UNIVERSITÀ DEGLI STUDI DI MILANO

PhD School in

Computer Science

Computer Science Department
“Giovanni Degli Antoni”

PhD in

Computer Science
XXX Cycle

Human Activity Recognition in Smart-Home Environments for Health-Care Applications

INF/01

PhD candidate:
Gabriele CIVITARESE

Advisor:
Prof. Claudio BETTINI

Co-Advisor:
Prof. Daniele RIBONI

School Director:
Prof. Paolo BOLDI

Academic Year 2016/17

This is how you do it: you sit down at the keyboard and you put one word after another until it's done. It's that easy and that hard.

Neil Gaiman

Abstract

With a growing population of elderly people, the number of subjects at risk of cognitive disorders is rapidly increasing. Many research groups are studying pervasive solutions to continuously and unobtrusively monitor fragile subjects in their homes, reducing health-care costs and supporting the medical diagnosis. Clinicians are interested in monitoring several behavioral aspects for a wide variety of applications: early diagnosis, emergency monitoring, assessment of cognitive disorders, etcetera. Among the several behavioral aspects of interest, anomalous behaviors while performing activities of daily living (ADLs) are of great importance. Indeed, these anomalies can be indicators of serious cognitive diseases like Mild Cognitive Impairment. The recognition of such abnormal behaviors relies on robust and accurate ADLs recognition systems. Moreover, in order to enable unobtrusive and privacy-aware monitoring, environmental sensors in charge of unobtrusively capturing the interaction of the subject with the home infrastructure should be preferred.

This thesis presents several contributions on this topic. The major ones are two novel hybrid ADLs recognition algorithms. The former is supervised while the latter is unsupervised. Preliminary results, which still need to be confirmed, show that the recognition rate of the unsupervised method is comparable to the one obtained by the supervised one, with the great advantage of not requiring the acquisition of an annotated dataset. Beyond ADLs recognition, other contributions on smart sensing and anomaly recognition are presented. Regarding unobtrusive sensing, we propose a machine learning technique to detect fine-grained manipulations performed by the inhabitant on household objects instrumented with tiny accelerometer sensors. Finally, a novel rule-based framework for the recognition of fine-grained abnormal behaviors is presented. Experimental results on several datasets show the effectiveness of all the proposed techniques.

Author's Publications

This thesis is based on the following publications, which have been written during my three years of PhD.

Journals

1. Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, Rim Helaoui, *"SmartFABER: Recognizing Fine-grained Abnormal Behaviors for Early Detection of Mild Cognitive Impairment"*. Artificial Intelligence in Medicine, Elsevier, 2016.

International conferences

1. Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, Rim Helaoui, *"Fine-grained Recognition of Abnormal Behaviors for Early Detection of Mild Cognitive Impairment"*. In Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 149-154, Computer Society, 2015.

2. Daniele Riboni, Timo Sztyler, Gabriele Civitarese, Heiner Stuckenschmidt, *"Unsupervised Recognition of Interleaved Activities of Daily Living through Ontological and Probabilistic Reasoning"*. UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2016.

International workshops

1. Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, Viola Bulgari, *"From Lab to Life: Fine-grained Behavior Monitoring in the Elderly's Home"*. In Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 344-349. IEEE Computer Society, 2015.

2. Gabriele Civitarese, Zaffar Haider Janjua, Daniele Riboni, Claudio Bettini, *"Demonstration of the FABER System for Fine-grained Recognition of Abnormal Behaviors"*. In Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 199-201. IEEE Computer Society, 2015.

3. Daniele Riboni, Gabriele Civitarese, Claudio Bettini. *"Analysis of Long-term Abnormal Behaviors for Early Detection of Cognitive Decline"*. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PASTA2016: Workshop on PervAsive Technologies and care systems for sustainable Aging-in-place), IEEE, 2016.

4. Gabriele Civitarese, Stefano Belfiore, Claudio Bettini. *"Let the objects tell what you are doing"*. UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, 2016.

5. Gabriele Civitarese, Claudio Bettini. *"Monitoring Objects Manipulations to Detect Abnormal Behaviors"*. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, 2017.

6. Gabriele Civitarese. *"Behavioral Monitoring in Smart-Home Environments for Health-Care Applications"*. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, 2017.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and problem description

Thanks to recent advances in medicine and to improved quality of life, life expectancy considerably increased thus allowing people to live longer and healthier with respect to previous generations. However, in an aging world population, more citizens are exposed to many challenges due to cognitive decline, chronic age-related diseases, limitations in physical activities and so on. This scenario brings negative consequences on the ability of independent living and the quality of life of these fragile subjects, but also on the sustainability of healthcare systems [1]. The majority of older adults prefer to stay in the comfort of their own homes, and given the costs of nursing home care [2], it is imperative to develop technologies that help older adults to age in place. For these reasons, independent living and pro-active health-care are becoming strategic application areas for major research programmes all over the world [3], considering that the senior population is projected to double as a percentage over the whole population in the next decades [4]. Indeed, many research groups are studying pervasive solutions to continuously and unobtrusively monitor fragile subjects at their homes, reducing health-care costs and supporting the medical diagnosis. These studies are possible due to the increased availability of affordable and reliable sensing infrastructures, which allowed to build the so-called smart-homes: residences equipped with technology (i.e. sensors and actuators) that enhances the safety of patients at home and monitors their health conditions [5]. Continuous in-home monitoring should avoid video/audio recording, since it is often perceived as too privacy obtrusive in a home environment. Moreover, there are indications of a general adversity or disaffection of users to wearables sensors targeted to health-care related applications [6]. Hence, smart-home sensing infrastructures should mostly rely on environmental sensors in charge of capturing the interaction of the subject with the home infrastructure.

Among the several health conditions which can be continuously monitored within smart-homes, clinicians are interested in understanding the everyday functioning of individuals to gain insights about difficulties that affect the quality of life [7]. One of the most frequent threats to independent living is cognitive decline, whose early symptoms often lead to a mild cognitive impairment (MCI) diagnosis. According to the International Working Group on MCI, there is evidence of subtle differences in performing activities of daily living (ADLs) among MCI patients compared to both healthy older adults and individuals with dementia [8]. Hence, from a medical point of view, there is a clear interest in methods to monitor ADLs of the elderly with the goal of identifying specific abnormal behaviors as indicators of cognitive decline. For these reasons, several pervasive frameworks to assess the behavior of smart-home's inhabitants have been proposed. Figure 1.1 illustrates the general architecture of such systems.

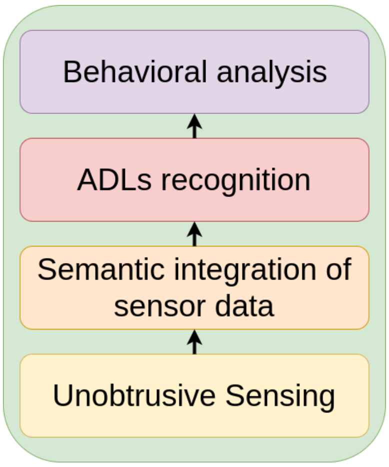


Figure 1.1: General architecture of behavioral analysis frameworks

In the following, we provide a high-level description for each component of this architecture.

1.1.1 Unobtrusive sensing

In order to recognize activities and anomalous behaviors, there is a need for a reliable and unobtrusive sensing infrastructure in charge of capturing the subject's interaction with the home environment [9]. Advancement of semiconductor fabrication technology and smart algorithms has made it possible to design and develop smart sensors with built-in intelligence [10]. For instance, power sensors have been developed in order to monitor usage time and duration of different appliances. Similarly, water sensors exploit an in-line flow transducer to give an indication of when water is being used. In the literature, several wireless sensors have been used to monitor the execution of ADLs, like magnetic sensors, presence sensors, pressure sensors, power sensors, water sensors, etcetera. A significant unobtrusive sensing system has been proposed in [11], where a smart apartment was equipped with a wide variety of sensing devices, like a smart floor to constantly measure gait and balance parameters of inhabitants, smart lighting, water monitoring, and many other sensors to continuously and unobtrusively monitor the subject's interaction with the home environment.

