omega}(t))/dt$
Angular Speed Magnitude	$\omega_{mag}(t)$	$\ \boldsymbol{\omega}(t)\ $

Table 4.3: Main signal processing operations applied to the smartphone sensors inertial signals

Window sampling is done in a fixed-width sliding windows fashion. We use a 50% overlap between windows as it has shown to be suitable for various recognition applications [Bao and Intille, 2004; Van Laerhoven and Cakmakci, 2000] in order to avoid missing events and activity truncation. Moreover, we chose an activity window length of 2.56 sec guided by the following reasons:

- The cadence of an average person walking is within $[90, 130]$ steps/min [BenAbdelkader et al., 2002], i.e. a minimum of 1.5 steps/sec;
- At least a full walking cycle is preferred on each window sample, this corresponds to a minimum of two steps;
- People with slower cadence such as elderly and disabled people with motor impairments should also benefit from this method. We supposed a minimum speed equal to 50% of average human cadence;
- Signals are also mapped in the frequency domain through the Fast Fourier Transform (FFT), which is optimized for vectors with power of two length: $N = (2.56 \text{ sec} \times 50 \text{ Hz} = 128 \text{ cycles})$.
- Longer windows sizes were not preferred as they would increase the latency times in the prediction of activities when used online.

The processed inertial signals and the labels file (Section 4.3.1) were used for the extraction and labeling of activity windows. The windowing process has been approached in two ways. In the first method (\mathcal{W}_1), we worked with the start and end times of each activity segment label and the signals defined within this time region. We partitioned them into fixed-width sliding windows and assigned them the corresponding activity label.

In method two (\mathcal{W}_2), we first divided the entire inertial signal into sliding windows (e.g. from the sequence of all the experiment activities) and then we assigned a label to each window using the linked activity segments. This suggests that sometimes uncertainty situations may appear. In particular, when more than one activity segment label from the ground truth overlaps a particular window. To solve this, we defined some conditions to choose the activity label to represent each window. In a window sample, the time length of the involved labels was measured

Name	Time	Freq.
Body Acc	1	1
Gravity Acc	1	0
Body Acc Jerk	1	1
Body Angular Speed	1	1
Body Angular Acc	1	0
Body Acc Magnitude	1	1
Gravity Acc Mag	1	0
Body Acc Jerk Mag	1	1
Body Angular Speed Mag	1	1
Body Angular Acc Mag	1	1

Table 4.4: Time and frequency domain signals obtained from the smartphone sensors.

and it was chosen the one that lasted longer. \mathcal{W}_2 is slightly harder to implement than \mathcal{W}_1 but it is more realistic because it takes into account that activities happen sequentially. It is more useful for its application in online recognition systems where transitions between activities occur. In the following sections we will be specific about which method is used in the generation of the activity datasets.

4.3.3 Feature Mapping and Dataset Generation

A reduced representation composed of features with relevant activity information can be obtained from the activity windows in the time domain. These windows are also transformed into frequency domain with the Discrete Fourier Transform (DFT) using a real-valued FFT algorithm [Duhamel and Vetterli, 1990; Ho, 2004]. In this work, we extract features from these two domains. They include standard measures that have already been proposed for HAR in several works [Yang et al., 2008] such as the mean, correlation between signal pairs, Signal Magnitude Area (SMA) and autoregression coefficients [Khan et al., 2010a], energy of different frequency bands [Samà, 2013]. We also include original measures such as frequency spectrum skewness and kurtosis, and angles between triaxial signals. These measures are applied to the processed accelerometer signals as well as the ones from the gyroscope (Table 4.3). Therefore, considering the amount of signals involved, the generated number of features can largely increase. Table 4.4 shows the selected signals and indicates the domains from which features were extracted.

Table 4.5 shows the measures applied to the signals for generating the datasets along with their formulation over the window signal \mathbf{s} of length N . Numerical indexes set under the \mathbf{s} indicate one of the three possible axis x , y or z . A total of 561 features were extracted to describe each activity window. Some of these features are well-know and their estimation is straightforward. These are the ones in the top box of the table. Others, on the other hand, are here introduced for clarification:

- **Signal Magnitude Area:** this measures helps to identify periods of activity over a triaxial signal in the time domain. It is defined as the sum of the absolute value of all axis

Function	Description	Formulation
mean (\mathbf{s})	Arithmetic mean	$\bar{s} = \frac{1}{N} \sum_{i=1}^N s_i$
std (\mathbf{s})	Standard deviation	$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (s_i - \bar{s})^2}$
mad (\mathbf{s})	Median absolute deviation	$\text{median}_i (s_i - \text{median}_j(s_j))$
max (\mathbf{s})	Largest values in array	$\max_i (s_i)$
min (\mathbf{s})	Smallest value in array	$\min_i (s_i)$
skewness (\mathbf{s})	Frequency signal Skewness	$E \left[\left(\frac{s - \bar{s}}{\sigma} \right)^3 \right]$
kurtosis (\mathbf{s})	Frequency signal Kurtosis	$\frac{E[(s - \bar{s})^4]}{E[(s - \bar{s})^2]^2}$
maxFreqInd (\mathbf{s})	Largest frequency component	$\arg \max_i (s_i)$
energy (\mathbf{s})	Average sum of the squares	$\frac{1}{N} \sum_{i=1}^N s_i^2$
sma ($\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$)	Signal magnitude area	$\frac{1}{3} \sum_{i=1}^3 \sum_{j=1}^N s_{i,j} $
entropy (\mathbf{s})	Signal Entropy	$\sum_{i=1}^N (c_i \log(c_i))$, $c_i = s_i / \sum_{j=1}^N s_j$
iqr (\mathbf{s})	Interquartile range	$Q3(\mathbf{s}) - Q1(\mathbf{s})$
autoregression (\mathbf{s})	4th order Burg Autoregression coefficients	$\mathbf{a} = \text{arburg}(\mathbf{s}, 4)$, $\mathbf{a} \in \mathbb{R}^4$
correlation ($\mathbf{s}_1, \mathbf{s}_2$)	Pearson Correlation coefficient	$C_{1,2} / \sqrt{C_{1,1} C_{2,2}}$, $C = \text{cov}(\mathbf{s}_1, \mathbf{s}_2)$
meanFreq (\mathbf{s})	Frequency signal weighted average	$\sum_{i=1}^N (i s_i) / \sum_{j=1}^N s_j$
energyBand (\mathbf{s}, a, b)	Spectral energy of a frequency band $[a, b]$	$\frac{1}{a-b+1} \sum_{i=a}^b s_i^2$
angle ($\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{v}$)	Angle between triaxial signal mean and vector	$\tan^{-1} (\ [\bar{s}_1, \bar{s}_2, \bar{s}_3] \times \mathbf{v}\ , [\bar{s}_1, \bar{s}_2, \bar{s}_3] \cdot \mathbf{v})$

Table 4.5: List of measures for computing feature vectors. N : signal vector length, Q : Quartile.

divided by the number of samples N in a signal window [Karantonis et al., 2006].

- **Entropy**: this measure of uncertainty commonly used in Information Theory is applied to the signal \mathbf{s} in the frequency domain and gives an estimation of the amount of information this provides. It is estimated using the normalized information entropy of the magnitudes of this signal [Ho, 2004].
- **Interquartile range**: this statistical measure calculates the difference between the upper (Q_3) and lower quartiles (Q_1) of a ranked set of elements. These quartiles are the points that divide the data by 25% and 75% respectively.
- **Autoregression coefficients**: are the coefficients found through the Burg's method that fit an autoregressive model of the input \mathbf{s} [Roth et al., 2003]. This operation is applied to the signal in the time domain and produces an output of 4 features corresponding to the algorithm order such as in [Khan et al., 2010a].
- **Pearson correlation coefficient**: measures the relationship between two signal vectors \mathbf{s}_1 and \mathbf{s}_2 . This statistical measure is bounded between the interval $[-1, 1]$ and denotes either a positive or negative correlation when this value is close to 1 or -1 respectively, and no correlation when it is close to 0. We apply this operation to the time signals from pair of axis from the accelerometer or gyroscope.

- **Signal weighted average:** this measure gives the average frequency of a signal \mathbf{s} considering that each point contributes in proportion to its magnitude.
- **Spectral energy of a frequency band:** this function returns an energy measure in a similar way to the energy (\mathbf{s}) function but only within an interval of the frequency signal. We selected contiguous intervals starting from zero with three different bandwidths (8, 16 and 24 points).
- **Angle between vectors:** this measure gives an estimation of the angle between the mean value of a triaxial time signal and a another vector (e.g. to measure the angle of the average acceleration with respect to the y axis $[0, 1, 0]$). It is useful for determining the average orientation of the smartphone in every activity window.

Features were normalized in order to lie within the $[-1, 1]$ range following this formulation:

$$\mathbf{x} = 2 \frac{\mathbf{x}_0 - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}} - 1, \quad (4.2)$$

where \mathbf{x}_0 is the original vector of features and the \mathbf{x}_{min} and \mathbf{x}_{max} vectors respectively contain the minimum and maximum values for each feature from the available data.

In order to ease the performance assessment, the dataset was randomly partitioned into two independent sets, where 70% of the data were selected for training and the remaining 30% for testing. This partition was done at the level of number of people and not the number of samples in order to guarantee that samples from both groups were completely independent. Moreover, the protocol of activities was planned to provide a balanced number of sample per each BA, however during the extraction of windows samples, classes resulted with a slightly different number of activities between them producing a nearly-balanced set. We preserved all the samples and avoided using balancing methods such as homogeneous multi-partitioning approaches [Aupetit, 2009]. Instead we worked with the parameters of the ML algorithms used to balance the data when required.

In this work, different partitions of the dataset are used for specific purposes. Therefore, in order to simplify the understanding in future sections, we collect in Table 4.6 these partitions as well as their characteristics including type of sensors, feature domains, activities, number of features and window sampling method. They have been named with the \mathcal{D} symbol followed by a number which represents a chronological order. We started with a simple version of the dataset for research (\mathcal{D}_1). It has a limited number of features (17) which are extracted from accelerometer data and it has fixed-point number representation. Following this, we added the gyroscope and extended the number of features in the time and frequency domain (\mathcal{D}_2). Moreover, as we will see in Chapter 6, we also consider the use of only time domain features and we make distinction of this particular subset as \mathcal{D}_{2T} . Finally the \mathcal{D}_3 and \mathcal{D}_{3T} partitions include PT information and modifies the window sampling method required for the implementation of an online HAR system in Chapter 7.

Table 4.6: Dataset partitions generated from the HAR experiment. * FP: Fixed-Point

Name	Acc	Gyro	Time	Freq	d	BAs	PTs	Sampling	FP*
\mathcal{D}_1	✓	-	✓	-	17	6	0	\mathcal{W}_1	✓
\mathcal{D}_2	✓	✓	✓	✓	561	6	0	\mathcal{W}_1	-
\mathcal{D}_{2T}	✓		✓	-	272	6	0	\mathcal{W}_1	-
\mathcal{D}_3	✓	✓	✓	✓	561	6	6	\mathcal{W}_2	-
\mathcal{D}_{3T}	✓	✓	✓	-	272	6	6	\mathcal{W}_2	-

Classifier	Acronym	Accuracy
Decision Tree C4.5	DT	83.06%
Random Forest	RF	89.34%
k-Nearest Neighbors	k-NN	88.19%
Naive Bayes	NB	78.89%
Logistic Regression	LR	96.40%
Multilayer Perceptron	MLP	94.60%
Support Vector Machine	SVM	96.50%

Table 4.7: Performance comparison of the ML classifiers.

4.4 Results

In this section the generated HAR dataset is tested in order to verify its usability. This is done in two ways. First, the inertial data are validated using various state-of-the-art classification algorithms. Learned models from training data are used to predict new samples and evaluate the algorithms' performance with the HAR test data. Second, we present the results of a HAR competition that was organized in order to encourage external researchers to find novel ML solutions to the same recognition problem.

4.4.1 Dataset Validation

A series of experiments were conducted for data validation. We employed some of the most well-known ML algorithms as described in Section 2.5.2. The Waikato Environment for Knowledge Analysis (WEKA) ML software suite for dealing with data [Hall et al., 2009] was used for this validation task. For the experiments, we worked with the \mathcal{D}_2 data partition as it contains the all the BAs and the complete set available features from both inertial sensors. For each of the ML algorithms used we estimated its performance in terms of classification accuracy of the test data. The results are depicted in Table 4.7.

Classification results show evident differences between the performance of the selected ML algorithms. We found that some algorithms have a relatively low performance, with accuracies below 90%. These are DT, RF, k-NN and NB. On the other hand, we observed a better outcome in the remaining three ML algorithms: MLP, LR, and SVM. The last two have comparable classification accuracy with SVM slightly outperforming, with a 96.50%, and only differing by 0.1%. These results reinforce SVMs as a good candidate for performing HAR with smartphone

MC-GK-SVM — \mathcal{D}_2									
Activity	WK	WU	WD	SI	ST	LD		Sensitivity	Specificity
WK	486	6	4	0	0	0		97.98%	99.31%
WU	12	458	1	0	0	0		97.24%	98.59%
WD	5	27	388	0	0	0		92.38%	99.80%
SI	0	2	0	450	39	0		91.65%	99.71%
ST	0	0	0	7	525	0		98.68%	98.39%
LD	0	0	0	0	0	537		100.00%	100.00%
Accuracy	96.50%								

MC-GK-SVM — \mathcal{D}_3									
Activity	WK	WU	WD	SI	ST	LD	PT	Sensitivity	Specificity
WK	539	1	3	2	0	0	1	98.72%	98.93%
WU	28	513	2	1	1	0	14	91.77%	99.65%
WD	2	5	498	0	4	0	0	97.84%	99.84%
SI	0	3	0	486	68	0	0	87.25%	99.24%
ST	1	0	0	19	591	0	1	96.57%	97.65%
LD	0	0	0	0	0	604	0	100.00%	100.00%
PT	3	2	0	2	0	0	322	97.87%	99.53%
Accuracy	95.61%								

Table 4.8: Confusion matrices for the \mathcal{D}_2 and \mathcal{D}_3 dataset using the multiclass SVM with Gaussian kernel (MC-GK-SVM).

inertial data. This algorithm is the one selected in our research and from now on we focus our attention on its study and implementation.

In the particular case of SVMs, we used a multiclass SVM model through an OVA approach which take one binary SVM with Gaussian Kernel (GK-SVM) per activity. From now on, we will refer to this SVM configuration as MultiClass GK-SVM (MC-GK-SVM). This Radial Basis Function (RBF) kernel is commonly used in SVMs because it has shown to deal successfully with non-linear data and it is considered a universal approximator [Wang et al., 2004]. We employed the recognized LIBSVM library [Chang and Lin, 2011] which can be run under WEKA or Matlab.

The classification results for the \mathcal{D}_2 dataset are presented in Table 4.8 as a confusion matrix. It includes the classification accuracy of the algorithm along with the sensitivity and specificity measures for each class. They show an overall accuracy of 96.50% for the test data composed of 2947 patterns. In the same way, Table 4.8 shows the performance of \mathcal{D}_3 using the same classification algorithm (MC-GK-SVM). It achieves an accuracy of 95.61%. Notice that their difference relies on the addition of the PT class which combines all the available transitions into one. Moreover, the number of window samples of this class is smaller than the other classes. However, we take this into account during the training in order to balance the data through the C hyperparameter of the binary SVM. This dataset will be covered more in detail in Chapter 7 where we clarify how this additional class is needed for improving the online HAR system.

4.4.1.1 Dataset Publication

The HAR dataset has been made available for public use. It is composed of the experiment raw inertial data and also of the processed feature vectors for each window sample. A first version

(with data from \mathcal{D}_2) was submitted as the *Human Activity Recognition using Smartphones* dataset in the UCI Machine Learning Repository [Bache and Lichman, 2013]. The details of the dataset are shown in the following reference:

- [Reyes-Ortiz et al., 2013a] Jorge-Luis Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human activity recognition using smartphones data set. <http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>, 2013a

Moreover, a second version based on the \mathcal{D}_3 data is also public. It is available in the Smartlab Laboratory website as displayed in the following reference:

- [Reyes-Ortiz et al., 2014a] Jorge-Luis Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Recognition of basic activities and postural transitions using smartphones data set. <http://www.har.smartlab.ws>, 2014a

4.4.2 HAR Competition

A competition targeting the development of novel learning approaches for the classification of a set of activities was planned as part of a special session in Human and Motion Disorder Recognition at the European Symposium on Artificial Neural Networks (ESANN) in 2013. Competitors were challenged to submit their proposals given the HAR training data from \mathcal{D}_2 before its final publication. Participants were also provided with an unlabeled test set in order to receive from them the predicted labels for each sample. The performance of their approaches was measured in terms of classification accuracy using the experiment ground truth.

We received proposals from different universities and research centers in Europe. The three best contributions are depicted in Table 4.9. In [Romera-Paredes et al., 2013], a One-Vs-One (OVO) Multiclass SVM with linear kernel was proposed for the classification task. The method used majority voting to find the most likely activity for each test sample from an arrangement of 6 binary classifiers. An overall accuracy of 96.40% was reached on the test data and this method became the competition winner. For comparative purposes, they also evaluated the performance of a OVA SVM and a k-NN model which exhibited poorer accuracies (93.7% and 90.6% respectively). In the same way, a sparse kernelized matrix Learning Vector Quantization (LVQ) model was employed in [Kästner et al., 2013] for the HAR dataset classification achieving 96.23% test accuracy, only differing 0.17% against the first approach. Their method was a variant of LVQ in which a metric adaptation with only one prototype vector for each class was proposed. Ultimately, a novel confidence-based boosting algorithm (Conf-AdaBoost.M1.) was presented in [Reiss et al., 2013] and assessed against the traditional decision tree classifier and the AdaBoost.M1 algorithm. The method is a direct multiclass classification approach which exploits confidence information from weak learners for the classification. They achieved an accuracy of 94.33% on the test set.

Ref.	Approach Implemented	Accuracy
[Romera-Paredes et al., 2013]	OVO Multiclass linear SVM with majority voting.	96.40%
[Kästner et al., 2013]	Kernel variant of learning vector quantization with metric adaptation	96.23%
[Reiss et al., 2013]	Confidence-based boosting algorithm Conf-AdaBoost.M1.	94.33%

Table 4.9: HAR Competition. Test data classification accuracy of the best performing approaches.

The results of this competition also show evidence of the benefits of using SVMs for HAR as its winner also employed this algorithm. Their approach was slightly different than our OVA MC-GK-SVM algorithm but the performance was similar. In the following chapters, we present variations of the original SVM formulation in order to solve the recognition problem and adapt it to limited hardware such as the smartphone.

4.5 Summary

In this chapter, a new dataset for HAR using smartphones has been introduced. We thoroughly described the process to achieve this by incorporating the most important stages such as device selection, data collection experiments, dealing with data and validation. We also acknowledged initial classification results using 6 up-to-date ML algorithms including a multiclass SVM. This latter approach showed a noticeable advantage in terms of classification accuracy confirming our purpose of using it as the core ML algorithm in this thesis. This was also confirmed by the organized HAR competition whose winning algorithm was SVM-based.

These findings allows to argue that the use of smartphones for motion information retrieval seems feasible. In addition, it is also less obtrusive and invasive than other special purpose solutions (e.g. wearable sensors), and a practical way to walk for effectively performing HAR.

Making the data available to the public has brought many advantages. First, it offers to the research community the opportunity of comparing different HAR related works based on the same data, therefore, ML methods can be better evaluated. Second, it provides feedback from many users regarding different aspects of the dataset such as signal processing, feature selection, and possible corrections in future versions of the dataset.

Chapter 5

Hardware-Friendly Activity Recognition with Fixed-Point Arithmetic

5.1 Introduction

Exploiting SVM models for HAR on smartphones requires a multitude of operations to be carried out per second: despite not being an issue from a theoretical point of view, this could lead to battery discharge after few hours of continuous operation, making this approach unfeasible to allow people’s mobility. In this chapter, we explore a fixed-point arithmetic based reformulation of the conventional SVM, targeted towards multiclass classification. Up to date we have no knowledge of other research works that have incorporated fixed-point arithmetic into the learning algorithms for the classification of human activities. However, extensive research on fixed-point arithmetic has been developed to integrate ML models on hardware with limited resources (e.g. [Wawrzynek et al., 1993]). This idea was initially motivated some years ago because the assemble of devices with floating-point units was infeasible. Moreover, limited devices are usually preferred for specific-purpose applications if they demonstrate similar performance to traditional processing units as their production (and/or acquisition) costs are generally lower. Nowadays, it has become particularly interesting to retake these approaches and apply them in the development of software applications for portable devices such as smartphones which are highly demanding in terms of energy consumption and system resources management.

The term Hardware-Friendly SVM (HF-SVM) was first presented in [Anguita et al., 2007]. This method was designed for binary classification problems by employing fixed-point arithmetic in the FFP of the SVM classifier, with the purpose of allowing its use in hardware-limited devices. In this work, we adapt the model to target HAR on smartphones through a modified multiclass HF-SVM learning algorithm MultiClass HF-SVM (MC-HF-SVM). It aims to provide faster predictions and better preserve the battery lifetime of these portable devices with respect to conventional floating-point formulations while maintaining comparable system accuracy levels.



Figure 5.1: Activity Recognition process pipeline.

The introduction of this ML algorithm for HAR gives origin to the HF-HAR system which is described throughout this chapter.

The organization of this chapter is as follows: Section 5.2 describes the methodology of the hardware-friendly approach including its mathematical formulation from the binary to the multiclass case. This is succeeded by Section 5.3 which investigates under the light of SLT how the implementation of this new model brings advantages with respect to the generalization ability of the algorithm. Experiments show in Section 5.4 comparative results between this approach and the traditional SVM in terms of recognition performance, speed and battery consumption. Finally the chapter is fully summarized in Section 5.5.