1.1.2 Semantic integration of sensor data

Raw sensor data by itself might not provide meaningful context that can be directly understood and utilized by behavioral reasoning algorithms. For instance, if a sensor identified as $P3$ attached to the electrical stove's plug detects a sudden increase of power consumption, we want to map it to the high-level concept *"the kitchen's stove has been turned on"*. An abstraction layer is then needed to annotate sensor measurements with semantics that are structured around a set of contextual concepts [12] (e.g., smart-home infrastructure, household objects, home locations). The main benefit of mapping raw sensor data to semantic is that reasoning algorithms are independent from the specific sensing infrastructure being used. Hence, there is a need for tools that model at high-level the context gathered by raw sensor data. The survey proposed in [13] gives a broad overview of context modeling techniques in pervasive computing scenarios (e.g., Context Modeling Language [14], ontologies [15], hybrid methods [16]). This thesis is not focused on context modeling, but we recognize that it is a fundamental aspect which should be taken into account when developing behavioral analysis frameworks.

1.1.3 ADLs recognition

In order to accurately detect abnormal behaviors, a reliable ADLs recognition system is needed. Indeed, detecting the specific activities being performed is sometimes a pre-requisite to detect an anomaly [17]. An ADLs recognition algorithm takes as input the pre-processed sensor events and produces as output the most likely performed activities. Several solutions have been proposed in the literature, and in general they are they are divided into two macro categories: *data-driven* and *knowledge-driven*. *Data-driven* methods rely on machine learning techniques to build the activities model from sensor data [18]. The strong point of this category of techniques is that they are good at handling the intrinsic noise and uncertainty of sensor data. The main drawback is that a large annotated dataset of ADLs should be acquired to capture most execution patterns in different situations [19]. Indeed, activity execution patterns are strongly coupled to the individuals characteristics and home environment, and the portability of activity datasets is an open issue [20]. As a consequence, ideally one extensive ADLs dataset should be acquired from each monitored individual. Unfortunately, acquiring ADLs datasets is very expensive in terms of annotation costs [21]. Besides, activity annotation by an external observer, by means of cameras or direct observation, violates the users privacy. In order to solve these issues, *knowledge-driven* solutions have been proposed to manually specify ADLs through logic languages and ontologies. Those models are matched with acquired sensor data to recognize the activities [22]. The main advantage of these techniques with respect to *data-driven* methods is that they can capture complex semantic relationships between sensor events and activities. The main shortcoming of this approach relies on the rigidity of specifications. For instance, complex ADLs are often specified through temporal sequences of simpler actions [23]. In fact, it is not always feasible to enumerate all the possible sequences of actions describing a complex ADL. Moreover, this rigidity does not allow to deal with noisy or uncertain sensor measurements.

1.1.4 Behavioral analysis

Automatic and continuous monitoring of the behavior of fragile subjects addresses many issues of classic solutions. Indeed, questionnaires and interviews have been used to assess cognitive health about the ability to perform various kinds of ADLs [24]. This approach is of course prone to reporting bias; moreover, it cannot be applied for continuous monitoring, since it incurs evident overheads in terms of time, resources and monetary costs.

An anomaly detection system needs an accurate model of the regular behavior of the monitored subject in order to detect when anomalies occur. Indeed, different subjects may perform the same activities in very different ways. Moreover, each subject may adhere to his/her specific medical prescriptions (e.g., medicine intake time, diet, rehabilitation exercises, ...). Time context (e.g., the day of the week, season, holidays, ...) should also be taken into account, since activities can be performed differently depending on temporal context.

The majority of the proposed anomaly recognition methods exploit probabilistic [25] or clustering [26] techniques in order to construct the "normal behavior" of the subject analyzing sensor data without abnormal behaviors. Anomalies are then detected on new sequences of sensor data when divergences from the original model are found. The main drawback of this type of approaches is that behavioral changes are detected without giving specific explanations of what happened. Some other works proposed supervised learning techniques to detect the general anomaly's category (e.g. omission, substitution, replacement, ...) [27]. However, the results show a high rate of false positives. Moreover, the acquisition of a comprehensive annotated dataset of abnormal behaviors is hardly feasible.

1.2 Research contributions

In this section, every research contribution of this thesis is introduced. It is important to note that these contributions have been achieved in collaboration with my research group, which is EveryWare Lab[1] at University of Milan (Italy). Moreover, the unsupervised ADLs recognition algorithm proposed in Chapter 4 is the result of a collaboration with researchers of University of Cagliari (Italy) and University of Mannheim (Germany). However, even if the proposed contributions are the result of a teamwork, I can list my specific contributions.

The major research contributions of this thesis are two hybrid smart-home ADLs recognition algorithms. We also propose an unobtrusive method to detect the manipulations that elderly people perform on household objects. Finally, we

[1] http://everywarelab.di.unimi.it/

introduce interesting research contributions regarding novel abnormal behaviors detection algorithms. In the following we introduce these contributions; in this context, my specific contributions are highlighted.

1.2.1 Hybrid techniques to recognize ADLs

First technique: Supervised activity recognition through statistical and symbolic reasoning

In Chapter 3, we propose a hybrid method to recognize ADLs which is based on a combination of supervised learning and knowledge-based conditions to refine the statistical predictions [3]. The proposed technique combines *data-driven* and *knowledge-driven* methods in order to exploit the strong points of both approaches. A machine learning algorithm is in charge of classifying, for each sensor event, the most likely performed ADL. In particular, time-based features are extracted from windows of consecutive sensor events to capture temporal relationships between events. A knowledge-based algorithm, named SMART AGGREGATION, is then in charge of grouping together those sensor events which most likely belong to the same activity instance and to correct possible mis-predictions produced by the machine learning algorithm. We experimentally compare the activity recognition ability of our approach with a state of the art technique proposed in the literature showing its superiority both on a lab-acquired dataset and on a real-home dataset.

Chapter 3 is based on the following publication:

- Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, Rim Helaoui, *"SmartFABER: Recognizing Fine-grained Abnormal Behaviors for Early Detection of Mild Cognitive Impairment"*. Artificial Intelligence in Medicine, Elsevier, 2016.

My Contributions:

- Concept and methodology design of the SMART AGGREGATION algorithm.
- System's implementation.
- Design of the evaluation methods.
- Experiments execution.
- Collaboration in results analysis and interpretation.

Second technique: Unsupervised activity recognition through ontological and probabilistic reasoning

In Chapter 4, we propose an unsupervised method which overcomes the limitations of *data-driven* and *knowledge-driven* approaches [19]. First, it does not need the acquisition of an expensive labeled dataset. Second, the activity model is based on general semantic relations among activities and smart-home infrastructure; hence, the model can be seamlessly reused with different individuals and in different environments. We rely on ontological reasoning to derive necessary conditions about the sensor events that must occur during the execution of a specific activity in the current environment. This also enables to extract semantic correlations among fired sensor events and executed ADLs. Based on the semantic correlations, a statistical algorithm (named STATISTICAL ANALYSIS OF EVENTS) pre-processes sensor events to identify candidate activity instances, i.e., initial hypotheses about the start and end time of occurred activities. Finally, we translate our ontological model in a Markov Logic Network (MLN) [28], and perform probabilistic reasoning to refine candidate activity instances and check their consistency. Our MLN model is carefully crafted to support the recognition of interleaved activities. We performed extensive experiments with real-world datasets of ADLs performed by 22 individuals in two different smart home environments. Results show that, even using a smaller number of sensors, the performance of our unsupervised method is comparable to the one of existing methods that rely on labeled activity datasets.

Chapter 4 is based on the following publication:

- Daniele Riboni, Timo Sztyler, Gabriele Civitarese, Heiner Stuckenschmidt, *"Unsupervised Recognition of Interleaved Activities of Daily Living through*

Ontological and Probabilistic Reasoning". UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2016.

> ***My contributions:***
>
> - Concept and methodology design of the STATISTICAL ANALYSIS OF EVENTS algorithm.
> - Collaboration in designing a part of the MLN knowledge base.
> - System's implementation (except ontological reasoning).
> - Collaboration in the design of the evaluation methods.
> - Collaboration in experiments execution.
> - Collaboration in results analysis and interpretation.

1.2.2 Recognition of objects manipulations

Monitoring the interaction of the subject with everyday objects is crucial to accurately detect ADLs. Moreover, clinicians are interested in monitoring *how* objects are manipulated in order to assess cognitive health. This topic is dealt in Chapter 5, where a novel framework to unobtrusively recognize object manipulations is presented [29]. In particular, we take advantage of current commercial low cost and low energy consumption multi-sensor devices that can be attached to everyday objects. We collected a dataset of more than two thousands labeled manipulations, and we report encouraging preliminary results on their recognition through machine learning techniques applied to accelerometer data collected from the objects. We believe that our study contributes to the design of a sensing subsystem that could be effectively integrated into the smart-home environments used in several previous works on monitoring complex activities at home [30], independently from the algorithmic method being used, since object manipulations may be considered as simple events.

Chapter 5 is based on the following publication:

- Gabriele Civitarese, Stefano Belfiore, Claudio Bettini. *"Let the objects tell* *what you are doing"*. UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, 2016.

My specific contributions:

- Problem identification and concept design.
- Methodology design.
- Design of evaluation methods.
- Results analysis and interpretation.

1.2.3 Fine-grained and long-term anomalies recognition

In Chapter 6, we propose a novel technique for fine-grained abnormal behavior recognition [31, 3, 17]. Our method relies on medical models describing abnormal activity routines that may indicate the onset of early symptoms of MCI. These models have been acquired through the collaboration with cognitive neuroscience experts, and they are translated into first-order logic rules. The input for anomaly recognition is high-level events produced by the sensing infrastructures and detected ADLs. The considered abnormal behaviors include inappropriate timing and unnecessary repetitions of sub-actions, but also high-level observations like *irregularly consuming meals* or *often consuming cold meals*. Different from previous works which consider an anomaly as a simple divergence from a regular behavior [25, 26], we use symbolic reasoning over recognized activities to detect abnormal behaviors at a fine-grained level. In particular, we focus our studies on abnormal behaviors related to object manipulations. Experimental results show that our technique is able to detect most of the abnormal behaviors that we have targeted while producing a small number of false positives.

Moreover, we also propose a novel long-term analysis method to detect significant changes in the trend of performing activities (e.g., changes in habits regarding the timing of meal consumption) [32]. Indeed, when considered in isolation, fine-grained abnormal behaviors are only weak indicators of possible cognitive issues. On the contrary, the frequency of anomalies detected over long periods of time and their temporal trend are much stronger indicators.

Chapter 6 is based on the following publications:

- Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, Rim Helaoui, *"SmartFABER: Recognizing Fine-grained Abnormal Behaviors for Early Detection of Mild Cognitive Impairment"*. Artificial Intelligence in Medicine, Elsevier, 2016.

- Daniele Riboni, Gabriele Civitarese, Claudio Bettini. *"Analysis of Long-term Abnormal Behaviors for Early Detection of Cognitive Decline"*. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PASTA2016: Workshop on PervAsive Technologies and care systems for sustainable Aging-in-place), IEEE, 2016.

- Gabriele Civitarese, Claudio Bettini. *"Monitoring Objects Manipulations to Detect Abnormal Behaviors"*. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, 2017.

My specific contributions:

- Collaboration in concept design of logic representations of fine-grained anomalies.

- Collaboration in methodology design of fine-grained anomalies recognition.

- Design of evaluation methods for specific fine-grained anomalies related to object manipulations.

- Results analysis and interpretation for the experiments concerning fine-grained abnormal behaviors.

1.3 Outline

The rest of the thesis is structured as follows. Chapter 2 provides a wide overview of the state-of-the-art for both activity and anomaly recognition, and it presents the specific challenges tackled by this thesis. Chapter 3 presents a hybrid statistical/symbolical framework to recognize ADLs which uses a *knowledge-based* algorithm to refine the prediction of a supervised classifier. In Chapter 4 we present an unsupervised ADLs recognition algorithm that combines ontological and probabilistic reasoning. Our contributions on object manipulations detection are presented in Chapter 5. Chapter 6 focuses on a use case of ADLs recognition, which is the detection of abnormal behaviors to support clinicians diagnosis of cognitive decline. Finally, Chapter 7 concludes this work.

Chapter 2

Related work

2.1 Introduction

Smart-home activity recognition systems proved to be effective for supporting the diagnosis and improving healthy aging [33, 34]. In the literature, various strategies have been proposed to devise effective and unobtrusive activity monitoring systems by exploiting pervasive computing technologies [35]. A popular research direction for activity recognition consists in exploiting audio-visual information recorded by cameras and microphones with the help of sound, image and scene recognition software [36, 37]. However, those methods are sometimes tolerated in retirement residences, but much less in private homes due to the privacy issues that they determine. Other proposed activity recognition systems are mainly based on data acquired from body-worn accelerometers [38, 39, 40] in order to recognize physical activities. However, solutions based on wearables are critical: there is no guarantee that wristbands or pendants are constantly worn. There are also indications of a general adversity or disaffection of users to wearables targeted to health-care related applications [41]. Moreover, those methods are not well suited to recognize complex activities, like ADLs executed at home, which are characterized by the interaction of the individual with several objects and furniture.

For the above-mentioned reasons, in this chapter we restrict our attention to non-invasive sensor-based techniques. In this context, this chapter introduces an overview of the state-of-the-art on unobtrusive sensing (Section 2.2), as well as on activity (Section 2.3) and anomaly (Section 2.4) recognition. Finally, Section 2.5 introduces the research questions addressed by this thesis.

2.2 Unobtrusive sensing

Unobtrusive ADLs recognition relies on several sensors that detect the inhabitant's interaction with objects and furniture and his/her movements in the home. The measurements produced by those devices are continuously transmitted to a home gateway, in order to be used by behavioral monitoring applications. In the following, we introduce an overview of unobtrusive sensing technologies adopted in the literature.

2.2.1 Environmental sensing

Environmental sensors are cheap devices that unobtrusively monitor the interaction of the subject with the home infrastructure and his/her movements. The most common are binary sensors [42]; i.e., sensors which produce as output "0" or "1" depending on the interaction being performed. Examples of such devices are magnetic sensors (e.g., to detect when doors or drawers are opened or closed) and pressure sensors (e.g., to detect when the subject sits on a chair). Passive InfraRed (PIR) sensors have been widely used [43, 44, 45] to monitor the presence of the subject in specific home locations and to track his/her motion patterns. Coarse-grained human movements have been monitored also by using air pressure sensors [46]. Power meter sensors have been proposed to detect the usage of home appliances [47, 48]. Water usage can be monitored using flow meters [49] or attaching low-cost microphones to the pipes of water distribution infrastructure [50].

2.2.2 Monitoring the interaction with everyday objects

Besides environmental sensors, smart-home activity recognition highly benefits from tracking the inhabitant's manipulations of everyday objects [51]. The majority of the solutions in the literature are based on a combination of RFID technologies and wearable devices [52, 53]. In those approaches, the subject needs to wear a glove which acts as an RFID reader. An RFID tag is hence attached to each object of interest in order to detect objects interaction. The main issue with this approach is that the monitored subject needs to continuously wear the glove. Moreover, it has been shown that RFID technologies are not reliable for a real deployment [54]. Finally, those methods only detect the generic interaction of the subject with the objects, thus not providing a specific information about the performed manipulation.

2.3 ADLs recognition

ADLs recognition methods in ubiquitous computing can be broadly classified into two categories: data-driven methods and knowledge-based methods [35]. Data-driven techniques are more flexible in terms of implementation (i.e., they do not require rigid definitions of activities) and they are more robust with respect to noisy and uncertain sensor data. Moreover, since they do not rely on a rigid specification of how ADLs should be performed, they potentially can capture more variations of the considered activities. For example, in case of the activity of preparing a meal, a person can turn on the stove in the beginning and then retrieve the food items from the cabinets or vice versa. Depending on the recipe, the person can skip or add some steps to the activity. Thus, in such cases we need techniques that are scalable. However, the acquisition of a comprehensive annotated dataset is expensive and often unfeasible. Moreover, it is difficult to incorporate the domain knowledge about the activity using these techniques.