5.2 The Hardware-Friendly Multiclass SVM

Here we introduce the MC-HF-SVM approach which consists of a reformulation of the SVM minimization problem in order to learn a model that allows the prediction of activity samples using only fixed-point notation. Although the learning process is itself performed with floating-point operations, its resulting model parameters and the FFP are not. Thereby, this approach makes possible a fully fixed-point implementation of prediction modules in hardware devices. MC-HF-SVM allows to vary the fixed-point number representation in terms of number of bits (k) to control over model accuracy and complexity, leading to noticeable improvements in terms of recognition speed and battery energy sparing without influencing recognition accuracy.

Figure 5.1 depicts schematically the basic HAR process including its main components: Data collection, signal processing, feature extraction and classification. In this section we focus only in the classification part, first by introducing the binary problem and then by generalizing the hardware-friendly approach to multiple classes.

5.2.1 The Binary Hardware-Friendly Formulation

The dual formulation of the original SVM (Equation (2.13)) is evidently invalid for its use in fixed-point arithmetic because the α_i values belong to the group of real numbers limited

between 0 and C . To overcome this issue, a normalization process can be employed. This process maintains the SVM accuracy unchanged because it does not affect the output sign of the classifier but only its magnitude. For this purpose, we propose the use of a new vector β which is defined as:

$$\beta_i = \alpha_i \frac{2^k - 1}{C}, \quad (5.1)$$

where k is the number of bits. Moreover, by omitting the equality constraint of the dual SVM formulation ($\mathbf{y}^T \boldsymbol{\alpha} = 0$) on Equation (2.13), we can search for an SVM solution without a bias term b . This is clearly advantageous when we deal with a fixed-point arithmetic formulation as this value can be difficult to control and instead the FFP can be easily computed as:

$$f(\mathbf{x}) = \sum_{i=1}^n y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) \quad (5.2)$$

This modification has no influence on the classification performance of the trained model as far as a Radial Basis Function (RBF) kernel, such as the Gaussian or the Laplacian ones, is exploited [Poggio et al., 2002]. These two modifications yield the following formulation:

$$\min_{\beta} \quad \frac{1}{2} \beta^T Q \beta - \mathbf{s}^T \beta \quad \text{s.t.} \quad 0 \leq \beta_i \leq \frac{2^k - 1}{C} \quad \forall i \in \{1, \dots, n\}, \quad (5.3)$$

where $s_i = (2^k - 1) / C \quad \forall i \in \{1, \dots, n\}$. Once the problem expressed in Equation (5.3) is solved, β can straightforwardly target fixed-point arithmetic through a simple nearest-integer normalization [Anguita et al., 2007].

To finally have a full FFP with only integer values, it is needed to modify the representation of the kernel $K(\cdot, \cdot)$ and the input vector \mathbf{x} in terms of number of bits (u and v bits respectively) [Anguita et al., 2007]. This produces:

$$0 \leq K(\mathbf{x}_i, \mathbf{x}) \leq 1 - 2^{-u} \quad \forall i \in \{1, \dots, n\}, \quad (5.4)$$

$$0 \leq x_j \leq 1 - 2^{-v} \quad \forall j \in \{1, \dots, d\}. \quad (5.5)$$

Consequently the modified Fixed-Point FFP formulation vector is:

$$f(\mathbf{x}) = \sum_{i=1}^n y_i \beta_i K(\mathbf{x}_i, \mathbf{x}) \quad (5.6)$$

In particular, we opted for a Laplacian kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = 2^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_1}$), instead of the more conventional Gaussian kernel, as it is more convenient for hardware limited devices [Anguita et al., 2007] because it can be easily computed using shifters. The Manhattan norm is defined as $\|\mathbf{x}\|_1 = \sum_{j=1}^d |x_j|$ and $\gamma > 0$ is the kernel hyperparameter which can be selected (altogether with

the regularization hyperparameter C), for example, through a k -Fold Cross Validation (KCV) procedure (with $k = 10$), where the hyperparameters space is explored according to a grid search procedure [Anguita et al., 2009].

5.2.2 Generalization to Multiclass SVM with probability estimates

In SVMs, the output of the FFP is either positive or negative. Its sign represents if the new sample is either classified as a given class or not. But the magnitude of its value, just as it is, does not represent a quantity directly comparable against other SVMs. Ideally, for a given sample in a multiclass problem, only one of the binary classifiers should be positive and the rest negative but this is not always the case. Therefore, output normalization methods are required to have comparable SVMs. We opted to compute probability estimates $p_c(\mathbf{x}) \in [0, 1]$ which represent how probable it is for a new sample pattern to be classified as a given class c . This is appropriate when used in OVA classification as each binary classifier is associated with a particular class (or, in this case, human activity). For a given number of classes m and a test sample \mathbf{x} , the probability output of each SVM ($p_c(\mathbf{x}) \forall c \in \{1, \dots, m\}$) is compared against the others to find the class c^* with the Maximum A Posteriori Probability (MAP). Assuming that all the classes have the same a priori distribution then: $c^* = \arg \max_c p_c(\mathbf{x})$.

The probability estimation is implemented using the approach presented in [Platt, 1999] in which the output of the SVM FFP ($f(\mathbf{x})$) obtained from the training set is fit to a sigmoid function of the following form:

$$p(\mathbf{x}) = \frac{1}{1 + e^{(\Gamma f(\mathbf{x}) + \Delta)}}, \quad (5.7)$$

where Γ and Δ are function parameters whose optimal values can be found using the $f(\mathbf{x}_i)$ values and the targets of the training samples (modified as $t_i = (y_i + 1)/2$) in the following error minimization function:

$$\arg \min_{\Gamma, \Delta} - \sum_{i=1}^n t_i \log(p(\mathbf{x}_i)) + (1 - t_i) \log(1 - p(\mathbf{x}_i)), \quad (5.8)$$

Considering the fixed-point arithmetic limitation, the sigmoid function, which works also with real numbers, cannot be directly used for estimating $p(\mathbf{x})$. This is solved by means of Look-Up-Tables (LUTs) which link $f(\mathbf{x})$ with $p(\mathbf{x})$ through a simple indexing operation. For this, a fixed number of bits must be defined in order to map the probability estimates $p(\mathbf{x})$ without the need of floating-point arithmetic.

The complete MC-HF-SVM process for the recognition of 6 activities is illustrated in Figure 5.2. It depicts the binary classifiers for each activity along with their associated LUT. The notation allows to understand how an input sample \mathbf{x} is processed until the most likely activity c^* is selected.

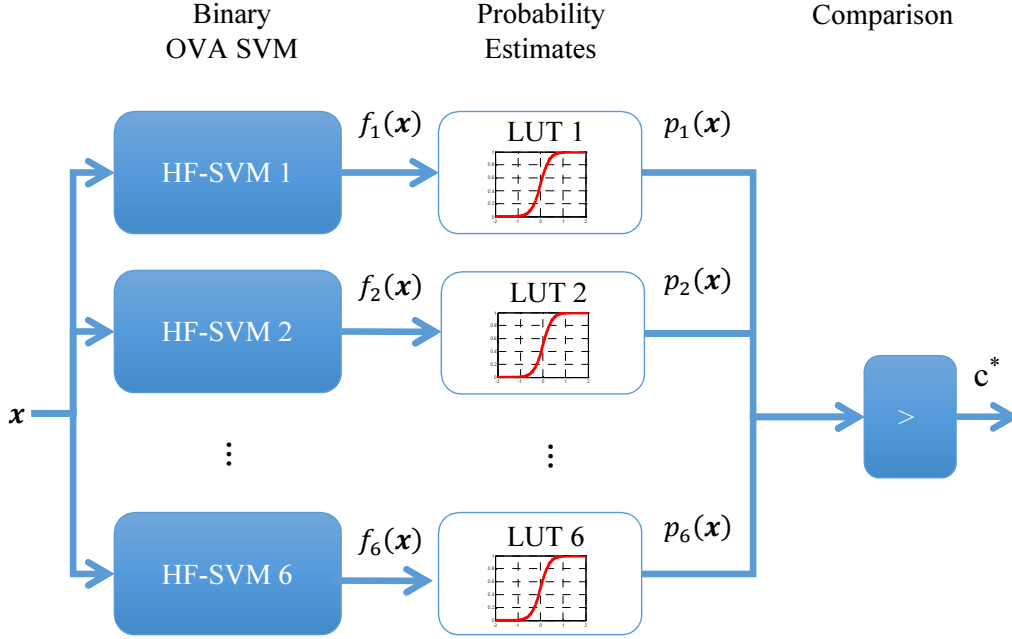


Figure 5.2: MC-HF-SVM using LUTs for probability estimates

5.3 HF-SVM and Statistical Learning Theory

In the last decades several works have been devoted to adapt ML approaches to specific hardware platforms [Epitropakis et al., 2010; Genov and Cauwenberghs, 2003; Irick et al., 2008; Lee et al., 2003] and, in particular, to analyze the effects of parameter quantization on the training and FFPs [Anguita et al., 2007; Lesser et al., 2011; Neven et al., 2009]. Motivations for these activities are usually linked to application-specific requirements but also to the basic principle of the SLT [Vapnik, 1995] where we have to search for the simplest model that correctly classifies the available data. The introduction of bit-based hypothesis spaces brings widespread benefits on the learning process of classifiers (i.e. classes of functions where models are described through a limited number of bits). This is due to the fact that reducing the number of bits largely influences the complexity of the hypothesis space [Anguita et al., 2013a], which is a key issue in Machine Learning as underlined in [Bartlett et al., 2005; Shawe-Taylor et al., 1998]. If we are able to reduce the complexity of the hypothesis space without affecting the ability of the algorithm to learn a function with low empirical error, in practice, we are able to learn more effectively [Herbrich and Williamson, 2003; Shawe-Taylor et al., 1998].

In this section we investigate how the adoption of a fixed-point arithmetic affects the generalization ability of a classifier in the form of Equation (5.6). In order to do this we describe

each parameter β_i as an integer value of k bits:

$$\beta_i = \sum_{j=0}^{k-1} b_i^j 2^j, \quad (5.9)$$

where $b_i^j \in \{0, 1\}$ is a binary valued variable and therefore β_i can be expressed as an integer variable such that $0 \leq \beta_i \leq 2^k - 1$. Since each b_i^j belongs to a finite set, for a fixed training set of cardinality n and a fixed kernel (with its hyperparameter), the number of classifiers that we can represent is finite. According to the notation of [Vapnik, 1995] we call N_f^n the number of classifiers that we can build with b_i^j , $i \in \{1, \dots, n\}$ and $j \in \{0, \dots, k-1\}$. Consequently we can exploit the well-known Vapnik's generalization bounds for finite hypothesis sets [Vapnik, 1995] which uses N_f^n as measure of complexity. Let then d^β be the number of nonzero parameters ($\beta_i \neq 0$) then:

$$N_f^n(k, d^\beta) \leq \sum_{i=1}^{d^\beta} \binom{n}{i} \left[\left(2^k - 1\right)^{d^\beta} - \left(2^{k-1} - 1\right)^{d^\beta} \right], \quad (5.10)$$

where we take into account the fact that if all the parameters are even numbers, they can be divided by two without changing the class estimate. If, instead, d^b is the number of nonzero parameters, $b_i^j \neq 0$, then

$$N_f^n(k, d^b) \leq \sum_{i=1}^{d^b} \binom{n}{i} \binom{k}{i}. \quad (5.11)$$

In the SLT and Structural Risk Minimization (SRM) frameworks [Vapnik, 1995], a good generalization capability on previously unseen data can be guaranteed [Anguita et al., 2012a; Vapnik, 1995] if a nested structure of the available hypothesis sets with increasing complexity is defined ($\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$). In this way, the generalization capability of a model can be controlled by choosing the set that achieves the best compromise between complexity and learning error.

In our case the complexity of the class can be defined through two quantities, k and d^β (or d^b). Starting from the set \mathcal{H}_1 with complexity $N_f^l(1, 1)$ we can increase the complexity by increasing the number of bits $k \rightarrow k + 1$ or by decreasing the sparsity of the representation $d^\beta, d^b \rightarrow d^\beta, d^b + 1$. In other words we have to search the best class which is as sparse as possible (smaller d^β or d^b) and represented with the minimum number of bits k .

Obviously a classifier that belongs to a space with smaller complexity is also more energy efficient respect to the one that belongs to a space with higher complexity, as will also be shown in the subsequent experiments.

Increasing the complexity of the space has also direct consequence on the generalization ability of the classifier since according to the bound of Vapnik [Vapnik, 1995], which holds with probability $(1 - \delta)$:

$$\pi \leq \nu + \sqrt{\frac{\ln \left[N_f^n(k, d) \right] - \ln(\delta)}{2l}} \quad (5.12)$$

where π is the generalization error and ν is the error obtained by the learning machine on the

Feature Vector	
Measure	Applied to
<i>SMA</i>	$\mathbf{a}_\tau(t), \mathbf{a}'_\tau(t), \mathbf{a}(t), \mathbf{a}'(t)$
<i>Mean</i>	$a_{\tau,1}(t), a_{\tau,2}(t), a_{\tau,3}(t)$
<i>STD</i>	$a_{\tau,1}(t), a_{\tau,2}(t), a_{\tau,3}(t)$
<i>Correlation</i>	$a_{\tau,1}(t)-a_{\tau,2}(t), a_{\tau,1}(t)-a_{\tau,3}(t), a_{\tau,2}(t)-a_{\tau,3}(t)$
<i>Entropy</i>	$A_{\tau,1}(\omega), A_{\tau,2}(\omega), A_{\tau,3}(\omega), A_{mag}(\omega)^*$

* Capitalized letters represent the signal in the frequency domain ω .

Table 5.1: List of measures for computing feature vectors.

dataset.

This result is similar to the one presented in [Neven et al., 2008]. The important outcome of this section is that the number of bits in the HF-SVM has a strong regularization effect with an impact on the generalization ability of the classifier. Between two classifiers with approximately the same performance, we have to choose the one that can be represented with less number of bits since it is more energy efficient and it has more capacity of performing well on previously unseen data.

Finally we want to highlight that the bound in Equation (5.12) is very loose since it is data independent and does not take into account the quality of the available samples for its estimation. In the last few years, proposed data dependent bounds [Bartlett and Mendelson, 2003; Bartlett et al., 2005] are becoming tighter and providing better interpretation of the generalization ability of classifiers. They have shown to work well on the performance estimation of real world problems such as in [Anguita et al., 2012a]. For these reason, the understanding of the influence of fixed-point arithmetic approaches in the estimation of these bounds is an interesting topic of research.

5.4 Results

The performance of the MC-HF-SVM was evaluated through a collection of experiments using the HAR dataset \mathcal{D}_1 described in Section 4.3.3. The data samples were composed of 17 features extracted from the smartphone triaxial accelerometer in the time and frequency domain. These features had been previously suggested in the HAR literature [Bao and Intille, 2004; Lovell et al., 2007; Sama et al., 2010] including measures such as SMA, mean, Standard Deviation (STD), entropy and signal-pair correlation, when applied to the available processed inertial signals, provided the set of features depicted in Table 5.1.

The method evaluation is divided in two parts. The first one analyzes the system recognition performance against a traditional approach (Multiclass LK-SVM (MC-LK-SVM)). Then, in the second part, we focus on evaluating other algorithm attributes more related with its use in hardware devices. These are the recognition speed and battery discharge.

5.4.1 System performance using Fixed-Point Arithmetic

\mathcal{D}_1 has been discretized in order to allow its use with the MC-HF-SVM algorithm. The data was converted from floating-point to fixed-point and represented with different number of bits k . Ten MC-HF-SVM models were learned, one for each value of k which ranged from 4 to 16 bits. Their performance was assessed in terms of test data error (Equation (2.21)) and compared against the standard conventional floating-point MC-LK-SVM. The classification results for each model are showed in Figure 5.3.

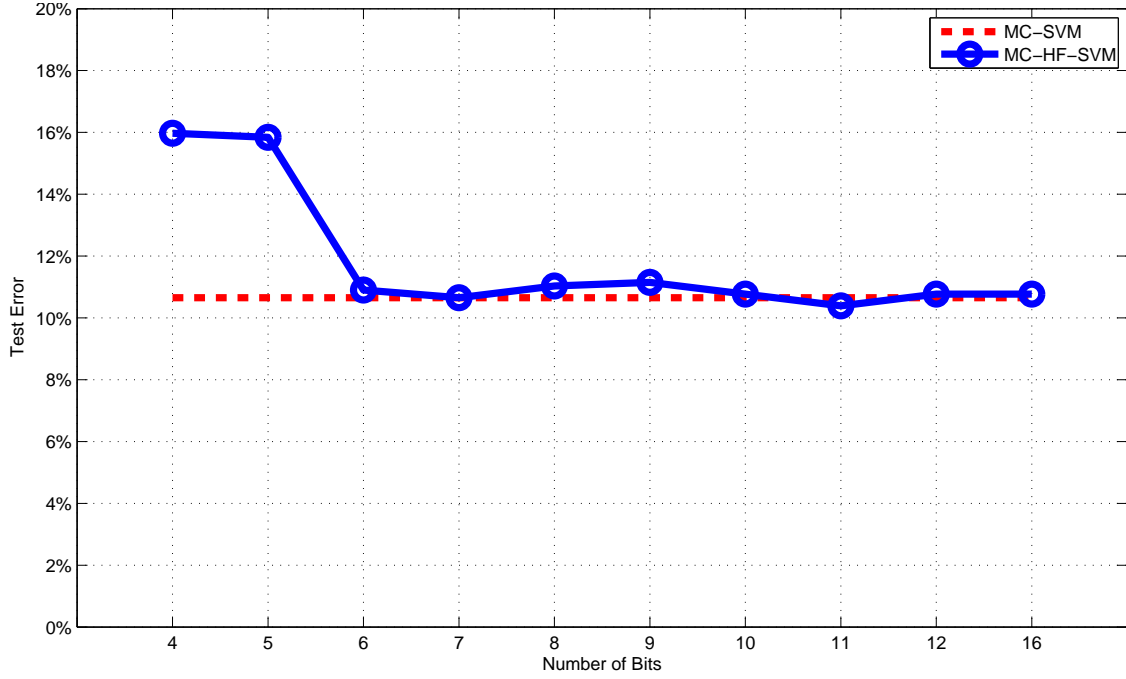


Figure 5.3: Comparison between error rates obtained with MC-LK-SVM (red dotted line) and MC-HF-SVM (blue line) as k is varied.

The test error curve shows a plateau which values appear to be stable (near 1% variation) for k ranging from 6 to 16 bits and are equivalent to the error obtained with the floating-point MC-LK-SVM (represented with the dotted red line). In addition, the experiment shows that for this HAR dataset, $k = 6$ bits are sufficient for obtaining a recognition performance similar to the MC-LK-SVM approach. Once the numbers of bits drops below this value, the test error significantly increases by around 50%. From a practical point of view, however, it is worth underlining that fixed-point libraries allow to use only values of k which are powers of 2, as will be also detailed in the forthcoming Section 5.4.2 where we will thus consider $k = 8$ as our minimum reference value.

Moreover, it is also noticeable from the graph that some of the error values with fixed-point representation were smaller than the one found with the MC-LK-SVM approach. This finding coincides with what is observed elsewhere in the literature (e.g. [Anguita and Sterpi, 2006;

MC-LK-SVM								
Activity	WK	WU	WD	SI	ST	LD	Sensitivity	Specificity
WK	109	0	5	0	0	0	95.61%	97.63%
WU	1	95	40	0	0	0	69.85%	97.86%
WD	15	9	119	0	0	0	83.22%	93.03%
SI	0	5	0	132	5	0	92.96%	99.38%
ST	0	0	0	4	108	0	100.00%	99.26%
LD	0	0	0	0	0	142	96.43%	100.00%
Accuracy	89.35%							

Table 5.2: Confusion Matrix of the classification results on the test data using the traditional floating-point MC-LK-SVM.

MC-HF-SVM $k = 8$								
Activity	WK	WU	WD	SI	ST	LD	Sensitivity	Specificity
WK	109	2	3	0	0	0	95.61%	97.63%
WU	1	98	37	0	0	0	72.06%	96.63%
WD	15	14	114	0	0	0	79.72%	93.81%
SI	0	5	0	131	6	0	92.25%	99.54%
ST	0	1	0	3	108	0	96.43%	99.11%
LD	0	0	0	0	0	142	100.00%	100.00%
Accuracy	88.97%							

Table 5.3: Confusion Matrix of the classification results on the test data using the MC-HF-SVM with $k = 8$ bits.

Neven et al., 2008]). It is worth noting that we can remarkably reduce the number of bits (from ∞ to 8 bits) without losing the possibility of representing the functions that are characterized by good performance on the training set (as underlined in [Anguita et al., 2011, 2013a; Koltchinskii, 2006]): these functions will be most likely chosen by the learning process and, then, there seem to be no reasons to search for more complex spaces. Moreover, note that few bits are required in order to represent these functions, thus contemplating an infinite-dimension space appears to be unmotivated by practical needs [Anguita et al., 2013a]. In the SRM framework we have to search for the simplest hypothesis space (before looking at the training set [Vapnik, 1995]) that guarantees the best trade off between accuracy on the training set and complexity of the space. The introduction of a bit-based hypothesis space is also encouraged by the basic ML idea to search for the simplest class of functions capable of solving the problem under examination.

In Table 5.2 and Table 5.3, the confusion matrices of the MC-LK-SVM and the MC-HF-SVM with $k = 8$ bits for the test data are depicted. In them, measures of overall accuracy, sensitivity and specificity are also given and exhibit very similar values in both approaches. Small variations are noticed in the recognition accuracy of dynamic activities within the two SVM approaches such as in the *walking downstairs* and *walking upstairs* activities, which also display some misclassifications mainly to their movement similarities. Static activities on the other hand performed better, such as *laying*, for which we reported 0% classification error. Furthermore, a small misclassification overlap was found between *standing* and *sitting*, which is

attributed to the waist-mounted smartphone physical location and the difficulty to discriminate between them: this is mainly due to the slight inclination difference of the phone with respect to the vertical axis when these activities are performed and largely depends on the user's body type.

These classification errors could be in some way improved by incorporating new types of features or sensors into the HAR system. For example, the inclusion of gyroscopes as we will see in the following Chapter 6 or the incorporation of additional accelerometers in different body parts.