On the other hand, the knowledge-driven techniques are more powerful to represent the semantics of the sensor events. These techniques use the domain knowledge to conceptually model an activity. In this way, an activity can be modeled without the need for large training data. However, such techniques lack the benefits of flexibility and scalability in the system. In addition to the above-mentioned methods, *hybrid* solutions have also been proposed to combine data-driven and knowledge-based approaches. In the following, we go into more details of these categories.

2.3.1 Data-driven methods

Data-driven methods rely on a training set of sensor data, labeled with executed activities, and machine learning algorithms to build the activities' model. Observations regarding the user's surrounding environment (in particular, objects' use), possibly coupled with body-worn sensor data, are the basis of those activity recognition systems [3, 55]. In [56] the authors propose a time series data analysis method to segment sequences of sensor events in order to recognize ADLs. The application of Hidden Markov Models inference is proposed in [42] to recognize activities based on features extracted from recent sensor events according to a sliding window. Conditional Random Fields [57] and Emerging Patterns [58] have also been proposed in order to detect sequential, interleaved and concurrent activities.

The authors in [59] combine Bayes Networks with interval algebra in order to explicitly model complex temporal dependencies over time intervals. In [60], the authors propose a supervised learning classifier that automatically adapts its model according to the dynamically discovered context (i.e., new data sources).

However, since training data is hard to acquire in realistic environments, systems relying on supervised learning are prone to serious scalability issues the more activities and the more context data are considered. Moreover, datasets of complex ADLs are strongly coupled to the environment in which they are acquired (i.e., the home environment and the sensors setup), and to the mode of execution of the specific individual. Hence, the portability of activity datasets in different environments is an open issue [61].

2.3.2 Knowledge-based methods

Knowledge-based methods rely on specification-based definitions of the characteristics and semantics of complex activities. These are matched with available sensor data to recognize the current activity. Those definitions are usually expressed through logical axioms, rules, or description logics [62, 63, 64]. Hence, different frameworks for knowledge representation and reasoning have been investigated to appropriately model complex human activities by means of ontologies. In particular, description logics [65] have emerged among other symbolic formalisms, mostly because they provide complete reasoning (i.e., every true well-formed formula can be derived) supported by optimized automatic tools.

Some proposed methods take uncertainty in sensor data acquisition into account by means of probabilistic and fuzzy logic [66]. Other works rely on description logic languages to formally express activity definitions [63, 64]. Background knowledge of ADLs has been used to create activity models, used to recognize ADLs based on the similarity of sequences of sensor events to the general models [67]. Ontological reasoning has also been proposed to perform dynamic segmentation of sensor data [68, 69, 70] or to refine the output of supervised learning methods [71]. Defeasible reasoning has been adopted to enhance existing sequential activity recognition systems by detecting interleaved activities and handling inconsistent or conflicting information [72]. A further method to segment activities based on their semantic description is proposed in [73]; that method also supports the recognition of overlapped activities.

However, those works rely on rigid assumptions about the simpler constituents of activities. Hence, while the specification-based approach is effective for activities characterized by a few typical execution patterns, it is hardly scalable to the comprehensive specification of complex ADLs in different contexts.

2.3.3 Hybrid methods

Given the limitations of both statistical and symbolic approaches, a few hybrid activity recognition systems have been proposed in the literature, which vary on the adopted reasoning techniques and on their interaction mechanisms.

An interesting instance of those approaches is Markov Logic Networks (MLN), a probabilistic first-order logic [74]. Given a training set, and a set of probabilistic formulas, with MLN it is possible to learn a weight for each grounded formula by iteratively optimizing a pseudo-likelihood measure. Those weights represent the confidence value of the formula. Deterministic formulas can be added to probabilistic ones to express deterministic knowledge about the domain of interest. Different reasoning tasks can be executed to infer additional information based on formulas and facts [31]. A similar approach was adopted in [23] to model and recognize activities at different levels of complexity using probabilistic description logic. The advantage of using probabilistic logic is that it allows defining complex knowledge-based constraints which can capture the intrinsic uncertainty of sensor measurements. Indeed, learning the weights of those constraints allows combining the strong point of knowledge-base and data-driven methods, thus improving the recognition rate. However, those approaches still require the acquisition of a labeled dataset.

In [75] the authors proposed to exploit ontologies in order to derive semantic similarity between sensor events. This similarity is then used to segment sensor data, obtaining sequential activities' patterns used to train a clustering model. The semantic segmentation of sensor data allows to accurately individuate transitions between activities without supervised techniques. The main drawback of this method is that it requires a comprehensive dataset of activities (even if not labeled) acquired from the monitored subject to construct an accurate activities model.

Hybrid ontological and statistical reasoning is also proposed in [76] to continuously assess the fall risk of a senior at home, by integrating data acquired from different fall detection systems and environmental sensors. Semantic reasoning considering the context is hence used to reduce the number of false positives obtained by the statistical fall-detection system.

2.4 Behavioral analysis

2.4.1 Applications of activity recognition to MCI diagnosis

Several studies in the neuropsychology research field show that it is possible to distinguish between cognitively healthy adults and cognitively impaired individuals based on subtle differences in their behavioral patterns [77, 78]. For instance, in [79], subjects were asked to execute a set of predefined ADLs in the observation room of a clinical center, while two cameras recorded their activities. Researchers annotated the dataset manually based on the observation of the video recordings, giving partial scores to the performed activities considering different factors, including activity duration, omitted steps, and the number of repeated steps. The partial scores were then aggregated to obtain a comprehensive score, who proved to be effective in distinguishing MCI subjects from Alzheimer's patients, and cognitively healthy seniors from MCI subjects. There is a growing interest in exploiting pervasive computing technologies to automatically capture and measure those differences [34].

Machine learning methods applied to accelerometer data and video recordings were used in [80] to distinguish between cognitively healthy seniors and Alzheimer's patients based on activity execution and gait events. Similarly, sensors and video recordings were used in [81] to distinguish between MCI and Alzheimer's patients. Those methods were applied in controlled environments, while we aim at monitoring the elderly's activities at a fine-grained level at home.

Several European projects have addressed the usage of ICT technologies for enhancing active and healthy aging [82, 83, 84] and for supporting people with dementia at home [85]. Based on this line of research, different works have proposed to apply machine learning techniques on data acquired in sensor-rich environments, for assessing the cognitive health status of an individual performing a set of ADLs. For instance, motion sensors and contact sensors have been used in [45] to measure low-level activity patterns, such as walking speed and activity level in the home; results have shown that the coefficient of variation in the median walking speed is a statistically significant measure to distinguish MCI subjects from healthy seniors.

A sensor-based infrastructure has been used in [86] to unobtrusively monitor the execution of ADLs by older adults in a smart-home; the results have shown a significant correlation between the cognitive health status of the subject and the level of assistance that he needs to complete the activities. In the work of Dawadi et al. [87], patients were invited to execute a list of routines (e.g., write a letter, prepare lunch) inside a hospital smart-home. Different kinds of sensors were used to detect movements inside the home and to track the use of furniture and appliances. Based on data coming from the home sensors, supervised learning methods were used to assign a score to each performed activity; the score measures the ability of the subject to perform the activity correctly. The achieved scores were used to predict the cognitive status of the patient (cognitive health or dementia). The supervised learning approach has been applied in other works, including [88, 89, 90], using several other learning methods. However, while a significant correlation exists between the inferred activity scores and the cognitive health status of the individual, those methods do not provide a description of the observed behavioral anomalies. On the contrary, the medical assessment would benefit from detailed knowledge of the abnormal behavior of the patient.

2.4.2 Long-term analysis of activity data

In the aforementioned works, the detection of abnormal behaviors is mostly done on a short-term basis and does not take into account the patient's personal habits. Other works have proposed methods to model the patient's usual behavior from the activities performed in the past and use this model to detect anomalies as changes from his/her usual behavior. In [91], a method has been proposed to monitor the circadian (24-hours) variability of the patient's activities using location sensors and statistical calculations were performed regularly on sensors data to recognize possible deviations in the patient's behavior. In [92] in-home activities and sleep restlessness were captured using different sensors and a simple alert system was implemented to detect changes in the activity patterns and generate health alerts that were sent to clinicians to be rated for their clinical relevance. These ratings were then used as ground truth in developing classifiers to recognize relevant alerts. In [93], the authors propose a technique to detect recurrent ADLs patterns, as well as their variations, by mining heterogeneous multivariate time-series from sensor data acquired in a smart home.

Another approach based on temporal data mining was presented in [94]. Frequently occurring temporal relationships between activities were extracted from the observed history of sensor events and used to model the probability that a particular event should or should not occur on a given day. A technique based on unsupervised learning is proposed in [95] to automatically discover ADL patterns and their variations. That technique is coupled with an activity recognition module and with visualization tools to allow practitioners inspecting the trend of activity patterns. Visualization of spatiotemporal data extracted from the long-term observation of elderly's activities at home is used in [96] to identify potential risk situations.

2.5 Research problems addressed by this thesis

In this section we outline the research questions tackled in this thesis. For each question, we introduce the research problem and we indicate the specific chapter of the thesis where the problem is addressed.

Q1) How can knowledge-based and data-driven methods be combined in order to improve ADLs recognition?
As previously mentioned, *data-driven* methods are more scalable and they are robust against the intrinsic noise and uncertainty of sensors measurements, while they lack the capability of capturing important semantic relationships between sensor events and activities. On the other hand, *knowledge-based* methods capture very well the above-mentioned complex semantic relationships, but their specification is often too rigid to cope with the variability of execution of ADLs and to handle noise and uncertainty.

In Chapter 3, we propose a novel hybrid ADLs recognition method which uses knowledge-based conditions to refine the statistical prediction of a supervised learning algorithm. This method thus combines the strong points of both approaches in order to improve the recognition rate.

Q2) How can ADLs be recognized with a scalable method which avoids the acquisition of an annotated dataset?
Supervised learning methods require the acquisition of a large annotated dataset, which is often unfeasible to obtain. On the other hand, *knowledge-based* methods

do not rely on a dataset, but they are often based on rigid specifications.

In Chapter 4, we propose an ADLs recognition method which overcomes the drawbacks of both approaches combining ontological and probabilistic reasoning. First of all, our method is unsupervised and it does not require a training set. Second, the activity model is based on general semantic relations among activities and smart-home infrastructure; hence, we can seamlessly reuse our model with different individuals and in different environments.

Q3) How to unobtrusively recognize fine-grained manipulations performed by the inhabitant on household objects?
Monitoring the interaction of the subject with household items is important to accurately recognize the performed ADLs. Moreover, clinicians are interested in monitoring how objects are manipulated in order to identify abnormal behaviors which can be early symptoms of cognitive decline. In the literature, computer-vision methods have been proposed to track the objects usage [97]. However, the use of cameras is too much privacy intrusive in home environments. Other methods proposed wearable solutions, but there are indicators of disaffection from elderly users to wear such technologies in their home [41].

In Chapter 5 we propose an unobtrusive method to monitor the manipulations of household items. We shift the monitoring burden to the objects' side, by attaching tiny wireless accelerometer directly on the objects. The continuous stream of acceleration data is then analyzed by a machine learning algorithm in order to identify the fine-grained manipulations that the subjects performed on the household items.

Q4) How to recognize abnormal behaviors at a fine-grained level?
A general approach to recognize anomalies is to build a model of the "regular" behavior of the subject in order to identify those activity patterns which diverge from the expected ones. The main drawback of this type of approaches is that behavioral changes are detected without giving specific explanations of what happened.

In Chapter 6 we propose a knowledge-based technique to detect abnormal behaviors at a fine-grained level. The description of anomalies provided by clinicians is translated in first-order-logic rules, which are evaluated considering sensor events, performed activities, and subject-specific information. In this thesis, we focus on abnormal behaviors based on objects manipulations.

Chapter 3

Supervised activity recognition through statistical and symbolic reasoning

3.1 Introduction

In this chapter we present a novel hybrid ADLs recognition algorithm based on a combination of supervised learning and knowledge-based conditions to refine the statistical predictions. We combine *data-driven* and *knowledge-based* approaches to take advantage of the strong points of both techniques. In particular, supervised learning allows us to handle the variability of execution and the intrinsic uncertainty and noise of sensor measurements, while knowledge-based reasoning allows us to consider complex semantic relations between events and activities.

Our method relies on a machine learning algorithm which classifies, for each sensor event, the most likely performed activity. We adopt a state-of-the-art feature extraction technique to capture important temporal relationships between sensor events. A novel knowledge-based algorithm is then in charge of a) grouping together those sensor events which most likely belong to the same activity instance and b) correcting possible mis-predictions produced by the machine learning algorithm.

Few other methods in the literature unified supervised and knowledge-based reasoning for activity recognition. The use of knowledge-based reasoning to refine the prediction of supervised learning has been already proposed for activity recognition [71]. However, that method is limited to simple physical activities. In [98], probabilistic description logics are used to support both modeling and reasoning with uncertainty by combining log-linear models and description logics. However, that work relies on rigid activity patterns, while our method exploits time-based feature extraction and machine learning to mine those patterns from a training set. The probabilistic first-order logic Markov Logic Network has been used in [31], where soft and hard constraints were combined in order to learn the activities performed by the subject. That technique obtained good results on a dataset acquired in a controlled environment, while it performed poorly in a real-home scenario [54].

We experimentally compare the recognition results of our approach with a state of the art technique proposed in the literature, showing its superiority both on a lab-acquired dataset and on a real-home dataset.

The rest of the chapter is structured as follows. Section 3.2 introduces our notation and the formulation of the activity recognition problem. In Section 3.3 the architecture of the proposed method is described in the details. Section 3.4 introduces our smart lab and real-home datasets and it presents the recognition results. Finally, Section 3.5 concludes the chapter.

3.2 Notation

3.2.1 Activities

We denote by *activity class* an abstract activity (e.g., cooking and cleaning), and by *activity instance* the actual occurrence of an activity of a given class during a certain time period. More formally, we define $\mathbf{A} = \{A_1, A_2, \ldots, A_k\}$ as the set of k considered high-level activity classes (e.g.: $\mathbf{A} = \{$*Preparing Meal, Eating Meal, Taking Medicines*$\}$). An instance ai of an activity class $A \in \mathbf{A}$ is an occurrence of A during a timespan. Intuitively, a timespan is a particular time interval represented by a start time and an end time, where not every timestamp between the boundaries necessarily belongs to the timespan. This representation allows us to consider activities performed in an interleaved fashion. In particular, we define a timespan ts as a non-convex time interval characterized by a finite set of non overlapping temporal intervals:

$$\{[x_1, y_1], [x_2, y_2] \ldots, [x_n, y_n]\}$$

where $\forall i\ x_i, y_i \in \mathbf{T}$. Given a timestamp $t \in \mathbf{T}$ and a timespan ts, we say that t belongs to the timespan ts if it belongs to one of its intervals. We denote with a^A_{ts} an instance of the activity class $A \in \mathbf{A}$ occurred during the timespan ts.

3.2.2 Sensor events and semantic integration

We assume a smart home instrumented with sensors to detect interactions with items and furniture, context conditions (e.g., temperature), and presence in certain locations. Figure 3.1 illustrates the relation between recorded sensor events and an activity instance. Hence, during the execution of activity instance ai_1 (*preparing dinner*), the subject executes the operations op_1 (opening the silverware drawer) and op_2 (turning on the microwave oven). Supposing that sensors are available to detect these operations, op_1 and op_2 generate two sensor events se_1 and se_2, whose timestamps corresponds to the time of the respective operation.

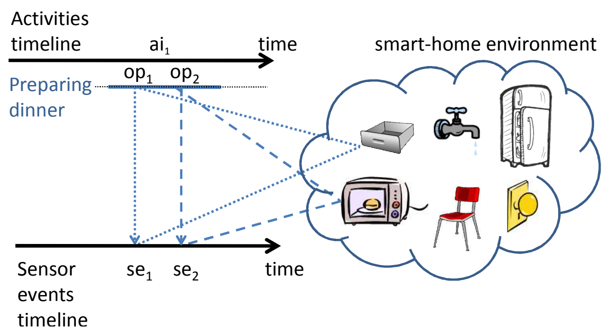


Figure 3.1: The relations between an activity instance, the performed operations, and the involved sensor events.