5.4.2 Processing Time and Battery Consumption

Various tests were performed on the smartphone to determine the advantages of using this novel hardware-friendly approach in terms of recognition speed and battery consumption. We expected that avoiding the use of the floating-point arithmetic for complex calculations could lead to energy sparing on stand-alone devices. For these trials, we used a SGSII smartphone equipped with a Li-Ion 1650 mAh battery with up to 610 hours of stand-by operation and Android OS as the operating system (Gingerbread version 2.3.4).

A phone app was developed for this purpose: measure prediction speed and battery consumption. All the expensive operations were written in C using the Android NDK (signal processing and Machine Learning algorithm). Before trials, we turned off all phone services (e.g. Wi-Fi, Bluetooth and 3G Network) and, more importantly, the phone screen. This latter is in general the most energy consuming part in a smartphone. We aimed to isolate this measurement process as much as possible to obtain a realistic approximation of the variables under evaluation.

A simulation of the HAR process was implemented on the smartphone with the possibility of adjusting the number representation. This was achieved for either fixed-point or floating-point arithmetic using the default data types available in the C language from 8 to 64 bits. They were selected because the available libraries have only power of two number representations.

On the first part of the measurements, we decided to continuously make activity predictions over a fixed period of time (5 min) and obtain an estimate of the average prediction rate (in number of predictions/sec). The time of the activity recognition process was measured starting from the sensor reading to determining the SVM FFP output was measured for each approach. Table 5.4 shows the obtained results. It is worth highlighting the large difference between the rates using the fixed-point representation instead of the floating-point and also the proportional relationship between the number of bits used and the processing time. For instance, the 32-bits integer model outperforms in speed the 32-bit float model by almost 7 times.

An additional test was carried out aimed to measure battery consumption with the floating-point and fixed-point representations. For this, the accelerometer sensor was set to constantly read the triaxial signal at a fixed frequency as described in Section 4.3.2 and then copied into a circular buffer. Every 1.28 sec an interruption started the activity recognition process using the last available window sample taking into account the 50% overlap between windows and

Data Type	No. Bits	No.Predictions/sec
<i>Fixed-Point Representation</i>		
char	8	315.35
short int	16	241.54
int	32	185.00
long int	64	141.70
<i>Floating-Point Representation</i>		
float	32	27.04
double	64	20.68

Table 5.4: Estimated prediction rates on the smartphone with basic data types.

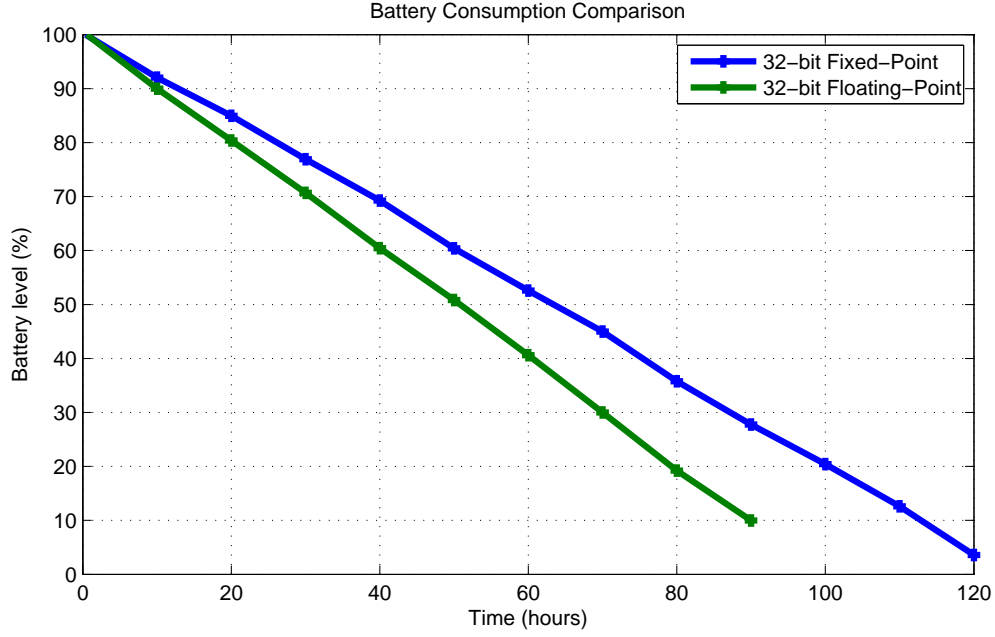


Figure 5.4: Comparison 32-bit floating-point MC-LK-SVM and 32-bit fixed-point MC-HF-SVM with respect to battery discharge

their 2.56 sec length. In this experiment we compared the data types with the same number of bits but with different arithmetic: 32-bit float and 32-bit integer. For this, we run three times for each data type the HAR smartphone application continuously and measured the time of battery discharge from a fully charged state until a minimum level of 10% was reached. We found that the average battery time using the 32-bit float model was of 89 hours and the time with the 32-bit integer was of 112 hours and they both described a linear discharge trend as it is visualized in Figure 5.4. Their time difference is equivalent to an increase of 25% of the battery life when the application is running alone. These measurements are dependent on the hardware and OS used but they are showing a trend on the improvements that can be reached with this hardware-friendly approach. For obtaining a more reliable measure of the relationship between the battery savings and processing time more experimental tests with different devices and operational conditions would be required.

In current scenarios, even small savings in battery consumption make a big difference in deciding whether or not to use a mobile app, such as in cases where the HAR application is required to deliver activity information to other higher-level decision applications (e.g. phone apps for maintaining a healthy lifestyle through HAR [Lane et al., 2012]), thus implying sharing system resources. In general, we aim to build a device able to operate at least during a full day in order to be able to recharge battery during idle times such as at nights. These results are a good indicator of the benefits that this method can offer for saving battery life and the possibility of being integrated into devices for everyday life.

5.5 Summary

In this section we presented a novel energy efficient system for the classification of BAs using smartphones (HF-HAR). It has been constructed based on a modified SVM model that works with fixed-point arithmetic (MC-HF-SVM). This approach brings a significant reduction of the processing time in the prediction of activities, attributable to the change of arithmetic, leading to a lower use of system resources. Moreover, results showed the method provides energy savings while maintaining comparable recognition performance when compared with the traditional SVM approach (MC-LK-SVM).

The proposed model was supported in terms of Structural Risk Minimization principles, where simpler models are preferred if they have equivalent ability to learn when compared to more complex approaches. This work is relevant to AmI and AAL applications where energy consumption becomes a critical issue such as in long term smartphone-based activity monitoring systems. Similarly, it could be explored the possibility of using this approach in low-cost devices (e.g. with fixed-point hardware) for applications including body sensor networks with local prediction of events and disposable wearable sensing.

The experimental results confirmed that it is possible to substitute the standard Multiclass SVM model with more efficient fixed-point representations. Further experimentation is required to evaluate the system in more realistic conditions when the smartphone system shared resources are allocated for different applications, and also using different smartphones for its evaluation.

Part III

Online Activity Recognition with Smartphones

Chapter 6

Linear SVM Models for Online Activity Recognition

6.1 Introduction

The exploitation of smartphones for HAR has been an active research area in which the development of fast and efficient ML approaches is crucial for providing real-time performance and preserving the device’s battery life. In this chapter, we focus on the exploration of three linear SVM algorithms for performing online HAR. Their formulation differs only in the regularization term. It uses different norms, namely L1, L2 and L1-L2, and gives the SVM different properties and behavior. Specifically, these differences affect the trade-off between dimensionality reduction and classification accuracy. Using the proposed learned linear models we implement our first online HAR system: Linear HAR System (L-HAR). It provides in real-time the classification of activities using a smartphone device using features extracted from its embedded accelerometer and gyroscope.

Moreover, we present a novel algorithm for training L1-L2-Norm SVM (L1-L2-SVM) classifiers. The proposed training approach allows the exploitation of all the well-known effective and reliable tools (e.g. Quadratic Programming (QP) solvers) already developed for solving the conventional L2-Norm SVM (L2-SVM), thus minimal effort is required by the user to implement L1-L2 model training. The proposed method is flexible, as it also allows to train L1-Norm SVM (L1-SVM), L2-SVM. The effectiveness of this approach is tested on our HAR dataset.

We also show the benefits of adding smartphones gyroscope signals into the recognition system against the common approach which only uses accelerometer data. MEMS gyroscopes made its entrance in the smartphones market a couple of years after accelerometers and they have not been fully explored [Wu et al., 2012]. We also study two feature selection mechanisms for allowing a faster recognition: the utilization of exclusively time domain features from the inertial data and the adaptation of the L1-L2-SVM which controls over the number of non-informative features as part of its model construction process.

This chapter is distributed in the following way: first we present the standard algorithms

based on L1 and L2 norms (Section 6.2) and show how they are adjusted in order to solve our multiclass HAR problem. In a similar way, we present in Section 6.3 the combined algorithm (MultiClass L1-L2-SVM (MC-L1-L2-SVM)) that allows to merge the effectiveness of L2 models and the feature selection characteristics of L1 solutions for HAR. Moreover, we describe our proposed training algorithm (Extended SMO (EX-SMO)). Experimental results regarding the proposed SVM approaches, addition of gyroscopes and feature selection mechanisms are presented in Section 6.4. Finally we summarize the chapter in Section 6.5.

6.2 L1-Norm and L2-Norm SVMs for Activity Recognition

Our target is to design a model, which can be effectively run on smartphones with limited battery life and computational restrictions. We have thus to identify the simplest possible classifier exploiting the smallest set of features that guarantees the best performance/computational burden ratio. For these purposes, we peruse the exploitation of linear models, which use only those selected inputs that are crucial to attain sufficient classification accuracy. In this section we formulate the OVA SVMs with the L1- and L2-Norms.

In the framework of supervised learning and in the case of binary classification problems, the goal is to approximate the relationship between examples from a set \mathcal{X} composed of $\mathbf{x}_i \in \mathbb{R}^d$ elements and a set \mathcal{Y} which contains outputs targets $y_i = \pm 1$. This relationship is encapsulated by a fixed, but unknown, probability distribution \mathcal{P} . A training set $D_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ is sampled according to \mathcal{P} . The learning algorithm maps D_n to $f \in \mathcal{F}$ with a linear separator in the original space $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$. Moreover, the accuracy in representing the hidden relationship \mathcal{P} is measured with reference to a loss function $\ell(f(\mathbf{x}), y)$.

In general, the hard loss function $\ell_H(f(\mathbf{x}), y) = [1 - y \text{sign}(f(\mathbf{x}))]/2$ seems the most natural choice, as it counts the number of misclassifications, but unfortunately it is non-convex. For this reason the hinge loss function $\ell_\xi(f(\mathbf{x}), y) = [1 - y f(\mathbf{x})]_+$ is exploited instead [Vapnik, 1998]. It is possible to introduce a regularization term in order to adjust the size of the class. In this case, we choose the Euclidean norm ($\|\mathbf{w}\|_2 = \sqrt{\sum_{j=1}^d w_j^2}$), also known as the L2-Norm [Tikhonov and Arsenin, 1978]. According to the SRM principle [Vapnik, 1998], we can derive, similarly to Equation (2.2), the primal formulation using the L2-Norm in the minimization problem:

$$\min_{\mathbf{w}, b, \xi} \quad \frac{1}{2} \|\mathbf{w}\|_2^2 + C \mathbf{1}_n^T \boldsymbol{\xi}, \quad \text{s.t. } Y(X\mathbf{w} + \mathbf{b}_n) \geq \mathbf{1}_n - \boldsymbol{\xi}, \quad \boldsymbol{\xi} \geq \mathbf{0}_n, \quad (6.1)$$

where $\xi_i = \ell_\xi(f(\mathbf{x}_i), y_i)$, $X = [\mathbf{x}_1 | \dots | \mathbf{x}_n]^T$, $\mathbf{y} = [y_1 | \dots | y_n]^T$, $Y = \text{diag}(\mathbf{y})$ (Y is a diagonal matrix where the element on the diagonal are the $y_{i \in \{1, \dots, n\}}$). Also by introducing n Lagrange multipliers $\boldsymbol{\alpha}$ we can obtain the dual formulation (Equation (2.13)) which is a CCQP problem and can be solved through many efficient techniques previously been proposed in the literature [Shawe-Taylor and Sun, 2011]. For example, the well-known and widely used SMO [Keerthi et al., 2001; Platt, 1998] which consists in iteratively updating the two $\alpha_{i \in \{1, \dots, n\}}$ that mostly

violate the KKT conditions until convergence is reached.

The L2-SVM method described above does not perform any dimensionality reduction. This is, however, not desirable in some practical applications where it is needed to highlight relevant features, discard noisy ones, and reduce the computational burden of performing the classification of new samples. For this issue, the replacement of the L2-Norm term with the Manhattan distance or L1-Norm ($\|\mathbf{w}\|_1 = \sum_{j=1}^d |w_j|$) has been proposed [Tibshirani, 1996]:

$$\min_{\mathbf{w}, b, \xi} \quad \|\mathbf{w}\|_1 + C \mathbf{1}_n^T \xi, \quad \text{s.t. } Y(X\mathbf{w} + \mathbf{b}_n) \geq \mathbf{1}_n - \xi, \quad \xi \geq \mathbf{0}_n, \quad (6.2)$$

This L1 regularization introduces an automatic feature selection effect which is embedded in the learning process. The optimization problem using L1-Norm is formulated in the following way:

$$\begin{aligned} \min_{\mathbf{w}^+, \mathbf{w}^-, b, \xi} \quad & \mathbf{1}_d^T (\mathbf{w}^+ + \mathbf{w}^-) + C \mathbf{1}_n^T \xi \\ \text{s.t.} \quad & Y[X(\mathbf{w}^+ - \mathbf{w}^-) + \mathbf{b}_n] \geq \mathbf{1}_n - \xi, \quad \xi \geq \mathbf{0}_n, \quad \mathbf{w}^+, \mathbf{w}^- \geq \mathbf{0}_d, \end{aligned} \quad (6.3)$$

The solution to this problem provokes many of the weights (w_j) to become zero which is useful for our needs as the calculations required to perform the FFP are directly related to the number of weights different from zero. The disadvantage of this approach is that effective tools developed for the conventional L2-SVM cannot be applied for solving it.

Note also that the conventional kernel trick cannot be exploited in the previous formulation, thus the effectiveness of linear models assumes an ever greater importance. This problem is a standard Linear Programming problem, for which many tools have been developed throughout the years [Press et al., 2007].

L1-SVM allows to perform dimensionality reduction thanks to the characteristics of the Manhattan norm exploited, that is several weights w_j will be (generally) nullified during the learning procedure: this is in contrast with the conventional L2-SVM, where $w_j \neq 0 \forall j = 1, \dots, d$ in the final model.

The extension of the L1-SVM binary problem into multiple classes for our application can be achieved using a OVA approach (MultiClass L1-SVM (MC-L1-SVM)) as we also did in Chapter 5 but with some modifications in order to take advantage of the intrinsic feature selection mechanisms of the model. The final classification is carried out by contemplating the output of m linear models:

$$f_c(\mathbf{x}) = \mathbf{w}_c^T \mathbf{x} + b_c, \quad \forall c \in \{1, \dots, m\} \quad (6.4)$$

Each binary model is learned independently and not all the zero-valued weights for one class, are also zero for the others. For such purposes, we can define the set \mathcal{S} which includes all the

indexes of the features whose weights are equal to zero for all the classes in the following way: $\mathcal{S} = \{j \mid w_{c,j} = 0, \forall c \in \{1, \dots, m\}\}$. In that sense, all the zero-valued weights can be removed from the FFP computation. We can also compute the effective feature dimensionality reduction as the fraction of selected features $\rho = \frac{d-|\mathcal{S}|}{d}$, where $|\mathcal{S}|$ indicates the cardinality of the set. Additionally, we can measure the feature reduction which is possible per class. This measure give us an idea of how fast the FFP computation can be. So we also define the mean feature dimensionality reduction as $\bar{\rho} = \frac{1}{m} \sum_{c=1}^m \frac{d-|\mathcal{S}_c|}{d}$, where \mathcal{S}_c are the indexes of zero-valued features per class.

6.3 L1-L2 SVM Algorithm

The conventional L2-SVM approach is considered as one of the state-of-the-art methods for classification, and several effective techniques have been developed throughout the years for training these models [Fan et al., 2008; Ghio et al., 2012; Keerthi et al., 2001; Platt, 1998; Shalev-Shwartz et al., 2007]. While allowing to derive sparse classifiers (i.e. models described by exploiting a limited subset of training patterns), L2-SVM [Vapnik, 1998] does not perform any feature reduction which becomes a limitation for the analysis of the dataset and the interpretability of the informative content of the inputs. On the other hand, L1-SVM allows to introduce in the learning process an automatic dimensionality reduction effect. However, despite this being very appealing for this task, L1-SVM is also characterized by some drawbacks:

1. No feature grouping effect characterizes L1 models, i.e. clusters of highly cross-correlated inputs are usually not entirely selected by the training procedure [Segal et al., 2003];
2. When the dimensionality of the dataset is remarkably larger than the number of samples, L1 models are able to exploit only a number of inputs at most equal to the cardinality of the training set, which could be restrictive in some applications [Zou and Hastie, 2005]
3. L1-SVM require custom ad-hoc algorithms to be developed for classifier training [Friedman et al., 2010], which do not exploit the huge effort spent in the last decades for designing effective solvers for the conventional SVM (e.g. [Keerthi et al., 2001; Platt, 1998]).

In order to deal with the first two points above, an SVM which combines L1- and L2-Norms has been proposed in [Zou and Hastie, 2005]. It allows to enhance feature grouping effects in model training, to properly balance sparsity and dimensionality reduction, and to combine the effectiveness of the L2 approach and the feature selection characteristics of L1-SVMs.

Moreover, to cope with the third issue, we present a new training tool allowing to efficiently deal with SVMs based on L1-, L2- and L1-L2-Norms. The proposal builds on the efficient solvers developed in the last decades for L2-SVM (e.g. [Keerthi et al., 2001; Platt, 1998]), and thus can be implemented with a minimal effort.

6.3.1 L1-L2 SVM Formulation

L1-L2-SVM has been proposed with the intention of combining the effectiveness of L2 solutions and the dimensionality reduction capabilities of L1 models [Zou and Hastie, 2005]:

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \lambda \|\mathbf{w}\|_2^2 + (1 - \lambda) \|\mathbf{w}\|_1 + C \mathbf{1}_n^T \boldsymbol{\xi} \\ \text{s.t.} \quad & Y(X\mathbf{w} + \mathbf{b}_n) \geq \mathbf{1}_n - \boldsymbol{\xi}, \quad \boldsymbol{\xi} \geq \mathbf{0}_n, \end{aligned} \quad (6.5)$$

where $\lambda \in (0, 1]$ is a constant that balances sparsity characteristics with feature selection ability between the two margin terms. It is worth noting that, if $\lambda \rightarrow 0$, we can derive the L1-SVM, while $\lambda = 1$ leads to the conventional L2-SVM. To solve this, we can introduce an identity matrix I_d of size $d \times d$ and reformulate Equation (6.5) in the following manner:

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\eta}^+, \boldsymbol{\eta}^-} \quad & \frac{1}{2} \lambda \mathbf{w}^T \mathbf{w} + (1 - \lambda) \mathbf{1}_d^T (\boldsymbol{\eta}^+ + \boldsymbol{\eta}^-) + C \mathbf{1}_n^T \boldsymbol{\xi} \\ \text{s.t.} \quad & Y(X\mathbf{w} + \mathbf{b}_n) \geq \mathbf{1}_n - \boldsymbol{\xi}, \quad \boldsymbol{\xi} \geq \mathbf{0}_n \\ & I_d \mathbf{w} = \boldsymbol{\eta}^+ - \boldsymbol{\eta}^-, \quad \boldsymbol{\eta}^+, \boldsymbol{\eta}^- \geq \mathbf{0}_d. \end{aligned} \quad (6.6)$$

Therefore, we can compute the dual formulation of Equation (6.6) using the method of Lagrange multipliers. This leads us to the following:

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \quad & \frac{1}{2} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}^T \begin{bmatrix} YX X^T Y & YX \\ X^T Y & I_d \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} - \begin{bmatrix} \sqrt{\lambda} \mathbf{1}_n \\ \mathbf{0}_d \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \\ \text{s.t.} \quad & \begin{bmatrix} \sqrt{\lambda} \mathbf{y} \\ \mathbf{0}_d \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathbf{0}_n \\ -\frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \leq \begin{bmatrix} \frac{C}{\sqrt{\lambda}} \mathbf{1}_n \\ \frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d \end{bmatrix}, \end{aligned} \quad (6.7)$$

where $\mathbf{w} = \frac{\sqrt{\lambda}}{\lambda} (X^T Y \boldsymbol{\alpha} + \boldsymbol{\beta})$, and $\boldsymbol{\alpha} \in \mathbb{R}^n$, $\boldsymbol{\beta} \in \mathbb{R}^d$ are the Lagrange multipliers of the inequality and equality constraints.

The problem depicted in Equation (6.5) is convex, the same as its dual formulation [Boyd and Vandenberghe, 2004]. For its solution, we present in the following section a novel approach to solve it.

6.3.2 Extended Algorithm for solving L1-L2 SVMs

Equation (6.7) can be split into two subproblems which can be solved iteratively until the solution is reached [Boyd and Vandenberghe, 2004]. In particular, we can fix $\boldsymbol{\beta}$ to some constant value $\hat{\boldsymbol{\beta}}$, which satisfy the constraints, and then reformulate Equation (6.7) as follows:

$$\begin{aligned} \alpha_t^*, b_t : \quad & \arg \min_{\alpha} \quad P1(\alpha, \hat{\beta}) = \frac{1}{2} \alpha^T Y X X^T Y \alpha + \left(X^T Y \hat{\beta} - \mathbf{1}_n^T \right) \alpha \\ \text{s.t.} \quad & \sqrt{\lambda} \mathbf{y}^T \alpha = 0, \quad \mathbf{0}_n \leq \alpha \leq \frac{C}{\sqrt{\lambda}} \mathbf{1}_n, \end{aligned} \quad (6.8)$$

where the bias b_t is derived analogously to the conventional **L2-SVM** approach. As a matter of fact, Equation (6.8) is a simple reformulation of the conventional dual formulation of **L2-SVM**, which can be solved by exploiting any of the several approaches proposed in the last decades [Shawe-Taylor and Sun, 2011]. If, instead, we fix α to some constant value $\hat{\alpha}$, which also satisfies the constraints, we can reformulate Equation (6.7) in the following manner:

$$\begin{aligned} \beta_t^* : \quad & \arg \min_{\beta} \quad P2(\beta, \hat{\alpha}) = \frac{1}{2} \beta^T I \beta + \left(\hat{\alpha}^T Y X \right) \beta \\ \text{s.t.} \quad & -\frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d \leq \beta \leq \frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d. \end{aligned} \quad (6.9)$$

Equation (6.9) has a closed form solution, as we have to identify the minimum of a paraboloid in a box, which is characterized by an identity Hessian matrix:

$$\beta_t^* = \max \left[-\frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d, \min \left[\frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d, \left(\hat{\alpha}^T Y X \right)^T \right] \right]. \quad (6.10)$$

Consequently, we propose an approach to solve Equation (6.7) which is detailed in Algorithm 1. For this, solutions of Equation (6.8) and Equation (6.9) are iteratively found. The former case can therefore be solved with any of the methods surveyed by [Shawe-Taylor and Sun, 2011].

Particularly, we have also described Algorithm 2 which is focused on the adoption of the SMO algorithm [Keerthi et al., 2001; Platt, 1998] for solving Equation (6.8). Note that, since, at every optimization step, two α_i coefficients are optimized by the SMO, in order to better balance the overall L1-L2 optimization procedure we can run $\frac{n}{2}$ iterations of SMO and, then, update $\beta_{i \in \{1, \dots, d\}}$.

By simply tuning the value of λ with the EX-SMO algorithm, it is also possible to exploit the proposed procedure for solving L1 ($\lambda \rightarrow 0$), L2 ($\lambda = 1$) and L1-L2 ($0 < \lambda < 1$) SVM training problems.

6.4 Results

Open rooms for improving smartphone-based HAR exist. In this section, we explore the performance of the various linear SVM models proposed in this chapter and apply them to our HAR dataset. We focus on the evaluation of different aspects: (i) the introduction of a larger set of

Algorithm 1: Algorithm for solving Problem (6.7)

Data: D_n , λ , C , ε and numerical precision ϵ **Result:** w^* , b^* $t = 0$, $\alpha_t = \mathbf{0}_n$, $\beta_t = \mathbf{0}_d$;**repeat**

$$\left| \begin{array}{l} \alpha_{t+1}, b_{t+1} = \arg \min_{\alpha} P1(\alpha, \beta_t); \\ \beta_{t+1} = \max \left[-\frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d, \min \left[\frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d, (\alpha_{t+1}^T Y X)^T \right] \right]; \\ t = t + 1; \end{array} \right.$$
until $\left[\left(\|\alpha_t - \alpha_{t-1}\|_2^2 + \|\beta_t - \beta_{t-1}\|_2^2 \right) < \epsilon \right]$;**return** $w^* = \frac{\sqrt{\lambda}}{\lambda} (X^T Y \alpha_t + \beta_t)$, $b = b_t$

Algorithm 2: EX-SMO algorithm for solving Problem (6.7)

Data: D_n , λ , C , ε and numerical precision ϵ **Result:** w^* , b^* $t = 0$, $\alpha_t = \mathbf{0}_n$, $\beta_t = \mathbf{0}_d$;**repeat**

$$\left| \begin{array}{l} \alpha_{t+1}, b_{t+1} = \arg \min_{\alpha} P1(\alpha, \beta_t) \text{ by running } \frac{n}{2} \text{ iteration of the SMO algorithm} \\ \text{[Keerthi et al., 2001]}; \\ \beta_{t+1} = \max \left[-\frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d, \min \left[\frac{(1-\lambda)}{\sqrt{\lambda}} \mathbf{1}_d, (\alpha_{t+1}^T Y X)^T \right] \right]; \\ t = t + 1; \end{array} \right.$$
until $\left[\left(\|\alpha_t - \alpha_{t-1}\|_2^2 + \|\beta_t - \beta_{t-1}\|_2^2 \right) < \epsilon \right]$;**return** $w^* = \frac{\sqrt{\lambda}}{\lambda} (X^T Y \alpha_t + \beta_t)$, $b = b_t$

gyroscope-based signals along with the ones that come from the smartphone’s accelerometer for the recognition of activities; (ii) the selection of the most useful features and simpler, though effective, models to make HAR more suitable for devices with limited battery life and computational restrictions. (iii) the exploration of alternative linear SVM approaches that allows to control sparsity and dimensionality reduction.

These three issues are targeted in this section in the following way: regarding point (i), we fully exploit the HAR dataset, which contains gyroscope measures plus a set of previously suggested features from the accelerometer [Bao and Intille, 2004; Karantonis et al., 2006; Khan et al., 2010a] and verify the improvements that can be achieved in the classification performance of the algorithm (Section 6.4.2). Concerning issue (ii), we resort to effective SVM classifiers and implement two feature selection mechanisms to allow faster and computationally non-intensive recognition: on the one hand, we evaluate features discriminating on sensor type and domain (either time or frequency) in Section 6.4.2; on the other hand, MC-L1-SVM models are implemented, allowing to perform an automatic selection of significant features emerging from the training set while keeping the appealing classification performance of conventional SVMs (Section 6.4.3). Finally, point (iii) is dealt with the analysis of the MC-L1-L2-SVM approach in order to find the ideal trade off point that combines the effectiveness of the L2-SVM and the feature selection characteristics of L1-SVMs, this is presented in Section 6.4.4.

One of the main drawbacks of L1-SVMs consists in the impossibility of resorting to non-linearity to improve classification accuracy through the kernel trick [Vapnik, 1998]: for targeting this issue, here we compare the use of conventional non-linear models with the one of linear classifiers in the particular case of HAR, showing how the latter ones are characterized by similar performance/complexity ratios than the former ones (Section 6.4.1). The HAR Dataset, that we use in the forthcoming analysis is \mathcal{D}_2 composed of a large set of time and the frequency domain features extracted from the accelerometer and gyroscope signals and also subsets of this, such as \mathcal{D}_{2T} , which only include time domain features.

6.4.1 Linear vs. Non-Linear SVMs

The first experiment aimed to compare the performance of SVM models based linear and non-linear kernels. For that, we performed the training of \mathcal{D}_2 using two models: first, the standard OVA MC-GK-SVM ($K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2)$). For model selection of this SVM, we used a KCV with $k = 10$ and we searched for the two SVM hyperparameters C and γ . With C in the range $[10^{-4}, 10^2]$ and γ between $[10^{-4}, 10^2]$. Both partitioned in 20 points equally spaced in a logarithmic scale. The second method used was the MultiClass L2-SVM (MC-L2-SVM), which was also trained using also $k = 10$ and the same partition for its only hyperparameter C .

The confusion matrices in Table 6.1 depict the classification results obtained given the 6 BAs using the complete set of features from \mathcal{D}_2 . The accuracies achieved with the two methods are very similar, thus showing the equivalence between these two models. The linear approach performs slightly better, only varying in a 0.04% with respect to the non-linear one. Results

also show sensitivity and specificity measures for all the activities.

Some large datasets have shown similar classification performance when linear or non-linear approaches are used [Schölkopf and Smola, 2001], meaning that data mapping into a higher-dimensional space is not always required. The advantage of a linear kernel is the rapider prediction that can be achieved. The MC-GK-SVM model selection was performed doing grid-search over two hyperparameters which is also computationally more expensive than MC-L2-SVM which only uses one. Our application only considers online prediction while learning is performed offline. The idea behind this experiment was to study the possibility of employing a linear classifier for the HAR dataset instead of a more complex approach without risking recognition performance. This is also justifiable from the SLT perspective in which the easiest solution that properly classifies the data is always preferred [Vapnik, 1995].

Taking into account these findings, the linear approach is consequently favored for the prediction of activities, more specifically for its application in limited resources devices: in fact, the prediction phase is much faster than the kernelized approach and linear models allow to exploit more sophisticated dimensionality reduction approaches, as will be shown in Section 6.4.3. From now on in this thesis, we will only make use of linear SVM models for HAR.

6.4.2 Selection of subsets of features

The second set of experiments consisted of evaluating the available features in the dataset aiming at a significant reduction in their number. This was firstly achieved by separating the inputs in groups with respect to: (i) the type of sensor employed, namely *Accelerometer* (*A*) and *Gyroscope* (*G*); and (ii) the feature domain, namely *Time* (*T*) and *Frequency* (*F*). The analysis aims to gather some evidence of the benefits the addition of gyroscope signals bring into the HAR system, and also assesses the need of frequency domain features for improving the recognition performance.

In practice, we expect to balance the trade-off between the addition of meaningful features and the removal of the ones that are redundant or that require expensive computations for their estimation. For such purposes, we tested the possible combinations of feature groups and computed the system accuracy performed by a linear SVM model (MC-L2-SVM), trained on the corresponding subset of features. Table 6.2 presents the results.

These results suggest that the more relevant features are added into the system, the higher classification accuracy is achieved. The whole set of features (AGTF) provides the best performance. However, from this analysis, it is also noticeable that frequency-related inputs are only producing a very small improvement in this application as they do not largely affect recognition performance when compared with the AGT subset. This indicates they are not strictly necessary for the classification as we already have a significant number of features in the time domain. Moreover, frequency domain features requires an extra effort for their derivation such as the calculation of the FFT for each window sample (Section 4.3.2). The accuracy difference between the AGT and AGTF subsets is only 0.24% which is a negligible value.

MC-GK-SVM — \mathcal{D}_2								
Activity	WK	WU	WD	SI	ST	LD	Sensitivity	Specificity
WK	486	6	4	0	0	0	97.98%	99.31%
WU	12	458	1	0	0	0	97.24%	98.59%
WD	5	27	388	0	0	0	92.38%	99.80%
SI	0	2	0	450	39	0	91.65%	99.71%
ST	0	0	0	7	525	0	98.68%	98.39%
LD	0	0	0	0	0	537	100.00%	100.00%
Accuracy	96.50%							
MC-L2-SVM — \mathcal{D}_2								
Activity	WK	WU	WD	SI	ST	LD	Sensitivity	Specificity
WK	493	1	2	0	0	0	99.40%	99.10%
WU	20	450	1	0	0	0	95.54%	99.56%
WD	2	6	412	0	0	0	98.10%	99.88%
SI	0	4	0	435	51	1	88.59%	99.43%
ST	0	0	0	14	518	0	97.37%	97.89%
LD	0	0	0	0	0	537	100.00%	99.96%
Accuracy	96.54%							
MC-L2-SVM — \mathcal{D}_{2T}								
Activity	WK	WU	WD	SI	ST	LD	Sensitivity	Specificity
WK	494	2	0	0	0	0	99.60%	98.82%
WU	20	451	0	0	0	0	95.75%	99.80%
WD	8	0	412	0	0	0	98.10%	100.00%
SI	0	3	0	428	60	0	87.17%	99.39%
ST	1	0	0	15	516	0	96.99%	97.52%
LD	0	0	0	0	0	537	100.00%	100.00%
Accuracy	96.30%							

Table 6.1: Confusion Matrices for the MC-GK-SVM algorithm with \mathcal{D}_2 (Top), MC-L2-SVM with \mathcal{D}_2 (center), and with only time domain features \mathcal{D}_{2T} (bottom)

Results also allow to gather some evidence of the benefits that the incorporation of gyroscope signals brings into the HAR system that counterbalance the limited slowdown in prediction due to the presence of these extra features. It can be observed, for instance, the visible accuracy difference between the ATF and AGTF feature groups which diverge only on the use of the gyroscope (5.31%). It is also noticeable, on the other hand, that the models trained with sets using only gyroscope features (GT, GTF) have a lower performance. This suggests that the use of the gyroscope by its own is not appropriate for its application in HAR, despite enhancing the recognition when exploited concurrently with accelerometers.

6.4.3 Dimensionality reduction with L1-SVM

In this section we explore the use of the MC-L1-SVM instead of the conventional MC-L2-SVM which does not perform any dimensionality reduction [Khan et al., 2012]. This fact is desirable in some practical applications to highlight relevant features as well as to reduce the

Acc	Gyro	Time	Freq	Feature groups	N. Features	MC-L2-SVM	MC-L1-SVM
0	1	1	0	GT	108	78.02%	—
0	1	1	1	GTF	213	81.04%	—
1	0	1	0	AT	164	90.62%	—
1	0	1	1	ATF	348	91.23%	—
1	1	1	0	AGT (\mathcal{D}_{2T})	272	96.30%	96.61%
1	1	1	1	AGTF (\mathcal{D}_2)	561	96.54%	96.17%

Table 6.2: Experiments with different feature subsets and conventional linear SVM models.

computational burden of performing the classification of new samples. Table 6.3 presents the confusion matrices resulting from the classification of the \mathcal{D}_2 and \mathcal{D}_{2T} datasets with this SVM algorithm. Moreover, in Table 6.4 we display the comparison of MC-L2-SVM and MC-L1-SVM, both in terms of accuracy and number of selected features (remembering that L2 procedures do not perform any dimensionality reduction).

In particular, we considered only the groups of features that showed to be necessary for HAR purposes according to the results derived in the previous section. These are the AGTF and AGT subsets which correspond to \mathcal{D}_2 and \mathcal{D}_{2T} respectively. They obtained classification accuracies equal to 96.17% and 96.61%. It is worth noting that L1 models perform comparably to L2 models with our datasets, furthermore allowing to remarkably reduce the dimensionality of the problem. In the literature it is common to find L2 models to outperform. Therefore, these findings show the classification performance of L1 is probably due to the intrinsic filtering of noisy features, which negatively afflict L2 classifiers.

Dimensionality reduction is also an important aspect in the comparison of the L1- and L2-Norm algorithms. As we can see in Table 6.4, MC-L1-SVM achieves an effective reduction in the number of features up to 42% of the total number of features. This means that the discarded features do not need to be estimated in our HAR system, speeding up the recognition process from feature extraction. The average dimensionality reduction $\bar{\rho}$ is also showing that each binary classifier needs less than 20% of the total number of available features for classification. This indicates that some features help to classify better certain activities from others. Moreover, the execution time of the FFP of each L1-SVM can be up to 5 times faster than L2-SVM, assuming a single process implementation. Overall, we can argue that the use of the L1 models is a suitable option for our HAR system because it brings the advantages of a linear classifier that performs dimensionality reduction while maintaining the system performance.

Figure 6.1 depicts the results of the hyperparameter optimization process for the MC-L1-SVM and MC-L2-SVM where the C_m values for each activity classifier are selected using a KCV with $k = 10$ on the training set. All the values are contained within the $[1, 100]$ interval. It is also visible that initial values of C present low accuracy, specially in the L1-SVM classifiers which display a sharp reduction in this area. Greater values display a nearly constant accuracy within the explored range. The selection process consist of a sequential evaluation of the accuracy with different C values. Only if a new value delivers greater accuracy than a previous recorded one, it is replaced and selected as optimal C . This procedure prevents the model from overfitting by

taking very large values of C .

As a final remark, the results obtained with the MC-L1-SVM algorithm are analogous to the ones obtained at the ESANN 2013 HAR competition (Refer to Section 4.4.2) in which contestants were challenged to propose novel approaches for the recognition of activities using the same HAR dataset. The work of [Romera-Paredes et al., 2013] achieved a maximum classification accuracy of 96.4%, where an OVO SVM classification approach [Rifkin and Klautau, 2004] was employed for the recognition task instead of our OVA MC-L1-SVM approach.

MC-L1-SVM — \mathcal{D}_2								
Activity	WK	WU	WD	SI	ST	LD	Sensitivity	Specificity
WK	493	2	1	0	0	0	99.40%	98.98%
WU	23	445	1	2	0	0	94.48%	99.56%
WD	2	5	412	1	0	0	98.10%	99.92%
SI	0	4	0	428	59	0	87.17%	99.35%
ST	0	0	0	13	519	0	97.56%	97.56%
LD	0	0	0	0	0	537	100.00%	100.00%
Accuracy	96.17%							
MC-L1-SVM — \mathcal{D}_{2T}								
Activity	WK	WU	WD	SI	ST	LD	Sensitivity	Specificity
WK	492	3	1	0	0	0	99.19%	99.06%
WU	18	452	0	1	0	0	95.97%	99.80%
WD	5	1	413	1	0	0	98.33%	99.92%
SI	0	1	1	436	53	0	88.80%	99.31%
ST	0	0	0	15	517	0	97.18%	97.81%
LD	0	0	0	0	0	537	100.00%	100.00%
Accuracy	96.61%							

Table 6.3: Confusion Matrices for the linear method MC-L1-SVM with \mathcal{D}_2 (top) and \mathcal{D}_{2T} (bottom)

Feature group	d	MC-L2-SVM	MC-L1-SVM			
		Accuracy	Accuracy	$d - \mathcal{S} $	ρ	$\bar{\rho}$
AGT	272	96.06%	96.61%	168	61.76%	19.24%
AGTF	561	96.54%	96.17%	239	42.6%	12.03%

Table 6.4: Comparison between MC-L2-SVM and MC-L1-SVM regarding accuracy and dimensionality reduction.

6.4.4 L1-L2 SVM with HAR data

In order to derive the experimental results with the MC-L1-L2-SVM algorithm, we replicated the model selection methodology adopted in Section 6.4.1 with the difference that we have an additional hyperparameter to tune: $\lambda \in (0, 1]$. In order to compare L1, L2 and L1-L2 SVM solutions, we set $\lambda = \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$.

After an inspection of the results obtained in the previous two sections with the different

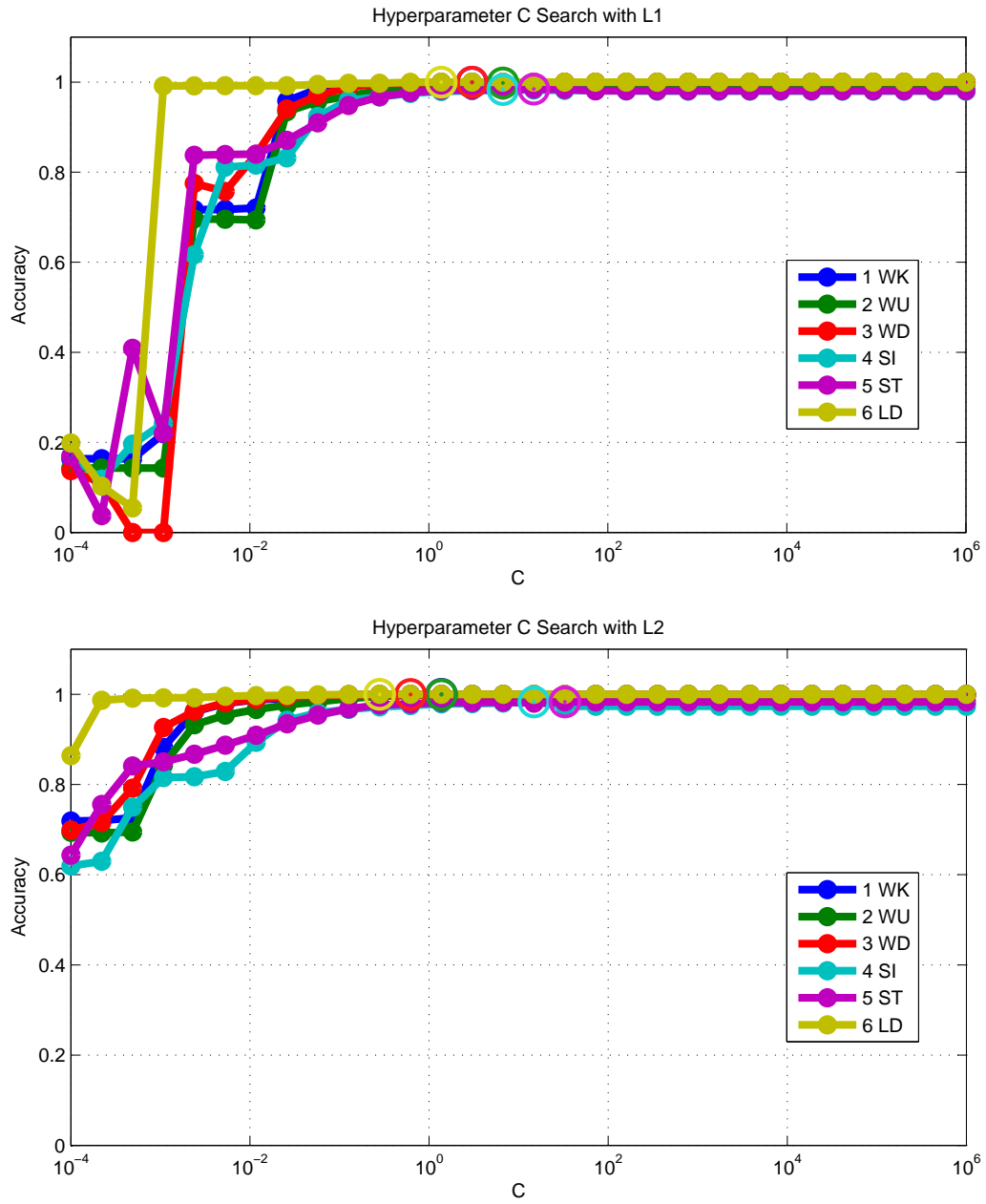


Figure 6.1: Hyperparameter C grid search results per class for MC-L1-SVM (top) and MC-L1-SVM (bottom). Circled points correspond to the selected C values.

combinations of features, we decided to work only with time features because the improvement granted with frequency domain features is minimal and it also increases the system’s computational load in the feature extraction and prediction stages. Therefore, we work with \mathcal{D}_{2T} in this experiment. Table 6.5 compares the training times for the EX-SMO training approach described in Algorithm 2 against commonly used solvers for L1 (the Simplex Method for Linear Programming (SMLP) [Flannery et al., 1992]) and L2 SVM (the conventional SMO [Keerthi et al., 2001]). Table 6.5 shows that EX-SMO performs comparably to SMO on L2 problems and outstrips SMLP on training L1 SVMs, albeit EX-SMO effectiveness tends to decrease as $\lambda \rightarrow 0$: this is expected, as we are using a QP tool to solve an (almost) LP problem, and this suboptimal approach leads to a slight loss in the algorithm’s performance.