Raw sensor data by itself might not provide meaningful context that can be directly understood and utilized by reasoning algorithms. An abstraction layer is necessary in order to annotate sensor events with semantics that are structured around a set of contextual concepts (e.g., smart-home infrastructure, household objects, home location, . . .). For this reason, we introduce the concept of *semantic integration*. By *semantic integration* we mean applying simple inference methods to derive high-level events from raw sensor events. In particular, we adopt rules to express conditions about the type of detected raw events, which determine the recognition of high-level events. Those rules may include conditions about the temporal occurrence of raw sensor events. Consider the example below.

Example 3.2.1. Suppose that the infrastructure includes one presence sensor covering the kitchen table area, and one pressure sensor installed on a chair to detect the weight of the person sitting on it. Then, the following rule is used to detect that the person is sitting on a chair at the kitchen table: "if at time t the presence sensor detects a presence near the kitchen table, and after a short lapse of time (at t') the pressure sensor detects that a person sits on the chair, then the current event at t' is *sitting on a chair at the kitchen table*". Formally, this rule expressed in natural language is encoded in propositional logic as follows:

$$ev(\text{SitOnChairAtKitchenTable}, t') \leftarrow ev(\text{PresenceAtKitchenTable}, t) \;\wedge\; ev(\text{SitOnChair}, t') \;\wedge\; t' \geq t \;\wedge\; (t' - t) \leq 5sec.$$

For the sake of this work, we defined those rules by considering common-sense knowledge, the available datasets and the targeted activities. We are aware that a more systematic methodology to define and specify such events would make our method more flexible, but this is out of the scope of this thesis. Moreover, when one (or more) of the sensors that are involved in the generation of an event stops working, we are not able to detect that event anymore. To address this problem, one of the possible solutions is *sensing redundancy*: the same high-level event should be captured by multiple sensing modalities and multiple rules to derive the same event. In future works we will investigate this fault-tolerance aspect.

We define $\mathbf{E}$ as the set of all the considered event types (e.g. $\mathbf{E} = \{$Door_is_opened, Door_is_closed$\}$). We adopt a simple temporal model for the events. We denote as $\mathbf{T}$ the set of all the possible timestamps. A sequence of events is represented as follows:

$$\langle\, ev(E_1, t_1), ev(E_2, t_2), \ldots, ev(E_m, t_m) \,\rangle,$$

where $ev(E_i, t_i)$ indicates that an instance of the event type $E_i \in \mathbf{E}$ occurred at timestamp $t_i \in \mathbf{T}$. A unique timestamp is assigned to each event, based on the time at which the related raw sensor events are received by the central mobile device. In this way we impose a total order on event timestamps $\langle\, t_1, t_2, \ldots, t_m \,\rangle$. Given a sensor-equipped environment, an activity instance a^A_{ts} generates a sequence of events that we call "a^A_{ts} observations", formally denoted by

$$Obs(a^A_{ts}) = \langle ev(E_1, t_1), ev(E_2, t_2), \ldots, ev(E_k, t_k) \rangle$$

where $\forall i\ E_i \in \mathbf{E}$ and $t_i \in ts$.

In this work, we do not adopt ontological reasoning to recognize high-level events. Instead, we use formal ontologies for the sake of interoperability. In particular, event types are represented using our OWL 2 ontology of events, actions, and activities. For instance, the event type *sit on chair at kitchen table* is defined as follows:

$$\begin{aligned} &\text{ETSitOnChairAtKitchenTable} \sqsubseteq \text{EventType} \sqcap \\ &\forall\, \text{hasActor}.\Big(\text{Person} \sqcap \exists\, \text{hasLocomotion}.\text{Sit} \sqcap \\ &\exists\, \text{hasObject}.\text{Chair} \sqcap \exists\, \text{hasLocation}.\text{KitchenTableArea}\Big). \end{aligned}$$

3.2.3 Activity recognition problem

Based on the observation of a set of timestamped events

$$\langle\, ev(E_1, t_1), ev(E_2, t_2), \ldots, ev(E_m, t_m)\, \rangle,$$

the goal of the activity recognition system is to reconstruct which activity instances generated those events. As shown in Figure 3.2, the objective is thus to assign each event to the observations of the activity instance that most probably generated it. Activity instances can also be performed in an interleaved fashion, as it is the case for ai_2 and ai_3 where the subject temporarily interrupts the meal to take medicines.

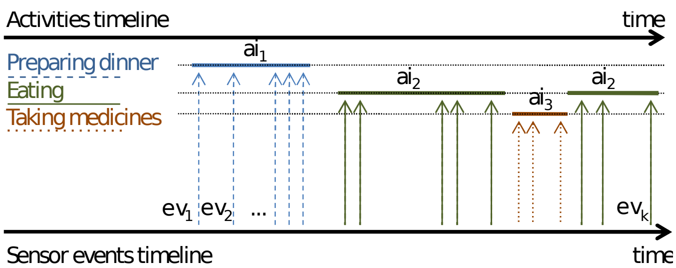


Figure 3.2: Reconstruction of the activity instances generating a set of sensor events.

3.3 The technique

Our hybrid reasoning framework, shown in Figure 3.3, exploits both statistical and knowledge-based methods. In particular, knowledge-based methods are used by the SEMANTIC INTEGRATION LAYER to recognize simple events from raw sensor data (like discussed in Section 3.2.2) and by the SMART AGGREGATION module to identify activity instances. Statistical reasoning, taking into account temporal features, is used to classify events into activities and to recognize activity instances. At each pre-processed event, a time-based supervised learning technique is applied to assign the most probable activity class. The classified events are then post-processed in order to identify the most probable activity instances. In the following,

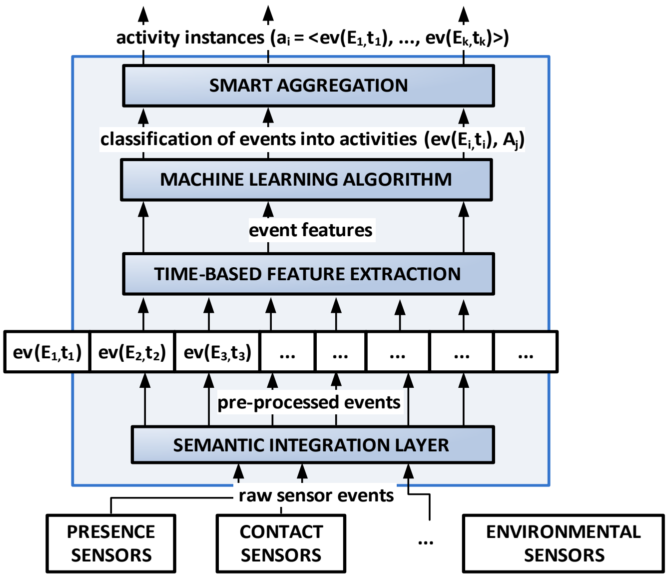


Figure 3.3: The system's architecture

we describe in detail the main components of the proposed system.