SVM	L1-SVM	L1-L2-SVM							L2-SVM
		λ							
		0.001	0.005	0.01	0.05	0.1	0.5	1	
Training	SMLP	EX-SMO							SMO
Time (h)	2.54	2.37	1.97	1.54	1.47	1.39	1.32	1.14	1.13

Table 6.5: Comparison of the EX-SMO algorithm training time (in hours) against SMLP and SMO.

Table 6.6 reports the confusion matrices for L1, L1-L2 and L2 SVMs obtained on the \mathcal{D}_{2T} . We do not present results for different solvers as no differences are shown in them. Different from the expected behavior, we observe a regular classification performance in all the methods in terms of accuracy (with variations below 1%). However, we found an interesting result and it is that when $\lambda = 0.05$ the highest accuracy (96.91%) is achieved instead of MC-L2-SVM and MC-L1-SVM. This finding is possibly linked to the fact that this intermediate solution is selecting relevant features and filtering noisy ones, two aspects that cannot be properly dealt with extreme cases such as when $\lambda \rightarrow 0$ and when $\lambda = 1$ respectively.

Moreover, in Table 6.7 we collect the experiment results for all the values of λ . They include classification accuracy, dimensionality reduction (overall ρ and average $\bar{\rho}$), and grouping ability σ . First, we can observe that ρ decreases (or increases) with λ . This corroborates the dimensionality reduction capability of L1-SVM and equivalently L1-L2-SVM with small values of λ . However, though the dimensionality reduction capability is maximized for L1-SVM, feature grouping effects, namely the ability of the algorithm in selecting (or neglecting) clusters of highly cross-correlated inputs, are usually absent when $\lambda \rightarrow 0$, although they are desirable in order to have more insights on the informative content of each input [Segal et al., 2003]. In order to evaluate whether L1-L2-SVM is able to overcome these L1-related issues, as expected from literature, we computed the correlation matrix $M^C \in \mathbb{R}^{d \times d}$ of X and we created feature clusters by joining the 10 most cross-correlated inputs. Our purpose was to verify the percentage σ of clusters features selected (or neglected) by the different procedures (ranging from L1-SVM to L2-SVM): a high value for σ is obviously desirable. Results are also shown in the table and it is thus worth noting that: a very small subset of features (L1-SVM) is necessary to guarantee

MC-L1-SVM							MC-L1-L2-SVM $\lambda = 0.001$						
Activity	WK	WU	WD	SI	ST	LD	Activity	WK	WU	WD	SI	ST	LD
WK	492	3	1	0	0	0	WK	493	2	1	0	0	0
WU	18	452	0	1	0	0	WU	16	455	0	0	0	0
WD	5	1	413	1	0	0	WD	5	2	413	0	0	0
SI	0	1	1	436	53	0	SI	0	1	0	436	54	0
ST	0	0	0	15	517	0	ST	0	0	0	14	518	0
LD	0	0	0	0	0	537	LD	0	0	0	0	0	537
Accuracy	96.61%						Accuracy	96.78%					
MC-L1-L2-SVM $\lambda = 0.005$							MC-L1-L2-SVM $\lambda = 0.01$						
Activity	WK	WU	WD	SI	ST	LD	Activity	WK	WU	WD	SI	ST	LD
WK	493	2	1	0	0	0	WK	493	2	1	0	0	0
WU	15	456	0	0	0	0	WU	15	456	0	0	0	0
WD	5	2	413	0	0	0	WD	4	2	413	1	0	0
SI	0	2	0	434	55	0	SI	0	2	0	436	53	0
ST	0	0	0	14	518	0	ST	0	0	0	14	518	0
LD	0	0	0	0	0	537	LD	0	0	0	0	0	537
Accuracy	96.74%						Accuracy	96.81%					
MC-L1-L2-SVM $\lambda = 0.05$							MC-L1-L2-SVM $\lambda = 0.1$						
Activity	WK	WU	WD	SI	ST	LD	Activity	WK	WU	WD	SI	ST	LD
WK	493	2	1	0	0	0	WK	493	2	1	0	0	0
WU	15	456	0	0	0	0	WU	17	454	0	0	0	0
WD	4	2	413	1	0	0	WD	5	2	412	1	0	0
SI	0	1	0	437	53	0	SI	0	1	0	436	54	0
ST	0	0	0	12	520	0	ST	0	0	0	13	519	0
LD	0	0	0	0	0	537	LD	0	0	0	0	0	537
Accuracy	96.91%						Accuracy	96.74%					
MC-L1-L2-SVM $\lambda = 0.5$							MC-L1-L2-SVM $\lambda = 1$						
Activity	WK	WU	WD	SI	ST	LD	Activity	WK	WU	WD	SI	ST	LD
WK	493	2	1	0	0	0	WK	494	2	0	0	0	0
WU	18	453	0	0	0	0	WU	20	451	0	0	0	0
WD	3	1	414	1	1	0	WD	8	0	412	0	0	0
SI	0	3	0	433	55	0	SI	0	3	0	428	60	0
ST	0	0	0	12	520	0	ST	1	0	0	15	516	0
LD	0	0	0	0	0	537	LD	0	0	0	0	0	537
Accuracy	96.71%						Accuracy	96.30%					
MC-L2-SVM													
Activity		WK	WU	WD	SI	ST	LD						
WK		494	2	0	0	0	0						
WU		20	451	0	0	0	0						
WD		8	0	412	0	0	0						
SI		0	3	0	428	60	0						
ST		1	0	0	15	516	0						
LD		0	0	0	0	0	537						
Accuracy		96.30%											

Table 6.6: Confusion matrices obtained with \mathcal{D}_{2T} and the MC-L1-L2-SVM algorithm using different values of λ .

an acceptable classification performance, though grouping effects are limited. By balancing the effects of L1 and L2 regularization terms, we can decide whether we want a higher accuracy, a smaller number of features or a higher grouping ability.

In the particular case of HAR using smartphones, as we are targeting the minimization of the computational burden to maximize battery duration and we are only partially interested in having insights on information content of each input, L1-L2 SVMs with (very) small values of λ are preferable, but this could not be, in general, the best choice. The advantage of a very flexible solver that copes with L1, L1-L2 and L2 SVMs, as the one presented in this chapter, consists in

Table 6.7: Accuracy, feature selection and grouping ability for the different approaches with \mathcal{D}_{2T} .

Method	Algorithm	% Accuracy	% ρ	% $\bar{\rho}$	% σ
L1 SVM	SMLP	96.61	61.76	19.24	0.0
L1-L2 SVM $\lambda = 0.001$	EX-SMO	96.78	61.40	19.42	0.0
L1-L2 SVM $\lambda = 0.005$	EX-SMO	96.74	62.87	20.34	0.0
L1-L2 SVM $\lambda = 0.01$	EX-SMO	96.81	66.18	21.38	10.8
L1-L2 SVM $\lambda = 0.05$	EX-SMO	96.91	72.06	24.88	20.4
L1-L2 SVM $\lambda = 0.1$	EX-SMO	96.74	74.63	27.45	60.5
L1-L2 SVM $\lambda = 0.5$	EX-SMO	96.71	91.54	39.58	90.6
L1-L2 SVM $\lambda = 1$	EX-SMO	96.30	100.00	91.67	100.0
L2 SVM	SMO	96.30	100.00	91.67	100.0

the possibility of identifying the best application-dependent trade-off between performance and dimensionality reduction, at the expense of a very small implementation effort.

6.5 Summary

In this chapter, we showed the benefits of adding gyroscope information into a HAR system based on smartphone technology. We verified that a set of common BAs can be accurately classified when this sensor is used along with the accelerometer. We also explored four SVM algorithms including linear (L1-SVM, L1-L2-SVM and the conventional L2-SVM) and non-linear (GK-SVM) approaches on the \mathcal{D}_3 dataset. We found similar performance between them in terms of classification accuracy, but our selection criterion was subject to prediction speed and the possibility of applying them in devices with limited resources to provide less computational complexity and energy consumption.

Linear approaches exhibited the best trade off between accuracy and prediction speed, conferring distinctive benefits to the MC-L1-SVM, which provides itself a reduction of the effective number of features needed for the prediction of BAs. Furthermore, the study between different feature domains lead us to disregard frequency domain features as they were not only marginally contributing to the recognition performance but also adding expensive computations for their estimation. The ideal set of features selected for our application was the AGT, which only takes into account time domain features from the accelerometer and the gyroscope. Additionally, we showed in experiments with the MC-L1-L2-SVM algorithm, which combines L1 and L2 norms, that it is provided with interesting characteristics that can be exploited in order to fine tune different aspects of the learned model such as accuracy, dimensionality reduction and grouping ability that can be convenient in different applications.

Moreover, we proposed in this chapter a novel approach for training L1-L2-SVM classifiers. The proposed method is characterized by two main advantages: (i) it is flexible, as it allows to solve L1, L1-L2 and L2 SVM problems and to properly tune the trade-off between dimensionality reduction and performance; (ii) it builds on conventional QP solvers, thus can be implemented

with a minimal effort. We tested our approach on our [HAR](#) application, that allowed us to compare the proposed approach with state-of-the-art alternatives and to highlight the usefulness of such a flexible solver in a real-world practical problem.

Chapter 7

Online Recognition with Postural Transition Awareness

7.1 Introduction

In this chapter we introduce an online system for the classification of human activities using smartphones: **PTA-HAR**. This system is an adaptation of the previously presented **L-HAR** system that deals with frequent **PTs** while sequences of **BAs** are carried out. As it will be shown, when **PTs** are not properly identified in a **HAR** system, they can affect its performance by triggering the appearance of false positives. In most **HAR** systems, transitions between activities are usually ignored since their incidence is generally low and duration is shorter when compared against other activities. Nevertheless, this assumption depends on each application and, therefore, it should be considered accordingly. For instance, in the design of activity monitoring systems for the disabled during rehabilitation practices, or for athletes while performing fitness/gymnasium activities, it is important to identify **PTs** because in these cases is common to do different tasks in short time periods. Incorrect classification of **PTs** can significantly affect the performance of the recognition system because transitions appear more frequently.

The proposed system to deal with **PTs** is based on a probabilistic analysis of consecutive predictions. More concretely, two different methods to handle these transitory events are explored. Both follow a similar recognition pipeline although in one of them **PTs** are learned by the **ML** classifier (**PTA-7A**) along with the **BAs**, while in the other they are handled only through filtering (**PTA-6A**). We employ linear **SVMs** whose probability estimates are analyzed in conjunction with the predictions of its neighboring samples in time and interpreted as activity signals. These signals are then heuristically filtered in order to clean and suppress unwanted noise since we assume that, in real-life applications, contiguous events are in general correlated.

We present the benefits of these two methods and show, through experiments over the **D_{3T}** **HAR** dataset which includes labels of **BAs** and **PTs**, that they outperform previously explored systems which exclude these transitions (e.g. **L-HAR**). Moreover, results depict their differences and show their usability is application-dependent. We also propose the detection of unknown

activities (UAs) that the system is not able to recognize or match to any of the learned activities based on probability estimates. Dealing with them can help make systems more functional for a variety of applications where other activities are unaccounted for in the learning process. We interpret UAs as an arbitrary subspace that contains unknown activities not learned by the ML algorithm, similarly to the NULL class presented in [Bulling et al., 2014].

Following a similar framework to previous chapters, the proposed system exploits a waist-mounted smartphone with embedded accelerometer and gyroscope. Moreover, it aims to provide near real-time activity prediction for monitoring applications within the device or through wireless connectivity to others. This work is, to the best of our knowledge, the first to evaluate the occurrence of PTs on online smartphone-based HAR systems. The studied PTs in this work are: *stand-to-sit*, *sit-to-stand*, *sit-to-lie*, *lie-to-sit*, *stand-to-lie* and *lie-to-stand*.

The following sections are organized as follows: Section 7.2 gives an introduction about postural transitions in HAR. This is followed by the description of the proposed methodology chosen for the two HAR methods and the main stages of the online system in Section 7.3. In Section 7.4, we present the experiments conducted regarding the built HAR dataset, the selection of the system performance evaluation metric and the developed smartphone app. Moreover, in Section 7.5, we depict the achieved results and the comparison against our previous system (L-HAR). Finally in Section 7.6 we summarize the obtained results along with critical a view of the proposed approach.

7.2 Postural Transitions in HAR Systems

A PT is an event with limited duration determined by its start and end times. Generally, this duration varies among healthy individuals to some extent. PTs are bounded by other two activities and represent the transition period between the two. Conversely, BAs such as *standing* and *walking* can prolong for a longer term. Data collection for these two types of activity is also different as PTs need to be executed repeatedly to get separate samples and BAs, as they are continuous, allows many window samples to be taken from a single test only limited by its time extent. Most of the proposed HAR systems in the literature do not include PTs in the set of studied activities (Section 3.3.1). Some assume, for instance, that the detection of two different SPs defines the occurrence of a PT (e.g. *standing* followed by *sitting* assumes a *stand-to-sit* transition between them). In [Lara and Labrador, 2012a], it is pointed out that PTs can be ignored in some situations if its occurrence is less frequent and duration is much shorter against other activities.

However, as this assumption is application specific, PTs can directly influence the system performance in the other cases. Therefore, they should be taken into consideration during the design and selection of the right recognition algorithms. HAR systems such as the one presented in [Khan et al., 2010b], combine 7 BAs with 7 PTs in the classification. In cases like this, however, large multiclass problems may arise which could eventually increase the prediction of false negatives of the activities we are particularly interested in (e.g. BAs).

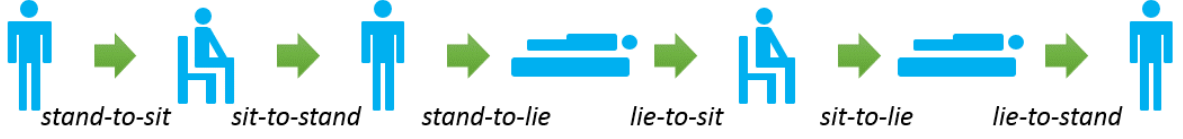


Figure 7.1: Postural Transitions from the 3 studied Static Postures: standing, sitting and lying-down

Online smartphone-based HAR systems have not yet studied PTs along with other activities. However, in cases such as in [Zhang et al., 2010] an offline HAR system combines various PTs as a single class along with other activities for a daily monitoring application. Moreover, in [Rednic et al., 2013] an approach for performing posture classification is proposed. Even though this is done using a multi-accelerometer BSN instead of smartphones, they also investigate the effects of PTs in their system and introduce exponentially weighted voting transition filters in order to improve by 1% the accuracy of their recognition system. They explore 7 activities usually performed during explosive ordnance disposal operations (e.g. kneeling, crawling and sitting). In this work, we present, in contrast, a method that aims for its real-time execution on a single smartphone and deals with activity information in the vicinity of each event occurrence.

7.3 HAR with Postural Transitions Awareness

This section presents the two proposed posture-aware methods that deal with the occurrence of PTs in an online activity recognition system. Their difference relies on the way they handle PTs and the number of activities that are learned in the ML algorithm. These are:

- **PTA-6A:** This first method, as its name designates, only takes into account the 6 studied BAs (standing, sitting, lying-down, walking, walking-downstairs and walking-upstairs) for the ML learning process with SVMs, while PTs are initially disregarded. Following the ML algorithm, a temporal filtering stage is introduced. It deals with PTs and misclassifications of BAs based on the classifier output of contiguous window samples.
- **PTA-7A:** The second method takes into account seven classes for the ML algorithm: the original 6 BAs plus an additional class which represents all the 6 PTs at once. The PTs are: StSi, SiSt, SiLi, LiSi, StLi, and LiSt (Refer to Figure 7.1 for an illustration). Similarly to the previous method, temporal filtering is also applied after prediction, however, PTs are handled differently provided that they are one of the ML algorithm possible outputs.

The entire recognition algorithm is composed of three main stages which are depicted in Figure 7.2. Moreover, a pseudocode of the entire recognition process is presented in Algorithm 3.

The first stage, *signal conditioning and feature extraction*, comprises data acquisition and signal conditioning from the inertial sensors to obtain the features that characterize each activity

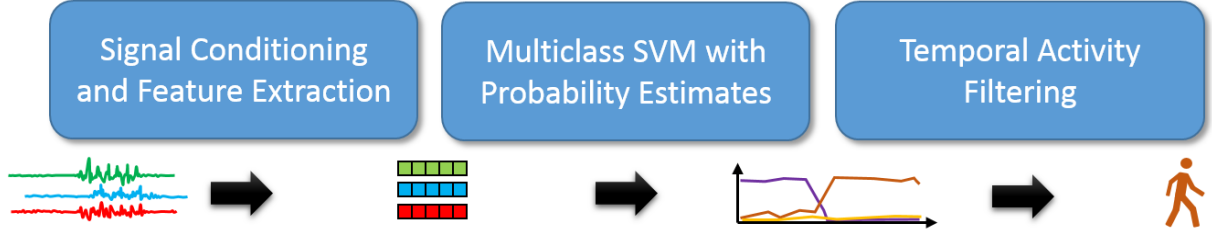


Figure 7.2: Online HAR algorithm stages. This illustration depicts schematically the input and output of each block.

sample. These features become the input of second stage, *multiclass SVM with probability estimates*, where they are evaluated for activity prediction. From each sample an array is extracted to indicate the probability of belonging to the learned classes. In the third stage, *temporal activity filtering*, these probabilities are then joined with the predictions of previous activity samples and processed by means of temporal filtering. This is achieved by applying a set of defined heuristic filters that only allow natural sequences of activities. They consider PTs and also unknown conditions (e.g. when output probabilities are marginal) for this purpose.

Both recognition methods (PTA-6A and PTA-7A) fit in the same pipeline with only change in the number of trained classes of the MC-L1-SVM.

7.3.1 Signal Conditioning and Feature Extraction

The inputs of the PTA-HAR system are the raw triaxial linear acceleration $\mathbf{a}_r(t)$ and angular velocity $\boldsymbol{\omega}_r(t)$ time signals. These are read at a constant frequency of 50Hz in the process which performs the signal conditioning: the *ProcessInertialSignals()* function in Algorithm 3. The set of filters used are for this task are: (i) noise reduction, whose transfer function is represented by $H_1()$, is achieved by applying a third-order median filter and a third-order low-pass Butterworth filter (cutoff frequency = 20Hz). These filters allow to obtain clean triaxial acceleration $\mathbf{a}_\tau(t)$ signal. Angular velocity $\boldsymbol{\omega}(t)$ is additionally processed with a high pass filter (0.3Hz cutoff frequency), represented by the transfer function $H_2()$, in order to remove any bias in the signal. (ii) The segmentation of the acceleration signal into gravity $\mathbf{g}(t)$ and acceleration due to body motion $\mathbf{a}(t)$. This is possible by also high-pass filtering the acceleration $\mathbf{a}_\tau(t)$ with $H_2()$ to obtain $\mathbf{a}(t)$. $\mathbf{g}(t)$ is subsequently found by subtracting $\mathbf{a}(t)$ from $\mathbf{a}_\tau(t)$ (Refer to Section 4.3.2 for more details).

In addition, the *OnlinePrediction()* function, which is in charge of the recognition of activities, is periodically executed to obtain and classify window samples (A, G, Ω) extracted from the filtered triaxial inertial signals $(\mathbf{a}(t), \mathbf{g}(t), \boldsymbol{\omega}(t))$ over a period T . Its periodicity satisfies the sliding-windows criteria: a time span of 2.56s and 50% overlap between them. Features are extracted from these window samples through measures in the time domain (Section 4.3.3), that we represent with the $\phi()$ function, in order to match a reduced set of features from dataset \mathcal{D}_{3T} according to the MC-L1-SVM learning intrinsic feature selection (Section 6.2). In Chapter 6 we

Algorithm 3: HAR Online Algorithm for the PTA-HAR system**Require:**

\mathbf{a} : Triaxial linear acceleration
 $\boldsymbol{\omega}$: Triaxial angular velocity
 \mathbf{g} : Gravity
 $H_1(\cdot)$: Noise reduction transfer function (Section 7.3.1)
 $H_2(\cdot)$: Body acceleration transfer function (Section 7.3.1)
 $\phi(\cdot)$: Feature extraction function (Section 7.3.1)
 T : Windows size
 m : Number of classes
 \mathbf{p} : activity probability vector $\mathbf{p} = [p_1, \dots, m]^T$
 P : Buffer of probability vectors $P \in \mathbb{R}^{m \times m}$
 P' : Filtered buffer of probability vectors
 \mathbf{z} : Buffer of discrete activity predictions $\mathbf{z} \in \mathbb{R}^s$
 $\Phi(\cdot)$: Probability filtering function (Section 7.3.3.1)
 $\Psi(\cdot)$: Discrete filtering function (Section 7.3.3.2)

function ProcessInertialSignals($\mathbf{a}_r(t), \boldsymbol{\omega}_r(t)$)

```

     $\mathbf{a}_\tau(t) = H_1(\mathbf{a}_r(t)),$  // Noise Filtering
     $\boldsymbol{\omega}(t) = H_2(H_1(\boldsymbol{\omega}_r(t)))$ 
     $\mathbf{a}(t) = H_2(\mathbf{a}_\tau(t))$  // Body acceleration Extraction
     $\mathbf{g}(t) = \mathbf{a}_\tau(t) - \mathbf{a}(t)$  // Gravity extraction
    return  $\mathbf{a}(t), \mathbf{g}(t), \boldsymbol{\omega}(t)$ 
  
```

end**function OnlinePrediction($t, \mathbf{a}(t), \mathbf{g}(t), \boldsymbol{\omega}(t), B, \mathbf{z}$)**

```

     $A = \{\mathbf{a}(t') : t' \in [t-T, \dots, t]\},$  // Window sampling
     $G = \{\mathbf{g}(t') : t' \in [t-T, \dots, t]\},$ 
     $\Omega = \{\boldsymbol{\omega}(t') : t' \in [t-T, \dots, t]\}$ 
     $\mathbf{x} = \phi(A, G, \Omega)$  // Feature Extraction and Normalization
    for  $c \in \{1, \dots, m\}$  do // Multiclass SVM
         $f_c(\mathbf{x}) = \mathbf{w}_c^T \mathbf{x} + b_c$  // FFP
         $p_i = 1 / (1 + e^{(\Gamma_c f_c(\mathbf{x}) + \Delta_c)})$  // Prob. Estimation (Section 7.3.2)
    end
     $P = \begin{bmatrix} \mathbf{p}^T \\ P_{(1:end-1,:)} \end{bmatrix}$  // Append probability vector
     $P' = \Phi(P)$  // Activity probability filtering
     $c^* = \arg \max_{c \in [1, \dots, m]} P'_{(s-1,c)}$  // MAP
     $\mathbf{z} = \begin{bmatrix} c^* \\ \mathbf{z}_{(1:end-1)} \end{bmatrix}$  // Append last activity prediction
     $\hat{c} = \Psi(\mathbf{z})$  // Discrete filtering and activity estimation
    return  $\hat{c}$ 
  
```

end

opted to work with this set of features in order to reduce computational costs without noticeably affecting classification accuracy.

7.3.2 Implementation of the SVM Feed Forward Phase

The ML algorithm chosen to be implemented for the PTA-HAR system was the MC-L1-SVM. This linear algorithm provides a fast approach for performing activity recognition with a reduced set of features without affecting the prediction accuracy as presented in the results of Section 6.4.3. Model training is performed offline and the SVM FFP is performed online. The FFP formulation for each class c , as depicted in Equation (6.4), avoids the non-zero weights ($w_i = 0$) making the computation faster. The outputs of the classifiers are compared after a normalization procedure. For this, we use probability estimates as we have previously described in Section 5.2.2 in order to obtain the probability output vector $\mathbf{p}(\mathbf{x})$ which represents, for a given window sample \mathbf{x} , the collection of probabilities of being classified as a certain class c :

$$p_c(\mathbf{x}) = \frac{1}{1 + e^{(\Gamma_c f_c(\mathbf{x}) + \Delta_c)}}, \quad (7.1)$$

where the Γ_c and Δ_c parameters can be learned for each class by solving the minimization problem in Equation (5.8). At this point, it is possible to determine which of the classes (activity c^*) is the one that best represents an input sample \mathbf{x} :

$$c^* = \arg \max_c p_c(\mathbf{x}). \quad (7.2)$$

This classification approach is applicable to the PTA-6A and PTA-7A methods.

7.3.3 Temporal Activity Filtering

The classification approach presented above (Equation (7.2)) produces only a discrete output that indicates the class that best represents a test input (window sample). Moreover, it is known that the SVM is itself a static method which only depends on its input \mathbf{x} and it is not affected, for instance, by other factors such as previously predicted samples or how probable the other activities are during the FFP.

Considering the fact that, in real world situations, activities can be described as a sequence of correlated events, we take advantage of the SVM with probability estimates in a more extensive way rather than utilizing just one discrete prediction. Instead, for a window sample, the SVM prediction of probabilities at time t for all the activities $\mathbf{p}_t = [p_1, \dots, p_m]^T$ when combined with the predicted output from neighboring previous samples $\{\mathbf{p}_{t-1}, \dots, \mathbf{p}_{t-s+1}\}$ as the $P \in \mathbb{R}^{s \times m}$ matrix, can be interpreted as m activity probability signals in time. This assumption provides an advantage to improve the recognition system as we can exploit signal processing techniques such as filtering to make the overall classification system more robust. We take into account aspects such as the interrelationship within activities and the notion that only one happens at a time (e.g. during transitions which is one of our areas of interest).

Figure 7.3 shows an example of how this output looks. It is noticeable that the probability of each class increases or decreases depending on the activity performed for each particular time. Even though there is some noise in the forecasted probabilities, it is possible to predict the performed activities in a way that is very close to the ground truth by the inspection of the signals analyzed as a whole. The figure also shows two common misclassification examples in the dataset. The first error type occurs during BAs (check nearby second 9 on the figure) and is due to similarities between two SPs (*standing* and *sitting*) which usually present high interclass error. The second type occurs during PTs in between two SPs (second 4). This misclassification is generally characterized by incorrectly predicting PTs as AAs.

In more realistic circumstances, it would be desirable to acknowledge unusual events (*unknown-activity*) such as PTs when the system does not match a new window sample to any of its studied activities. The figure also depicts the expected correct predictions assuming only the 6 BAs as its classification output and the case when the *unknown-activity* class is also taken into account. Moreover, the exploration of human habits in real life examples also suggests that some sequences of events are very unlikely to happen, for instance, that a person walks upstairs right after *lying-down* and just before *standing*.

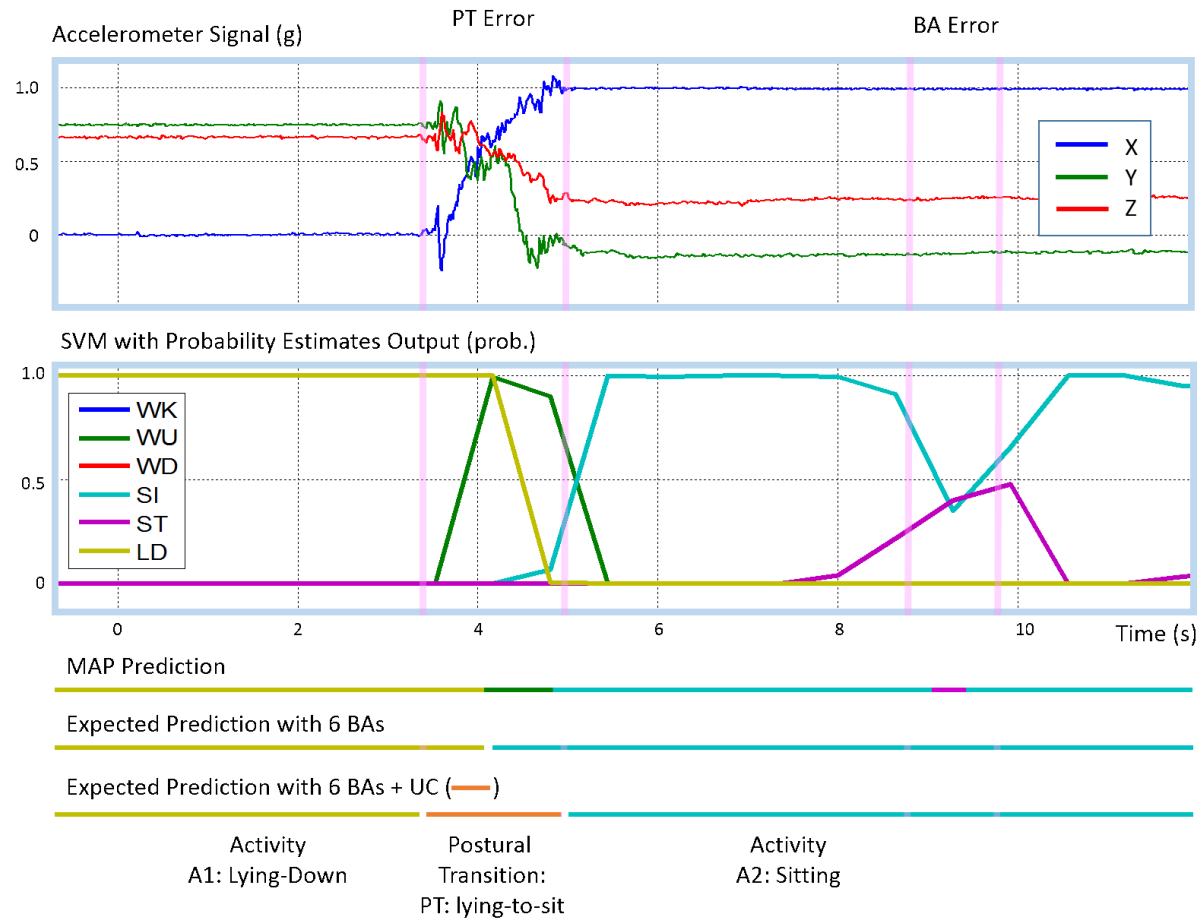


Figure 7.3: Misclassification examples during PTs and SPs (PT and BA errors). (Top) Acceleration signals clearly show the transition between the two postures. (Middle) The output of the multiclass SVM with probability estimates shows how likely each activity is for each window sample. (Bottom) The prediction of the activities using the MAP approach is compared against the expected output with the 6BAs with and without considering the unknown-activity (UA).

We have developed a set of filters to heuristically improve the probabilistic output of the SVM using temporal information from each prediction and its neighboring samples. This process is divided in two parts: *probability filtering* which directly handles the probability signals and *discrete filtering* that further filters the activity output after the discretization of probabilities into activities.

7.3.3.1 Probability Filtering

The visualization of activity probability signals in time provided us with information regarding the behavior of BAs. SPs are different from AAs with respect to the way their associated activity probabilities manifest (the normalized SVM output). This signal fluctuates more in AAs. As a result, the implemented filters are conditioned with the type of activity, whether they are an SP or not. They are rule-based and use the P matrix as input which is composed of the activity probability vectors of the last s overlapped windows. This number of windows is selected based on the filter requirements. In this application $s = 5$ which is equivalent to a prediction delay of 5.12s.

The filters use probability thresholds to define, for instance, whether a class is considered active (e.g. $p_c > threshold$) or condition the filtering of an activity based on the value of other classes. $P' = \Phi(P)$ represents the application of the probability filters over the activity sequence and they are described as follows:

Transition Filter This filter is aimed to remove peaks and transients of dynamic activities when they appear amongst static ones. This is applied to AAs as they exhibit a spiky behavior (e.g. during PTs). These usually take a short time (from 2 to 3 seconds), therefore this filter measures the length of the activation of these dynamic signals for a number of overlapping windows (maximum 3). Their filtering is also conditioned with the intensity of the SPs probability signals in the selected windows. A high probability in these indicates it is unlikely that an AA can appear simultaneously.

For the PTA-7A method, the AAs are also filtered when the PT output probability surpasses a threshold. The transition filter is not applied over the PT class because contrarily to AAs, its appearance is rather short and it is desired to be kept in such way instead of removing it.

Smoothing Filter This filter targets the probability signals during the occurrence of BAs. It helps to stabilize signal fluctuations when their probability values are greater than a threshold (0.2) within the activity sequence. This is aimed to make evident small differences between activities with high interclass misclassification e.g. *standing* and *sitting* or *walking* and *walking-upstairs*. Oscillations are smoothed using a linear interpolation.

7.3.3.2 Discrete filtering

The next step after the probability signals have been filtered is to define c^* , the most likely activity for each window sample. This is done using MAP over the probability vector ($\mathbf{p}' = P'_{(s-1,:)}\text{)$ extracted from the filtered activity matrix. From this one of the classes is selected as the predicted activity.

However, under some circumstances, the entire probability vector contains small values. This indicates that none of the classes seems to be representative of the current activity. To this end, we have defined a minimum activity threshold which is used to label samples as undefined (**unknown-activity**) when none of the classes reaches this value. This is particularly useful during PTs in the PTA-6A method as they are not learned by the SVM model. But in general, this approach can be beneficial in real life situations when the HAR system is used while activities outside the studied set occur. Consequently, these will not be categorized as any of them, instead the system will show that an unknown event has occurred.

This filter removes sporadic activities that appear for a short time and are unlikely to happen for only a window sample. It also includes cases when the **unknown-activity** is detected and its contiguous activities belong to the same class. The filter allows to relabel them as their neighbors. The final predicted activity is the result of this discrete filter $\hat{c} = \Psi(\mathbf{z})$, where \mathbf{z} is the buffer containing the last 3 predicted activities c^* .

7.4 PTA-HAR Experiments

This section presents a collection of experiments carried out for the evaluation of our PTA-HAR system. It starts with an evaluation of the \mathcal{D}_{3T} dataset focused on PTs which are analyzed in terms of duration and then compared against BAs. Moreover, we review the system error metric proposed for this work which takes into account the possible detection of the PT and **unknown-activity** classes. Finally, we describe the main features of the smartphone application which has been developed for the online recognition of activities.

7.4.1 The HAR Dataset with Postural Transitions

From the \mathcal{D}_{3T} dataset, relevant information regarding the duration of PTs against the other activities was extracted. During the experiments, every patient performed twice each PT producing a total of 60 labels for each PT. This comprises in time 9% of the entire recorded experimental data. Figure 7.4 depicts the average duration of the 6 PTs. Overall, they all have a duration of $3.73s \pm 1.17$ seconds but it can be clearly noticed that their duration is different for each PT. Even inverse transitions (e.g. LiSi and SiLi) have different durations. Some PTs, such as StLi which has the longest average duration (4.9s), are actually a sequence of other two (StSi and SiLi) as it can be observed from the experimental video data. These figures were useful for defining conditions that allow the filtering of transitions that are described in Section 7.3.3.

In the particular case of SiSt and StSi PTs, the duration results are slightly different to

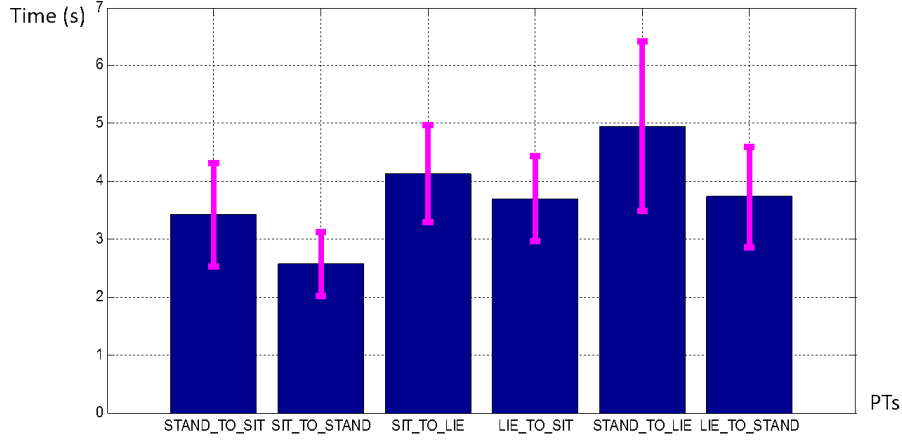


Figure 7.4: Transition times in seconds for the 6 studied PTs

the ones found in [Najafi et al., 2002] where the transition duration between these two PTs was not very significant and supported their finding that the type of transition did not have a specific role for the fall risk evaluation. Although the mean duration of these two transitions all together in our experiment is very close to their estimations in low fall-risk patients (2.98s for us against 2.92s for them), we found that StSi takes in average considerably more time than SiSt. In comparison to their approach, we selected a higher number of participants which seems to provide a more representative statistical measurement.

On the other hand, the rest of activities (BAs) took instead longer times ($17.3s \pm 5.7$ in average). This is important given that, in general, the execution time of activities in real life takes longer than PTs which have nearly-fixed duration. This element is also fundamental for the filtering of activities of our system.

7.4.2 Online System Error Estimation and Performance Evaluation

The method for the evaluation of the online system error requires some modifications given that PTs are now taken into account. Table 7.1 explains our error assessment method during the occurrence of PTs and BAs. It defines the conditions when a prediction is considered correct or not. It also takes into account the detection of the `unknown-activity`.

From these conditions we have developed an error metric to evaluate the system performance as a function of the predicted activity \hat{c}_t and the ground truth targets of the labeled data in the vicinity of time t : y_{t-1} , y_t , and y_{t+1} . It is valid for PTA-6A and PTA-7A methods, and it has

Ground-Truth	Prediction	Error Evaluation
Basic Activities		
A1 - A1 - A1	A1 - A1 - A1	Correct
A1 - A1 - A1	A1 - A2 - A1	Incorrect
A1 - A1 - A1	A1 - UA - A1	Incorrect
A1 - A1 - A1	A1 - PT - A1	Incorrect ¹
Postural Transitions		
A1 - PT - A2	A1 - A1 \vee A2 - A2	Correct
A1 - PT - A2	A1 - A3 - A2	Incorrect
A1 - PT - A2	A1 - UA - A2	Correct
A1 - PT - A2	A1 - PT - A2	Correct ¹

Table 7.1: Classification error assessment conditions for BAs and PTs. A = Activity, U = Unknown. ¹ Only applicable to the PTA-7A method

the following formulation:

$$\epsilon(\hat{c}_t, y_{t-1}, y, y_{t+1}) \begin{cases} 0 & \text{if } \begin{cases} y_t = \hat{c}_t \vee \\ (y_t = \text{PT} \wedge y_{t-1} \neq y_{t+1} \wedge (\hat{c}_t = y_{t-1} \vee \hat{c}_t = y_{t+1})) \vee \\ (y_t = \text{PT} \wedge \hat{c}_t = \text{UA}) \end{cases} \\ 1 & \text{otherwise,} \end{cases} \quad (7.3)$$

Notice that the error function penalizes the detection of either **unknown-activity** or **PT** during the occurrence of **BAs** as on this dataset we expect them to occur only during transitions. Consequently, any aim to reduce the error in **PTs** can reduce the overall performance as it can have unfavorable effects in the predictions of **BAs**.

Moreover, we make use of the confusion matrix for evaluating the system classification performance. As we presented in Section 2.5.4.1, in a typical confusion matrix $\mathcal{C} \in \mathbb{R}^{m \times m}$, the number of actual classes matches the number of predicted classes. For the evaluation of our methods, however, we slightly modify this idea because the number of classes of our ground truth (7 classes: 6 **BAs** plus the **PT** class) is different to the number of possible outputs of the **HAR** system: the **unknown-activity** is incorporated. This produces a confusion matrix $\mathcal{C}' \in \mathbb{R}^{m \times m+1}$.

To make the \mathcal{C}' still informative, we preserve in its diagonal the correct classifications, and the misclassifications outside from it except on its last column $\mathcal{C}'_{(:,c+1)}$ which corresponds to the predicted unknown activities. This column is instead used to allocate the correct predictions, based on the proposed error metric, of the **unknown-activity** class that occur during **PTs** so they do not appear as *misclassifications* outside the diagonal. In a similar way, we assign as true positives of the **PT** class the samples predicted during **PTs** that match the ground-truth of any of the neighboring window samples, so they do not appear outside the diagonal either.



Figure 7.5: HARApp smartphone user interface

7.4.3 HARApp: The Android App for HAR

We developed **HARApp**, a smartphone application based on the **PTA-HAR** system. An Android OS equipped smartphone was selected for this task (**SGSII**). The code for the application user interface was written in Java and the most expensive tasks such as signal processing, ML algorithm and activity filtering were written in C for allowing faster performance. The **NDK** facilitated to embed these native code components into the application.

The app was structured according to Algorithm 3. Two separate threads process the main functions: *ProcessInertialSignals()* periodically receives from the OS the inertial signals for conditioning which get stored in a circular buffer. In parallel, the execution of the *OnlinePrediction()* function controls the prediction of activities which is triggered by scheduled interruptions every 1.28 sec (half window sample). The duration of a complete cycle of this function, from window sampling to activity estimation, takes in average about 152 ms for the **PTA-6A** method and 162 ms for the **PTA-7A** using a **SGSII** smartphone. These times are similar as they share the same feature extraction process which takes nearly 92% of the processing time. The remaining time is dedicated to the **SVM**, which only varies in the number of predicted classes per method, and the filtering stages. The app consumes around 6 MB of memory and 4.2% of the CPU available time.

The multiclass **SVM** prediction is performed in real-time although the model parameters (\mathbf{w}_c and b_c) are learned offline and loaded into the app beforehand. Furthermore, the online prediction can be visualized through the touchscreen of the smartphone. Although this is rarely possible during operation as the device is located on the waist which restricts this visualization. Figure 7.5 shows a screenshot of the **HARApp** graphical interface. A log file also records the predicted activities and their associated timestamps for subsequent analysis. Moreover, a communications interface allows to access live prediction data from any other device through a Wi-Fi connection.

7.5 Results

In this section we present the experimental results obtained for the two HAR methods. For this, we make use of the error metric defined in section Section 7.4. We have divided this evaluation first by giving a global overview of the methods performance and then we go further into detail with the analysis of the activities through the use of confusion matrices and measures of specificity and sensitivity.

7.5.1 System Error Evaluation

To evaluate the performance of the proposed methods, we took our previous L-HAR system presented in chapter Chapter 6 as a reference point. This is equivalent to PTA-HAR system with the PTA-6A method without applying temporal activity filtering. We first learned the \mathcal{D}_{2T} dataset which only considered the 6 BAs and was generated using \mathcal{W}_1 sampling (Section 4.3.2). A system error of 3.39% was achieved (Section 6.4.4). Then, we applied the same procedure to \mathcal{D}_{3T} , which contains PT labels and \mathcal{W}_2 window sampling, and obtained an recognition error of 7.72% using the proposed error metric. This showed an increase of the system error by 4.33% percentage points mainly due to the misclassifications that occurred in PTs. This finding showed how the first offline approach fails to work online when it is under a large number of transitory events between activities such as PTs. In this dataset they cover nearly 9% of the available data. Although this is a rather small portion, it is influential in the overall system performance.

Henceforth, we consider separately the effect of PTs and BAs in the system and work only on the \mathcal{D}_{3T} dataset. We assess the two proposed methods (PTA-6A and PTA-7A). This evaluation is presented in Table 7.2 where we include the overall system error and also the error of intermediate stages of the processing. In this way, it is possible to have an idea how the different stages of the algorithm are progressively affecting the overall classification performance. The three stages are: no filtering (SVM output), probability filtering, and discrete filtering. On the table, every row represents these stages of the algorithm and the columns each HAR method.