3.3.1 Classification of events

The events produced by the SEMANTIC INTEGRATION LAYER are communicated to the TIME-BASED FEATURE EXTRACTION module. For each event $ev(E_i, t_i)$, this module is in charge of building a feature vector representing the sequence S

No.	Feature name	Description
$1 \dots 5$	Repository usage	Each of these features measures the temporally-weighted occurrences of usage of an individual repository (fridge, kitchen cabinet, ...)
6	Stove usage	Measures the temporally-weighted occurrences of stove usage events
7	Cooking pot usage	Measures the temporally-weighted occurrences of cooking pot usage events
8	Food retrieval	Measures the temporally-weighted occurrences of food retrieval events
9	Medicine retrieval	Measures the temporally-weighted occurrences of medicine retrieval events
10	Approaching table	Measures the temporally-weighted occurrences of the event type "approaching kitchen table"
11	Leaving table	Measures the temporally-weighted occurrences of the event type "leaving kitchen table"
12	Start time	Time of the day of the oldest event in the sequence S
13	End time	Time of the day of the most recent event in the sequence S
14	Duration	Difference between end time and start time

Table 3.1: List of considered features

of the n most recent events:

$$S = \langle\, ev(E_{i-n+1}, t_{i-n+1}), \dots, ev(E_{i-1}, t_{i-1}), ev(E_i, t_i)\,\rangle.$$

We adopt the feature extraction technique proposed in [99], since it takes into account temporal aspects, and proved to be effective in recognizing activities based on streams of sensor events. We consider the features listed in Table 3.1. Some of the features (i.e., from feature 1 to feature 11) measure the number of events happened in S that are related to the usage of a particular object or to the presence in a particular area of the home. When computing the value of those features, we use a weighting factor to fine-tune the contribution of each event in S, so that recent events contribute more than older ones. In particular, we use an exponential function to compute the weight for each $ev(E_j, t_j) \in S$ based on the time distance between t_j and t_i (the latter is the most recent event in S):

$$w(t_j, t_i) = exp(-\chi(t_i - t_j)),$$

where the factor χ determines the time-based decay rate of the events. The value of feature $F_k(S)$ (with $k = 1 \dots 11$) is computed as:

$$F_k(S) = \sum_{ev(E_j,t_j) \in S} w(t_j, t_i) \cdot f_k(ev(E_j, t_j)),$$

where $f_k(ev(E_j, t_j))$ is the time-independent contribution of $ev(E_j, t_j)$ to the computation of the F_k value. For instance, when we consider F_3 that measures the number of events in S related to the usage of the fridge, the value of $f_3(ev(E_j, t_j))$ is 1 if E_j corresponds to either "open fridge" or "close fridge"; it is 0 otherwise. The feature vector computed based on S is given as input to a supervised MACHINE LEARNING ALGORITHM to infer the most probable class of the activity instance carried out at t_i. The algorithm is trained using a dataset of activities and feature vectors.

3.3.2 Naive aggregation

The next step is to infer the actual activity instances from the output of the MACHINE LEARNING ALGORITHM by grouping together those events which can be considered observations generated by the same activity instance. Intuitively, temporally close events classified with the same activity class are most likely generated by the same activity instance. We first discuss the baseline approach, named *naive aggregation*. The basic idea of this algorithm is the following: if two consecutive events occurred respectively at t_i and t_{i+1} are classified with the same activity class $A_i = A_{i+1}$, they are considered as observations generated by the same instance of an activity of class A_i. Otherwise, they are considered observations generated by different activity instances.

Example 3.3.1. Consider the case illustrated in Table 3.2. This table illustrates in the first three columns a sequence of events associated with activity classes predicted by the MACHINE LEARNING ALGORITHM; in the fourth column, the ground truth about activity instances; and in the last column the output of the *naive aggregation* method. The naive aggregation algorithm produces 5 different instances of activities. However, it is easy to see that this aggregation is not correct. Indeed, the events E_2, E_4 and E_6 share the same activity class and are temporally close. With high probability, the inferred activity classes for the events E_3 and E_5 are mispredictions, since the "Eating meal" and the "Preparing meal" activity instances would have a too short duration. Moreover, consider the case where events E_1 and

E_2 correspond respectively with *Presence in the kitchen* and *Open the medicine repository*. These two events alone can not be considered as the only observations generated by an instance of a *Taking Medicines* activity: the medicine repository can possibly contain items not related with medicines and it is also possible that it may be opened just to check the content. Another issue of this technique is that two consecutive events labeled with the same activity class but temporally distant (like E_6 and E_7) would be grouped together, while they most likely belong to separate activity instances. Hence, the particular sequence of events illustrated in Table 3.2 should identify a single instance of "Taking medicines" that generated the events from E_1 to E_6.

Table 3.2: An example of *naive aggregation* based on a sequence of classified events

<table>
<tr><th>Event type</th><th>Timestamp</th><th>Predicted activity class</th><th>Actual instances</th><th>Predicted instances</th></tr>
<tr><td>E_1</td><td>08:32:31</td><td>Taking medicines</td><td rowspan="6">$takingMedicines_1$</td><td rowspan="2">$takingMedicines_1$</td></tr>
<tr><td>E_2</td><td>08:32:48</td><td>Taking medicines</td></tr>
<tr><td>E_3</td><td>08:32:55</td><td>Eating meal</td><td>$eatingMeal_1$</td></tr>
<tr><td>E_4</td><td>08:33:02</td><td>Taking medicines</td><td>$takingMedicines_2$</td></tr>
<tr><td>E_5</td><td>08:34:11</td><td>Preparing meal</td><td>$preparingMeal_1$</td></tr>
<tr><td>E_6</td><td>08:34:13</td><td>Taking medicines</td><td rowspan="4">$takingMedicines_3$</td></tr>
<tr><td>E_7</td><td>11:34:27</td><td>Taking medicines</td><td rowspan="3">$takingMedicines_2$</td></tr>
<tr><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td></tr>
<tr><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td></tr>
</table>

3.3.3 Smart aggregation

In order to overcome the limitations of the naive aggregation algorithm, illustrated in Example 3.3.1, we refined our recognition method. We introduce for each activity class $A \in \mathbf{A}$ a set of conditions that are necessary for a sequence of events to be considered observations generated by an instance of that class. For example, assuming that the infrastructure includes sensors to detect the stove usage, any instance of the activity *Preparing a hot meal* should generate some observations related to the usage of the stove. Other examples of conditions may be constraints on the duration of the activity instance or on the number of generated events. Among those conditions, we consider the upper bound about the duration of activity interruptions: the time distance between every pair of consecutive events within the observations generated by an activity instance a_{ts}^{A} must be lower than $maxDelay_A$.

ALGORITHM 1: Smart aggregation

```
Input: A set S = {(ev(E_1, t_1), A_1), (ev(E_2, t_2), A_2), ..., (ev(E_n, t_n), A_n)} of
       events associated with the predicted current activity classes, where ∀i
       E_i ∈ E, t_i ∈ T and A_i ∈ A.
Output: A set 𝒜 of activity instances.

𝒜 ← ∅;
mispredictions ← ∅;
foreach A ∈ A do
    X ← the events in S predicted with A;
    G ← segmentation(X, maxDelay_A);
    foreach g ∈ G do
        if g satisfies all the conditions in C^(A) then
            a ← an activity instance of class A that generated the observations g;
            𝒜 ← 𝒜 ⋃ {a};
        else
            mispredictions ← mispredictions ⋃ g;
        end
    end
end
foreach ev(E, t) ∈ mispredictions do
    I ← ∅;
    foreach a^A_ts ∈ 𝒜 do
        if t lies between the boundaries of ts then
            x = Obs(a^A_ts) ⋃ {ev(E, t)};
            if x satisfies all the conditions in C^(A) then
                I ← I ⋃ {a^A_ts};
            end
        end
    end
    if I ≠ ∅ then
        a' = argmax_{a^A_ts ∈ I} freq(E, A);
        add ev(E, t) to the observations of a';
    else
        consider ev(E, t) as observation of an activity instance of class "other
          activity";
    end
end
return 𝒜;
```

ALGORITHM 2: Segmentation of activity instances

Input: A set of events $X = \{ev(E_1, t_1), ev(E_2, t_2), \ldots, ev(E_n, t_n)\}$ and the threshold $maxDelay$.

Output: A partition of X according to $maxDelay$.