From the table, it can be also noticed that the error in the PTA-6A method without filtering is the highest achieved (7.72%). As we decompose this, we can see that this is mainly due to a large error of 41.34% in the classification of PTs. BAs error instead remains much lower with a 4.46%. We can also observe that the temporal activity filters widely improve the classification of PTs reaching a minimum error of 5.77%. BAs instead improve only slightly after filtering. The final error of the PTA-6A method is 3.34% which nearly matches the one achieved with L-HAR and \mathcal{D}_{2T} that did not take into account PTs.

Alternatively, the PTA-7A method presents a different behavior. As PTs are learned, its recognition error is much lower from the first beginning when they are classified by the SVM (2.12%) instead of the value obtained with the previous method which was much larger (41.34%). However, the classification error of BAs is always slightly higher when compared to the previous method. Primarily because the addition of the extra class in the learning stage causes some

Filter	BAs	PTs	Overall
PTA-6A			
No Filtering	4.46%	41.34%	7.72%
Probability	3.45%	18.24%	4.76%
Discrete	3.10%	5.77%	3.34%
PTA-7A			
No Filtering	4.60%	2.12%	4.39%
Probability	3.63%	0.61%	3.36%
Discrete	3.51%	0.61%	3.25%

Table 7.2: System error based on filtering stage and type of activity

BAs to get misclassified as PTs as it would be expected. The temporal filtering does help to improve all BAs but its effect is minor. This method is showing that the learning of postural transitions can be helpful to the classification of activities. The final error of both methods is similar, being slightly lower for the PTA-7A by only 0.09%.

7.5.2 Activity Classification Performance

The confusion matrices for the first HAR method PTA-6A are shown in Table 7.3. They depict the classification results of the system before and after the activity temporal filters. The first noticeable difference between them is the matrix size as the `unknown-activity` appears only after filtering. Additionally, it is evident that the number of false negatives for the PT class is quite large before filtering. In particular, dynamic activities such as `walking-upstairs` provide most of these misclassifications indicating that the system was incorrectly predicting them during the occurrence of PTs. Therefore the temporal activity filters have helped to minimize this error. It is also noticeable the reduction of interclass misclassifications between similar activities such as in the static postures `sitting` and `standing`, and also between `walking` and `walking-upstairs`. After filtering, the false negatives of the `standing` class produced by `sitting` samples become nearly zero (from 18 to 1), however, the opposite case, which has a reduction of 26% in the number of misclassification, still preserve some errors.

Table 7.4 shows the results for the PTA-7A method. The addition of the PT class into the learned SVM model shows how most of the PT samples are correctly classified even before applying temporal activity filtering as opposed to the previous method which had a large number of false negatives of the PT class. Moreover, after filtering, small improvements are still evident: for instance, the number of false negatives for the PT class is further reduced (from 7 to 2), and also the interclass misclassification of static postures is also diminished. The SVM output is, however, generating some false positives of the PT class. This means that some actual BAs are being confused with PTs. Therefore, the addition of an additional activity (PT) into the model is producing an increase on the BAs error which can be unfavorable in applications where the occurrence of BAs is larger than PTs. This explains why the BAs error with the PTA-7A method is higher than with the PTA-6A. Notice also that in our model, transitions are considered as a single class but for some other applications it might be needed to learn them separately. This

can decrease even further the recognition performance of BAs as the number of classes of the classifier increases.

Method	PTA-6A							
Activity	WK	WU	WD	SI	ST	LD	PT	
WK	542	0	3	1	0	0	0	
WU	32	523	2	0	2	0	0	
WD	3	4	498	0	4	0	0	
SI	0	4	0	481	71	1	0	
ST	3	3	0	18	588	0	0	
LD	0	0	0	0	0	604	0	
PT	12	101	1	18	4	0	193	

Method	PTA-6A							
Activity	WK	WU	WD	SI	ST	LD	PT	UA
WK	545	0	0	0	0	0	0	1
WU	28	514	1	0	1	0	0	15
WD	0	0	505	0	3	0	0	1
SI	0	0	0	504	52	0	0	1
ST	0	1	0	1	610	0	0	0
LD	0	0	0	0	0	604	0	0
PT	0	10	0	9	0	0	310	0

Table 7.3: PTA-6A Confusion Matrices. Before and after filtering

Method	PTA-6A							
Activity	WK	WU	WD	SI	ST	LD	PT	
WK	539	1	3	2	0	0	1	
WU	28	513	2	1	1	0	14	
WD	2	5	498	0	4	0	0	
SI	0	3	0	486	68	0	0	
ST	1	0	0	19	591	0	1	
LD	0	0	0	0	0	604	0	
PT	3	2	0	2	0	0	322	

Method	PTA-6A							
Activity	WK	WU	WD	SI	ST	LD	PT	UA
WK	542	1	0	1	0	0	1	1
WU	26	510	1	0	1	0	16	5
WD	0	0	506	0	3	0	0	0
SI	0	0	0	499	57	0	1	0
ST	0	0	0	3	607	0	2	0
LD	0	0	0	0	0	604	0	0
PT	0	0	0	2	0	0	327	0

Table 7.4: PTA-7A Confusion Matrix. Before and after filtering

The sensitivity and specificity measures for each activity were also estimated and presented in table Table 7.5. In this way, we can analyze separately how the actual positives and negatives are correctly classified. For the 6 BAs we obtained a high specificity, also something similar occur with the classes sensitivity except for the classes *walking-upstairs* and *sitting* which have a smaller value. They are being mostly misclassified as *walking* and *standing* respectively which are similar activities. It is also visible that the detection of PTs is working well as the sensitivity

Activity	Sensitivity		Specificity	
	PTA-6A	PTA-7A	PTA-6A	PTA-7A
WK	99.82%	99.27%	99.11%	99.18%
WU	91.95%	91.23%	99.65%	99.97%
WD	99.21%	99.41%	99.97%	99.97%
SI	90.48%	89.58%	99.68%	99.81%
ST	99.67%	99.18%	99.67%	98.03%
LD	100.0%	100.0%	100.0%	100.0%
PT	-	99.39%	-	99.41%

Table 7.5: Sensitivity and Specificity

and sensitivity are above 99%. This is only applicable to the **PTA-7A** as in the other method PTs are not learned.

7.6 Summary

In this chapter, we presented a fully operational **HAR** system for the recognition of activities using smartphones (**PTA-HAR**). We showed through two different methods how to achieve online recognition while taking into account the effect of postural transitions in the overall system classification performance. Although, these transitory events are usually disregarded in most applications, they become relevant when their incidence is high (e.g. sports activities and housekeeping), as well as the need to explore and evaluate them.

The two proposed methods (**PTA-6A** and **PTA-7A**) exploit correlation between contiguous activities to reduce misclassifications, in particular during transitory events. This is achieved through the filtering of activities which are interpreted as probability signals that change over time.

Results have shown that the two proposed methods of the **PTA-HAR** system have similar classification performance and are therefore suitable options for **HAR** applications. They have also confirmed to be more appropriate for online classification than **L-HAR** which does not consider activity temporal filtering. Moreover, here we provide some considerations in order to guide the selection of the most suitable method depending on its application:

- When the precise detection of PTs is required, the **PTA-7A** approach is the only one that can be applied, though most **HAR** applications are only concerned about the detection of BAs. The **PTA-6A** method instead avoids learning these events but still prevents problems that could arise in the presence of PTs during classification.
- The method selection is also a trade-off between having a simpler learning algorithm and system accuracy. The **PTA-6A** method is in particular easier to implement as the learning stage does not require PTs. For example, in applications with a larger number of activities (e.g. by adding activities such as bent, reclined, lying down facing up/down, etc.), the recording of transitions between basic activities becomes more complex as the number of

possible PTs increases quadratically in a proportion of $v(v - 1)$, where v is the number of studied BAs ($v = 3$ SPs in our case). This, therefore affects the learning on the PTA-7A method.

- The method selection can be also associated with the number of PTs that occur with respect to the time extent of other activities. If they do not occur too often or the time between PTs is rather large, then we can be less rigorous about learning transitions and use the PTA-6A approach.
- Some datasets only include information about BAs and do not have transitions labeled. This limits the study to use only the PTA-6A method.

The adaptation of the **unknown-activity** class into the system was useful for the performance of the PTA-6A approach as PTs were detected as unfamiliar events. Furthermore, this concept can be extended to real life applications such as in the monitoring of activities where there are chances to perform different activities that are not known in advance. It is preferable to have a system that notifies that an activity seems unknown rather than always classifying it only as one of the studied group of activities. Furthermore, the repeated detection of this unknown events can be also an indication that the system is not working correctly. For example, if the sensor is not located in the right position or if the activities performed by a new user seem not to be recognized properly, etc. This idea should be further studied and it is proposed as a future direction of research.

This work explored probability estimates as a measure of how likely an activity is to be carried out. This has shown to be useful for performing filtering techniques when it is also combined with the idea of interrelationship between different activities. Henceforth, it would be interesting to explore novel approaches to make the recognition of activities more robust by applying probabilistic approaches such as Markov chains on smartphones. These can, for example, define the PTs as transitions between states (BAs) and use activity probability estimates as the observations or input to the system.

Chapter 8

Conclusions

8.1 Achievements

In this thesis we presented a collection of contributions related to the recognition of human activities in the [AmI](#) framework. We exploited existing commercial hardware (smartphones) and state-of-the-art [ML](#) algorithms ([SVMs](#)) in order to contribute with the design of human-centered services that improve people’s [QoL](#). In particular, here we summarize our most relevant achievements:

- We developed a [HAR](#) system for the real-time classification of [BAs](#) using a single waist-mounted smartphone device ([PTA-HAR](#)). It can provide activity information to other context-aware applications within the same device or externally located through wireless communications (e.g. for daily monitoring systems for the elderly and physical activity trackers for athletes). The system also handles [PTs](#) through learning and filtering in order to improve the classification performance during transitions between [BAs](#). Human body motion signals are continuously captured through the device’s accelerometer and gyroscope. These signals are then processed and segmented into windows to extract relevant activity features in the time and frequency domain. The features become the input of an [ML](#) algorithm: a multiclass linear [SVM](#) ([MC-L1-SVM](#)) which allows to make activity predictions. Finally, consecutive predictions in time are post-processed in order to minimize classification errors through a filtering module ([Table 7.2](#)) which is aware of [PTs](#) and considers the relationship between contiguous activities in real-life.
- We proposed a novel hardware-friendly [SVM](#) intended to predict activities using only fixed-point arithmetic ([HF-SVM](#)) to be applied to mobile devices for lower power consumption or hardware without floating-point units (e.g. low-cost disposable wearable sensors). We found that the classification of the selected [BAs](#) allows a reduction of the number representation of the sensor data (e.g. from 32 to 6 bits) without substantially damaging the system performance ([Figure 5.3](#)). In this way, we can control over model complexity and accuracy in order to improve recognition speed and reduce battery consumption against

typical floating-point based SVM formulations.

- We presented a multiclass linear SVM algorithm (MC-L1-L2-SVM) to be applied in smartphone-based HAR. Its advantages against non-linear approaches include its faster prediction ability, memory-reduced learned model, and distinctive embedded mechanism of performing feature selection that discards irrelevant or noisy features during the training process (Table 6.7). The method also allows to trade-off dimensionality reduction against classification accuracy. Moreover, we provided along with this algorithm a novel and flexible training approach (EX-SMO) that only requires any of the widely known QP solvers available such as SMO for its implementation.

8.2 Future Work

There is still room for improvement in our work which can be addressed from two different perspectives: i) by solving current limitations of our proposed systems, and ii) by extending our achievements through complementary and novel applications. In the first case, some issues have arisen such as the limited number of activities the system can deal with, the fixed smartphone position on the waist, and the adoption of novel approaches to deal users with distinct differences in their motion patterns (e.g. people with walking difficulties) into the system. On the second case, new ideas about how to exploit HAR information in order to provide new services can be explored. These include the development of context-aware apps for health and sports monitoring, elderly care and understanding interaction between users using similar systems. In this section we focus on some of these aspects and propose them as future research directions:

- For a new user, the performance of the proposed HAR system can be improved if his motion data is integrated into the learned model. Although this can be done, for instance, through retraining after following a controlled sequence of activities, this process can be tedious for the user. But considering that during a normal day it is possible to gather large amounts of data, we can explore *semi-supervised learning* strategies which can allow to combine this unlabeled data with already existing labeled trained data in order to produce considerable improvements in the system learning accuracy. This can bring advantages to new users, specially those with particular conditions such as very slow motion or physical disabilities which are normally difficult to incorporate in the training and the ML algorithm generalization capability is not sufficient to include them.
- The proposed HAR systems are specific to be used with waist-mounted smartphones. Although this position allows some degree of variability, it is important to explore if it is also possible to place the device in different body parts such as shirt or pants pockets and even worn around the arms (e.g. with armbands for running). In the first case some problems may arise due to the free and continuous motion of the devices with respect to the body position which can be hard to control (e.g. for distinguishing between standing

and sitting activities). Moreover, now that its is becoming increasingly popular the use of wearable devices such as smartwatches with embedded inertial sensors, it is interesting to investigate how to combine them with our current smartphone-based system in order to improve the recognition or to add new activities that involve upper limbs such as typing, brushing teeth and writing.

- The outcome of the presented HAR system can be used in higher-level context-aware applications. For example, when combined with location-based services such as GPS or home presence sensors for achieving indoor and outdoor activity detection. Also, if activity information is merged with vital sign sensors, it is possible develop apps for medical diagnostic and monitoring of ill patients during their daily life without third-party supervision. Likewise, it is possible to merge activity information with smartphone connectivity status such as incoming calls/messages, in order to control the smartphone behavior when some specific activities are occurring, e.g. to avoid receiving phone calls while users are running or sleeping, and to automatically produce unavailability responses. Later, when users activity has changed, the information can be provided in a timely manner. On the other hand, we also propose the study of the interaction between two or more users wearing the HAR recognition system in order to infer collective behavior such as chatting, dancing, playing sports, etc. In conclusion, there is a wide range of new possibilities and services that need to be explored where HAR can contribute in their development.

Bibliography

- Emile Aarts and Reiner Wichert. Ambient intelligence. In *Technology Guide*, 2009.
- Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7:29–58, 2000.
- A Aizerman, Emmanuel M Braverman, and LI Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.
- Felicity R Allen, Eliathamby Ambikairajah, Nigel H Lovell, and Branko G Celler. Classification of a known sequence of motions and postures from accelerometry data using adapted gaussian mixture models. *Physiological Measurement*, 27:935, 2006.
- Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46:175–185, 1992.
- Kerem Altun and Billur Barshan. Human activity recognition using inertial/magnetic sensor units. In *Human Behavior Understanding*, 2010.
- Oliver Amft, Clemens Lombriser, Thomas Stiefmeier, and Gerhard Tröster. Recognition of user activity sequences using distributed event detection. In *European conference on Smart sensing and context*, 2007.
- Android, 2013. Android developers. <http://developer.android.com/index.html>. Accessed: 05/11/2013.
- D. Anguita, A. Ghio, S. Pischiutta, and S. Ridella. A hardware-friendly support vector machine for embedded automotive applications. In *International Joint Conference on Neural Networks*, 2007.
- D. Anguita, A. Ghio, S. Ridella, and D. Sterpi. K-fold cross validation for error rate estimate in support vector machines. In *International Conference on Data Mining*, 2009.
- D. Anguita, A. Ghio, L. Oneto, and S. Ridella. Selecting the hypothesis space for improving the generalization ability of support vector machines. In *International Joint Conference on Neural Networks*, 2011.
- D. Anguita, A. Ghio, L. Oneto, and S. Ridella. In-sample and out-of-sample model selection and error estimation for support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 23:1390–1406, 2012a.

- D. Anguita, A. Ghio, L. Oneto, and S. Ridella. A learning machine with a bit-based hypothesis space. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013a.
- Davide Anguita and Dario Sterpi. Nature inspiration for support vector machines. In *International conference on Knowledge-Based Intelligent Information and Engineering Systems*, 2006.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient Assisted Living and Home Care*, 2012b.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. Human activity recognition on smartphones for mobile context awareness. In *Neural Information Processing Systems. Workshop on Machine Learning Approaches to Mobile Context Awareness*, 2012c.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic. *Journal of Universal Computer Science*, 19:1295–1314, 2013b.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013c.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge-Luis Reyes-Ortiz. Training computationally efficient smartphone-based human activity recognition models. In *International Conference on Artificial Neural Networks*, 2013d.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Jorge-Luis Reyes-Ortiz, and Sandro Ridella. A novel procedure for training l1-l2 support vector machine classifiers. In *International Conference on Artificial Neural Networks*, 2013e.
- L. Atallah, B. Lo, R. King, and Guang-Zhong Yang. Sensor placement for activity detection using wearable accelerometers. In *International Conference on Body Sensor Networks*, 2010.
- Michaël Aupetit. Nearly homogeneous multi-partitioning with a deterministic generator. *Neurocomputing*, 72:1379–1389, 2009.
- Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *International Conference on Architecture of Computing Systems*, 2010.
- K. Bache and M. Lichman. UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2013.
- M Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and G Troster. Wearable assistant for parkinson’s disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14:436–446, 2010.
- G. Bahle, P. Lukowicz, K. Kunze, and K. Kise. I see you: How to improve wearable activity recognition by leveraging information from environmental cameras. In *IEEE International Conference on Pervasive Computing and Communications Workshops*, 2013.

- L. Bao and S.S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*, 2004.
- P.L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.
- P.L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33:1497–1537, 2005.
- C. BenAbdelkader, R. Cutler, and L. Davis. Stride and cadence as a biometric in automatic person identification and verification. In *International Conference on Automatic Face and Gesture Recognition*, 2002.
- M. Berchtold, M. Budde, D. Gordon, H.R. Schmidtke, and M. Beigl. Activity recognition service for mobile phones. In *International Symposium on Wearable Computers*, 2010.
- Xuehai Bian, Gregory D Abowd, and James M Rehg. Using sound source localization in a home environment. In *Pervasive Computing*, 2005.
- Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Annual workshop on Computational learning theory*, 1992.
- Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University, 2004.
- Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
- T. Brezmes, J.L. Gorricho, and J. Cotrina. Activity recognition from accelerometer data on a mobile phone. In *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, 2009.
- Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa, Tullio Vernazza, and Renato Zaccaria. Human motion modelling and recognition: A computational approach. In *IEEE International Conference on Automation Science and Engineering*, 2012.
- Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa, Tullio Vernazza, and Renato Zaccaria. Analysis of human behavior recognition algorithms based on acceleration data. In *IEEE International Conference on Robotics and Automation*, 2013.
- Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 46:33, 2014.
- Andrew Campbell and Tanzeem Choudhury. From smart to cognitive phones. *IEEE Pervasive Computing*, 11:7–11, 2012.
- A.T. Campbell, S.B. Eisenman, N.D. Lane, E. Miluzzo, R.A. Peterson, Hong Lu, Xiao Zheng, M. Musolesi, K. Fodor, and Gahng-Seop Ahn. The rise of people-centric sensing. *IEEE Internet Computing*, 12: 12–21, 2008.
- Claudette Cedras and Mubarak Shah. Motion-based recognition a survey. *Image and Vision Computing*, 13:129–155, 1995.

- Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27–54, 2011.
- Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. *Semi-supervised learning*. MIT press Cambridge, 2006.
- Liming Chen, J. Hoey, C.D. Nugent, D.J. Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42:790–808, 2012a.
- Liming Chen, C.D. Nugent, and Hui Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 24:961–974, 2012b.
- T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. Legrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J.A. Landay, J. Lester, D. Wyatt, and D. Haehnel. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7:32–41, 2008.
- Brian Coley, Bijan Najafi, Anisoara Paraschiv-Ionescu, and Kamiar Aminian. Stair climbing detection during daily physical activity using a miniature gyroscope. *Gait & posture*, 22:287–294, 2005.
- Diane J Cook and Sajal K Das. How smart are our environments? an updated look at the state of the art. *Pervasive and Mobile Computing*, 3:53–73, 2007.
- Diane J. Cook and Sajal K. Das. Pervasive computing at scale: Transforming the state of the art. *Pervasive and Mobile Computing*, 8:22–35, 2012.
- C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.
- Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- S. Dernbach, B. Das, N.C. Krishnan, B.L. Thomas, and D.J. Cook. Simple and complex activity recognition through smart phones. In *International Conference on Intelligent Environments*, 2012.
- Kenji Doya. Reinforcement learning in continuous time and space. *Neural computation*, 12:219–245, 2000.
- Harris Drucker, Chris JC Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.
- Pierre Duhamel and Martin Vetterli. Fast fourier transforms: a tutorial review and a state of the art. *Signal processing*, 19:259–299, 1990.
- Jessica Ekholm and Sylvain Fabre. Forecast: Mobile data traffic and revenue, worldwide, 2010-2015. In *Gartner Mobile Communications Worldwide*, 2011.
- M.G. Epitropakis, V.P. Plagianakos, and M.N. Vrahatis. Hardware-friendly higher-order neural network training using distributed evolutionary algorithms. *Applied Soft Computing*, 10:398 – 408, 2010.
- Miikka Hermes, Juha Parkka, and Luc Cluitmans. Advancing from offline to online activity recognition with wearable sensors. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008.

- Eur, 2011. The 2012 ageing report: Underlying assumptions and projection methodologies. European Commission. Directorate-General for Economic and Financial Affairs, 2011.
- R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- Rob Fergus, Yair Weiss, and Antonio Torralba. Semi-supervised learning in gigantic image collections. In *Neural Information Processing Systems*, 2009.
- Brian P Flannery, Wiliam H Press, Saul A Teukolsky, and William Vetterling. *Numerical recipes in C*. Cambridge University Press, 1992.
- Vojtech Franc, Alexander Zien, and Bernhard Schölkopf. Support vector machines as probabilistic models. In *International Conference on Machine Learning*, 2011.
- J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1, 2010.
- A. Ganapathiraju, J.E. Hamaker, and J. Picone. Applications of support vector machines to speech recognition. *IEEE Transactions on Signal Processing*, 52:2348 – 2355, 2004.
- M Gandetto, L Marchesotti, S Sciutto, D Negroni, and CS Regazzoni. From multi-sensor surveillance towards smart interactive spaces. In *IEEE International Conference on Multimedia and Expo*, 2003.
- Juan Pablo García-Vázquez, Marcela D. Rodríguez, Mónica E. Tentori, Diana Saldaña, Ángel G. Andrade, and Adán N. Espinoza. An agent-based architecture for developing activity-aware systems for assisting elderly. *Journal of Universal Computer Science*, 16:1500–1520, 2010.
- R. Genov and G. Cauwenberghs. Kerneltron: support vector machine in silicon. *IEEE Transactions on Neural Networks*, 14:1426 – 1434, 2003.
- Jose George, Kathleen Vunckx, Sabine Tejpar, ChristopheM. Deroose, Johan Nuyts, Dirk Loeckx, and Paul Suetens. Fuzzy statistical unsupervised learning based total lesion metabolic activity estimation in positron emission tomography images. In *Machine Learning in Medical Imaging*, 2011.
- A. Ghio, D. Anguita, L. Oneto, S. Ridella, and C. Schatten. Nested sequential minimal optimization for support vector machines. In *International Conference on Artificial Neural Networks*, 2012.
- Anshul Gupta, Carolina Milanesi, Roberta Cozza, and CK Lu. Market share analysis: Mobile phones, worldwide, 2q13. Technical report, Gartner, Inc., 2013.
- Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11:10–18, 2009.
- Zhenyu He and Lianwen Jin. Activity recognition from acceleration data based on discrete cosine transform and svm. In *IEEE International Conference on Systems, Man and Cybernetics*, 2009.
- R. Herbrich and R.C. Williamson. Algorithmic luckiness. *Journal of Machine Learning Research*, 3: 175–212, 2003.

- Simon Herrlich, Sven Spieth, Rachid Nouna, Roland Zengerle, Libero I. Giannola, Diego-Esteban Pardo-Ayala, Eugenio Federico, and Pierangelo Garino. Ambulatory treatment and telemonitoring of patients with parkinsons disease. In *Ambient Assisted Living*, 2011.
- Joyce Joyce Carmen Ho. *Interruptions: using activity transitions to trigger proactive messages*. PhD thesis, Massachusetts Institute of Technology, 2004.
- Jerry Honeycutt. *Introducing Windows 8: An Overview for IT Professionals*. Microsoft Press, 2012.
- Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13:415–425, 2002.
- K. Irick, M. DeBole, V. Narayanan, and A. Gayasen. A hardware efficient support vector machine architecture for fpga. In *International Symposium on Field-Programmable Custom Computing Machines*, 2008.
- Valérie Issarny, Daniele Sacchetti, Ferda Tartanoglu, Françoise Sailhan, Rafik Chibout, Nicole Levy, and Angel Talamona. Developing ambient intelligence systems: A solution based on web services. *Automated Software Engineering*, 12:101–137, 2005.
- Luciana C. Jatoba, Ulrich Grossmann, Chistophe Kunze, Jorg Ottenbacher, and Wilhelm Stork. Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In *International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008.
- Rui José, Helena Rodrigues, and Nuno Otero. Ambient intelligence: Beyond the inspiring vision. *Journal of Universal Computer Science*, 16:1480–1499, 2010.
- Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10:156–167, 2006.
- William Karush. Minima of functions of several variables with inequalities as side constraints. Master’s thesis, Department of Mathematics, University of Chicago, 1939.
- Marika Kästner, Marc Strickert, and Thomas Villmann. A sparse kernelized matrix learning vector quantization model for human activity recognition. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.
- S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to platt’s smo algorithm for svm classifier design. *Neural Computation*, 13:637–649, 2001.
- Adil Mehmood Khan, Y-K Lee, SY Lee, and T-S Kim. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *IEEE International Conference on Future Information Technology*, 2010a.
- Adil Mehmood Khan, Young-Koo Lee, Sungyoung Y Lee, and Tae-Seong Kim. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE transactions on information technology in biomedicine*, 14:1166–1172, 2010b.

- NaimulMefraz Khan, Riadh Ksantini, ImranShafiq Ahmad, and Ling Guan. A sparse support vector machine classifier with nonparametric discriminants. In *International Conference on Artificial Neural Networks*, 2012.
- Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. Ambient intelligence in assisted living: enable elderly people to handle future interfaces. In *Universal access in human-computer interaction. Ambient interaction*, 2007.
- V. Koltchinskii. Local rademacher complexities and oracle inequalities in risk minimization. *Annals of Statistics*, 34:2593–2656, 2006.
- Mustafa Kose, Ozlem Durmaz Incel, and Cem Ersoy. Online human activity recognition on smart phones. In *Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*, 2012.
- HW Kuhn, AW Tucker, et al. Nonlinear programming. In *Berkeley Symposium on Mathematical Statistics and Probability*, 1951.
- Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explorations Newsletter*, 12:74–82, 2011.
- Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine Learning*, 59:161–205, 2005.
- Nicholas Lane, Mashfiqui Mohammod, Mu Lin, Xiaochao Yang, Hong Lu, Shahid Ali, Afsaneh Doryab, Ethan Berke, Tanzeem Choudhury, and Andrew Campbell. Bewell: A smartphone application to monitor, model and promote wellbeing. In *IEEE International ICST Conference on Pervasive Computing Technologies for Healthcare*, 2012.
- Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48:140–150, 2010.
- O. Lara and M. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys Tutorials*, 1:1–18, 2012a.
- O.D. Lara and M.A. Labrador. A mobile platform for real-time human activity recognition. In *IEEE Consumer Communications and Networking Conference*, 2012b.
- 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*, 8:717 – 729, 2012.
- Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Müller, E. Säckinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In *International Conference on Artificial Neural Networks*, 1995.
- Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1:541–551, 1989.
- Seon-Woo Lee and K. Mase. Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, 1:24–32, 2002.

- S.W. Lee, S.W. Lee, and H.C. Jung. Real-time implementation of face recognition algorithms on dsp chip. In *Audio-and Video-Based Biometric Person Authentication*, 2003.
- B. Lesser, M. Mücke, and W.N. Gansterer. Effects of reduced precision on floating-point svm classification accuracy. *Procedia Computer Science*, 4:508–517, 2011.
- Qiang Li, J.A. Stankovic, M.A. Hanson, A.T. Barth, J. Lach, and Gang Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Wearable and Implantable Body Sensor Networks*, 2009.
- Weiyao Lin, Ming-Ting Sun, R. Poovandran, and Zhengyou Zhang. Human activity recognition for video surveillance. In *IEEE International Symposium on Circuits and Systems*, 2008.
- Chang Liu, Qing Zhu, Kenneth A. Holroyd, and Elizabeth K. Seng. Status and trends of mobile-health applications for ios devices: A developer’s perspective. *Journal of Systems and Software*, 84:2022 – 2033, 2011.
- Stephen Ronald Lord. *Falls in older people: risk factors and strategies for prevention*. Cambridge University Press, 2007.
- NH Lovell, Ning Wang, E Ambikairajah, and Branko George Celler. Accelerometry based classification of walking patterns using time-frequency analysis. In *IEEE Annual International Conference of the Engineering in Medicine and Biology Society*, 2007.
- Paul Lukowicz, Jamie Ward, Holger Junker, Mathias Stäger, Gerhard Tröster, Amin Atrash, and Thad Starner. Recognizing workshop activity using body worn microphones and accelerometers. In *Pervasive Computing*, 2004.
- Andrea Mannini and Angelo Maria Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10:1154–1175, 2010.
- J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In *IEEE International Conference on Systems, Man, and Cybernetics*, 2001.
- Uwe Maurer, Asim Smailagic, Daniel P Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *IEEE International Workshop on Wearable and Implantable Body Sensor Networks*, 2006.
- Sabato Mellone, Carlo Tacconi, and Lorenzo Chiari. Validity of a smartphone-based instrumented timed up and go. *Gait & Posture*, 36:163 – 165, 2012.
- Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
- B. Najafi, K. Aminian, F. Loew, Y. Blanc, and P.A. Robert. Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly. *IEEE Transactions on Biomedical Engineering*, 49:843–851, 2002.
- Bijan Najafi, Kamiar Aminian, Anisoara Paraschiv-Ionescu, François Loew, Christophe J Bula, and Philippe Robert. Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *IEEE Transactions on Biomedical Engineering*, 50:711–723, 2003.

- H. Neven, V.S. Denchev, G. Rose, and W.G. Macready. Training a large scale classifier with the quantum adiabatic algorithm. In *arXiv preprint arXiv:0912.0779*, 2009.
- Hartmut Neven, Vasil S Denchev, Geordie Rose, and William G Macready. Training a binary classifier with the quantum adiabatic algorithm. In *arXiv preprint arXiv:0811.0416*, 2008.
- Ben Nham, Kanya Siangliulue, and Serena Yeung. Predicting mode of transport from iphone accelerometer data. Technical report, Tech. report, Stanford Univ, 2008.
- M. Ogawa, R. Suzuki, S. Otake, T. Izutsu, T. Iwaya, and T. Togawa. Long term remote behavioral monitoring of elderly by using sensors installed in ordinary houses. In *International IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, 2002.
- M.W. Oliphant. The mobile phone meets the internet. *IEEE Spectrum*, 36:20–28, 1999.
- Luca Oneto and Noemi Greco. Model selection for support vector machines: Advantages and disadvantages of the machine learning theory. Master’s thesis, Department of Biophysical and Electronic Engineering, 2010.
- R. Parasuraman, T.B. Sheridan, and C.D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man and Cybernetics*, 30:286–297, 2000.
- J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Microsoft Research, 1998.
- John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.
- T Poggio, S Mukherjee, R Rifkin, A Raklin, and A Verri. b. In *Uncertainty in geometric computations*, 2002.
- R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108:4–18, 2007.
- Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28: 976–990, 2010.
- William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.
- J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.
- John Ross Quinlan. *C4. 5: Programs for Machine Learning*. Morgan kaufmann, 1993.
- Nishkam Ravi, Nikhil D, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Innovative Applications of Artificial Intelligence*, 2005.
- Ramona Rednic, Elena Gaura, John Kemp, and James Brusey. Fielded autonomous posture classification systems: design and realistic evaluation. In *ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2013.

- Attila Reiss, Gustaf Hendeby, and Didier Stricker. A competitive approach for human activity recognition on smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.
- Jorge-Luis Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human activity recognition using smartphones data set. <http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>, 2013a.
- Jorge-Luis Reyes-Ortiz, Alessandro Ghio, Davide Anguita, Xavier Parra, Joan Cabestany, and A Catal. Human activity and motion disorder recognition: Towards smarter interactive cognitive environments. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013b.
- Jorge-Luis Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Recognition of basic activities and postural transitions using smartphones data set. <http://www.har.smartlab.ws>, 2014a.
- Jorge-Luis Reyes-Ortiz, Luca Oneto, Samà Albert, Alessandro Ghio, Xavier Parra, and Davide Anguita. Human activity recognition on smartphones with awareness of basic activities and postural transitions. In *International Conference on Artificial Neural Networks*, 2014b.
- Jorge-Luis Reyes-Ortiz, Luca Oneto, Samà Albert, Alessandro Ghio, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Personal and Ubiquitous Computing*, (Waiting for Acceptance):–, 2014c.
- Daniele Riboni and Claudio Bettini. Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15:271–289, 2011.
- Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.
- Daniel Rodríguez-Martín. Sistema inercial vestibular amb capacitat de desenvolupament i implementació algorísmica. Master’s thesis, Universitat Politècnica de Catalunya, 2010.
- Daniel Rodríguez-Martín, Carlos Pérez-López, Albert Samà, Joan Cabestany, and Andreu Català. A wearable inertial measurement unit for long-term monitoring in the dependency care area. *Sensors*, 13:14079–14104, 2013a.
- Daniel Rodríguez-Martín, Albert Samà, Carlos Perez-Lopez, Andreu Català, Joan Cabestany, and Alejandro Rodriguez-Molinero. Svm-based posture identification with a single waist-located triaxial accelerometer. *Expert Systems with Applications*, 40:7203 – 7211, 2013b.
- D. Roggen, A. Calatroni, M. Rossi, T. Holleczeck, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkel, and A. Ferscha. Collecting complex activity data sets in highly rich networked sensor environments. In *International Conference on Networked Sensing Systems 2010*, 2010.
- Bernardino Romera-Paredes, Hane Aung, and Nadia Bianchi-Berthouze. A one-vs-one classifier ensemble with majority voting for activity recognition. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.

- Kari Roth, Ismo Kauppinen, Paulo AA Esquef, and Vesa Valimaki. Frequency warped burg's method for ar-modeling. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.
- Arash Salarian, Heike Russmann, François JG Vingerhoets, Pierre R Burkhard, and Kamiar Aminian. Ambulatory monitoring of physical activities in patients with parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 54:2296–2299, 2007.
- A Sama, C Perez-Lopez, J Romagosa, D Rodriguez-Martin, A Catala, J Cabestany, DA Perez-Martinez, and A Rodriguez-Molinero. Dyskinesia and motor state detection in parkinson's disease patients with a single movement sensor. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2012.
- Albert Samà. *Human movement analysis by means of accelerometers: Application to human gait and motor symptoms of Parkinsons Disease*. PhD thesis, Universitat Politècnica de Catalunya, 2013.
- Albert Sama, Diego E Pardo-Ayala, Joan Cabestany, and A Rodríguez-Molinero. Time series analysis of inertial-body signals for the extraction of dynamic properties from human gait. In *IEEE International Joint Conference on Neural Networks*, 2010.
- Albert Samà, Cecilio Angulo, Diego Pardo, Andreu Catal, and Joan Cabestany. Analyzing human gait and posture by combining feature selection and kernel methods. *Neurocomputing*, 74:2665 – 2674, 2011.
- Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *Handheld and ubiquitous computing*, 1999.
- B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
- Mark R Segal, Kam D Dahlquist, and Bruce R Conklin. Regression approaches for microarray data analysis. *Journal of Computational Biology*, 10:961–980, 2003.
- S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *International Conference on Machine Learning*, 2007.
- Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4:107–194, 2011.
- J. Shawe-Taylor and S. Sun. A review of optimization methodologies in support vector machines. *Neurocomputing*, 74:3609–3618, 2011.
- J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44:1926–1940, 1998.
- Liyanage C. De Silva, Chamin Morikawa, and Iskandar M. Petra. State of the art of smart homes. *Engineering Applications of Artificial Intelligence*, 25:1313 – 1321, 2012.
- M. Stikic, K. van Laerhoven, and B. Schiele. Exploring semi-supervised and active learning for activity recognition. In *IEEE International Symposium on Wearable Computers*, 2008.
- M. Stikic, D. Larlus, S. Ebert, and B. Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:2521–2537, 2011.

- Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Cambridge University Press, 1998.
- Boris Takač, Andreu Català, Daniel Rodríguez Martín, Nico van der Aa, Wei Chen, and Matthias Rauterberg. Position and orientation tracking in a ubiquitous monitoring system for parkinson disease patients with freezing of gait symptom. *Journal of Medical Internet Research*, 15:1, 2013.
- Emmanuel Tapia, Stephen Intille, Louis Lopez, and Kent Larson. The design of a portable kit of wireless sensors for naturalistic data collection. In *Pervasive Computing*, 2006.
- 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 *IEEE International Symposium on Wearable Computers*, 2007.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58:267–288, 1996.
- A.N. Tikhonov and V.Y. Arsenin. Solutions of ill-posed problems. *Mathematics of Computation*, 32:1320–1322, 1978.
- 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*, 18:1473–1488, 2008.
- Department of Economic United Nations. *World population prospects: The 2004 Revision: Volume I: comprehensive tables*. United Nations Publications, 2006.
- K. Van Laerhoven and O. Cakmakci. What shall we teach our pants? In *International Symposium on Wearable Computers*, 2000.
- Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, 1995.
- V.N. Vapnik. *Statistical learning theory*. Wiley-Interscience, 1998.
- Sethu Vijayakumar, Tomohiro Shibata, and Stefan Schaal. Reinforcement learning for humanoid robotics. In *Autonomous Robot*, 2003.
- Junping Wang, Quanshi Chen, and Yong Chen. Rbf kernel based support vector machine with universal approximation and its application. In *Advances in Neural Networks – ISNN*, 2004.
- Zhelong Wang, Ming Jiang, Yaohua Hu, and Hongyi Li. An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 16:691–699, 2012.
- John Wawrzynek, Krste Asanovic, Nelson Morgan, and Senior Member. The design of a neuro-microprocessor. *VLSI for Neural Networks and Artificial Intelligence*, 4:103–127, 1993.
- Mark Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36:75–84, 1993.
- Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, University of London, Department of Computer Science, 1998.

- WHO. World report on disability : Summary. Technical report, UN World Health Organization, 2011.
- D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82, 1997.
- Oliver J Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory, 2007.
- Wanmin Wu, Sanjoy Dasgupta, Ernesto E Ramirez, Carlyn Peterson, and Gregory J Norman. Classification accuracies of physical activities using smartphone motion sensors. *Journal of Medical Internet Research*, 14:105–130, 2012.
- Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In *National Conference on Artificial Intelligence*, 2005.
- Che-Chang Yang and Yeh-Liang Hsu. A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, 10:7772–7788, 2010.
- Guang-Zhong Yang and Magdi Yacoub. *Body sensor networks*. Springer, 2006.
- J.Y. Yang, J.S. Wang, and Y.P. Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern recognition letters*, 29:2213–2220, 2008.
- N. Yazdi, F. Ayazi, and K. Najafi. Micromachined inertial sensors. *Proceedings of the IEEE*, 86:1640–1659, 1998.
- Shumei Zhang, P. McCullagh, C. Nugent, and Huiru Zheng. Activity monitoring using a smart phone’s accelerometer with hierarchical classification. In *International Conference on Intelligent Environments*, 2010.
- Zhongtang Zhao, Yiqiang Chen, Junfa Liu, and Mingjie Liu. Cross-people motion activity recognition. In *International Joint Conference on Artificial Intelligence*, 2010.
- Vincent Wenchen Zheng, Derek Hao Hu, and Qiang Yang. Cross-domain activity recognition. In *International conference on Ubiquitous computing*, 2009.
- Chun Zhu and Weihua Sheng. Human daily activity recognition in robot-assisted living using multi-sensor fusion. In *IEEE International Conference on Robotics and Automation*, 2009.
- H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B*, 67:301–320, 2005.

Glossary

AA Ambulation Activity	HAR Human Activity Recognition
AAL Ambient Assisted Living	HARApp Human Activity Recognition Application
ADL Activities of Daily Living	HF-HAR Hardware-Friendly HAR System
ADT Android Development Tools	HF-SVM Hardware-Friendly SVM
AmI Ambient Intelligence	HMM Hidden Markov Models
ANN Artificial Neural Networks	
API Application Programming Interface	k-NN k-Nearest Neighbors
	KCV k -Fold Cross Validation
BA Basic Activity	KKT Karush-Kuhn-Tucker
BDM Bayesian Decision Making	
BSN Body Sensor Network	L-HAR Linear HAR System
	L1-L2-SVM L1-L2-Norm SVM
CA Complex Activity	L1-SVM L1-Norm SVM
CCQP Convex Constrained Quadratic Programming	L2-SVM L2-Norm SVM
cHMM Continuous Emissions HMM	LD lying-down
CPU Central Processing Unit	LDA Linear Discriminant Analysis
	LiSi lie-to-sit
DFT Discrete Fourier Transform	LiSt lie-to-stand
DT Decision Tree	LK-SVM SVM with Laplacian Kernel
DTW Dynamic Time Warping	LR Logistic Regression
	LSM Least Squares Method
EDGE Enhanced Data rates for GSM Evolution	LUT Look-Up-Table
ESANN European Symposium on Artificial Neural Networks	LVQ Learning Vector Quantization
EX-SMO Extended SMO	
FFP Feed-Forward Phase	MAP Maximum A Posteriori Probability
FFT Fast Fourier Transform	MC-GK-SVM MultiClass GK-SVM
FN False Negatives	MC-HF-SVM MultiClass HF-SVM
FP False Positives	MC-L1-L2-SVM MultiClass L1-L2-SVM
	MC-L1-SVM MultiClass L1-SVM
GDA Gaussian Discriminant Analysis	MC-L2-SVM MultiClass L2-SVM
GK-SVM SVM with Gaussian Kernel	MC-LK-SVM Multiclass LK-SVM
GMM Gaussian Mixture Models	MEMS Microelectromechanical Systems
GMR Gaussian Mixture Regression	ML Machine Learning
GPRS General Packet Radio Service	MLP Multilayer Perceptron
GPS Global Positioning System	MMS Multimedia Messaging Service
GSM Global System for Mobile Communications	
GUI Graphical User Interface	NB Naive Bayes
	NDK Native Development Kit
	NFC Near Field Communication

OS Operating System	SiSt sit-to-stand
OVA one-vs-all	SLT Statistical Learning Theory
OVO one-vs-one	SMA Signal Magnitude Area
PCA Principal Component Analysis	SMLP Simplex Method for Linear Programming
PD Parkinson's disease	SMO Sequential Minimal Optimization
PDA Personal Digital Assistant	SMS Short Message Service
PerComp Pervasive Computing	SP Static Posture
PET Positron Emission Tomography	SRM Structural Risk Minimization
PT Postural Transition	ST standing
PTA-6A PTA-HAR with 6 BAs	STD Standard Deviation
PTA-7A PTA-HAR with 6 BAs + 6 PTs	StLi stand-to-lie
PTA-HAR Postural Transition Aware HAR System	StSi stand-to-sit
QoL Quality of Life	SVM Support Vector Machine
QP Quadratic Programming	TN True Negatives
RBF Radial Basis Function	TP True Positives
RF Random Forest	UA unknown-activity
RL Reinforcement Learning	UbiComp Ubiquitous Computing
SDK Software Development Kit	USB Universal Serial Bus
SGSII Samsung Galaxy SII	WD walking-downstairs
SI sitting	WEKA Waikato Environment for Knowledge Analysis
SiLi sit-to-lie	WK walking
SIM Subscriber Identity Module	WU walking-upstairs