$groups \leftarrow \emptyset$;
$currGroup \leftarrow \{ev(E_1, t_1)\}$;
for $i \leftarrow 2$ ***to*** n **do**
  if $t_i - t_{i-1} < maxDelay$ **then**
    $currGroup \leftarrow currGroup \bigcup \{ev(E_i, t_i)\}$;
  else
    $groups \leftarrow groups \bigcup \{currGroup\}$;
    $currGroup \leftarrow \{ev(E_i, t_i)\}$;
  end
end
return $groups \bigcup \{currGroup\}$;

The value of the upper bound $maxDelay_A$ depends on the activity class A. Those values are determined statistically (i.e., mined from available datasets); as a result, for example, “Preparing meal” will have a higher $maxDelay$ than “Taking medicines”. Formally, let $C^{(A)} = \{c_1, c_2, \ldots, c_k\}$ be a set of necessary conditions expressed in logic over a sequence of events that are observations of any instance of a class $A \in \mathbf{A}$ (e.g. {”The sequence of events must last more than 3 minutes”, “The sequence of events must contain an event regarding the usage of the stove”, $\ldots$}). A sequence of events $s = \langle ev(E_1, t_1), ev(E_2, t_2), \ldots, ev(E_k, t_k)\rangle$ can be considered as observations generated by an activity instance a^A_{ts} if it satisfies every condition in $C^{(A)}$. The set of conditions for each class are determined after a detailed analysis of the semantics of the activity class and on statistics about the available observations acquired from the sensor infrastructure. For instance, the previously discussed condition about the duration of the interruption of an activity $A \in \mathbf{A}$ over a sequence of events s can be expressed as:

$$\forall i : \; ev(E_i, t_i), ev(E_{i+1}, t_{i+1}) \in s \rightarrow (t_{i+1} - t_i < maxDelay_A)$$

The condition about overall duration of the activity $A \in \mathbf{A}$ over a sequence of events s can be expressed as:

$$\forall k : \; ev(E_1, t_1), ev(E_k, t_k) \in s \rightarrow (t_k - t_1 \leq maxDuration_A)$$

where $ev(E_1, t_1)$ is the first event identified in s, and $maxDuration_A$ indicates the maximum duration of the activity A (which is determined statistically like $maxDelay_A$). As a last example, a condition for the activity *Cooking* which expresses that the sequence of events s must contain an event regarding the usage of the stove can be expressed as:

$$\exists i: \ ev(E_i, t_i) \in s \mid E_i = \text{“\textit{Turning on the stove}”}$$

We now introduce the SMART AGGREGATION algorithm, a refined activity instance recognition method. The pseudo-code is shown in Algorithm 1. The first step of the algorithm is a segmentation over the output of the MACHINE LEARNING ALGORITHM: all the events associated with the same activity class A and temporally close (according to $maxDelay_A$) are grouped together. For each group g of events classified with an activity class A, it is checked if it satisfies all the conditions in $C^{(A)}$. If all the conditions are satisfied, an activity instance a^A_{ts} that generated the observations contained in g is recognized. All the events contained in those groups which did not satisfy the conditions of their class are considered as mispredictions. Hence, the algorithm tries to include them in one of the activity instances recognized at the previous step. For each misprediction $ev(E, t)$, the algorithm builds a set I of activity instances a^A_{ts} (that have been recognized in the previous step) such that t lies between the boundaries of the timespan ts and $\{ev(E, t)\} \bigcup Obs(a^A_{ts})$ satisfies all the conditions in $C^{(A)}$. When $|I| > 1$ we choose the most probable instance based on a function $freq$, which computes the frequency of an event being an observation of the instances of a particular activity class. The values of $freq$ for each possible combination of event type and activity class are computed offline based on the annotated dataset. The event $ev(E, t)$ is added to the observations of the instance $a^A_{ts} \in I$, where A is the most frequent activity class. If I is empty, $ev(E, t)$ is considered an observation of an “other activity” instance.

Example 3.3.2. Continuing Example 3.3.1, suppose to apply the smart aggregation algorithm to the same sequence of events considered in Table 3.2. The fourth column of Table 3.3 shows the result of the first step of the algorithm, which applies a segmentation based on the predictions of the MACHINE LEARNING ALGORITHM for the occurred events. Four groups are created at this step.

The first group g_1, classified as *Taking medicines*, is formed by the first, second, fourth and sixth events, which are temporally close according to the threshold $maxDelay_{\textit{TakingMedicines}}$ and associated with the same activity class. The seventh

Table 3.3: An example of *smart aggregation* based on a sequence of classified events

Event type	Timestamp	Predicted activity class	First step	Predicted instances
E_1	08:32:31	Taking medicines	g_1	$takingMedicines_1$
E_2	08:32:48	Taking medicines	g_1	
E_3	08:32:55	Eating meal	g_3	
E_4	08:33:02	Taking medicines	g_1	
E_5	08:34:11	Preparing meal	g_4	
E_6	08:34:13	Taking medicines	g_1	
E_7	11:34:27	Taking medicines	g_2	$takingMedicines_2$
$\ldots$	$\ldots$	$\ldots$		
$\ldots$	$\ldots$	$\ldots$		

event does not belong to g_1 since it is not temporally close to the events in that group; hence, it is assigned to another group g_2, together with other subsequent events. The other two groups are g_3, classified as *Eating meal* and formed by the third event only, and g_4, classified as *Preparing meal* and formed by the fifth event only. Note that g_3 and g_4 are interleaved with g_1. Then, for each group, the algorithm check if it satisfies the conditions $C^{(A)}$ for its activity class. Suppose that both g_1 and g_2 satisfy the conditions for *Taking medicines*; hence, the algorithm creates an activity instances for each of them (TakingMedicines$_1$ and TakingMedicines$_2$, respectively). Then, the algorithm considers g_3 and checks if it satisfies the conditions for *Eating meal*. Suppose that g_3 does not satisfy the conditions, since it violates the constraint about the minimum temporal duration of Eating meal. Hence, the predicted activity class for the third event is considered a misprediction. Since the third event has happened during the timespan of TakingMedicines$_1$, the algorithm tries to include it in that activity instance, and checks if the conditions for *Taking medicines* are still respected. Suppose that the conditions are respected; hence, the third event is included in TakingMedicines$_1$. Similarly, suppose that the group g_3 (containing the fifth event only) violates the constraint about the minimum temporal duration of *Preparing meal*. Hence, the activity class *Preparing meal* associated to the fifth event is considered a misprediction. Supposing that no condition for *Taking medicines* is violated, the algorithm includes the third event in TakingMedicines$_1$. The predicted instances, which correspond to the actual ones, are reported in the fifth column.

3.4 Evaluation

3.4.1 A smart lab dataset

We have acquired a dataset of ADLs and abnormal behaviors, asking to voluntary actors to reproduce the daily routine of 21 elderly persons in our smart home lab. Executed ADLs and anomalies have been carefully designed in collaboration with neuroscience experts to realistically mimic the behavior of two groups: 7 healthy seniors (group 1), and 14 elderly persons with early symptoms of MCI (group 2). During the execution of the daily routines, we have acquired the timestamped data

(a) Magnetic sensor attached to a drawer (b) Presence sensor above the kitchen table (c) RFID reader for medicine boxes and food items

Figure 3.4: Some sensors used in the smart home lab.

coming from the sensors deployed in the smart home and manually annotated the dataset with the start- and end-time of the performed activities and anomalies. In this chapter we will only focus on the evaluation of activity recognition. Results on anomaly recognition can be found in [3]. The following ADLs have been selected to validate our method:

- Preparing food: the patient has to prepare the daily meals (breakfast, lunch, dinner) at appropriate times.

- Consuming meal: when the patient prepares a meal, he has to consume it within a reasonable time period.

- Taking medicines: the patient has to take the prescribed medicines in the due time. We assume that no smart dispenser is used; instead, we assume that the patient keeps all the medicines in a dedicated cabinet.

3.4.2 A real-home deployment

As a first step towards the evaluation of our methods in the actual home of elderly persons, we took advantage of our cooperation with a medical institution and a tele-medicine company as partners of the SECURE[1] project, and deployed our prototype inside the home of an elderly woman aged 74, with a diagnosis of MCI and medical co-morbidities, who lives alone. We will call her Mary in the following. Details about the technical implementation of the system in Mary's home are reported in [54].

(a) Magnetic contact sensor on the fridge door

(b) Passing a tagged medicine box over the RFID reader

(c) A board with temperature sensor over the stove

(d) Passive infrared presence sensor over the kitchen table

Figure 3.5: Part of the sensors deployed at the elderly's home

We acquired a dataset consisting of 55 days of ADLs performed by Mary. In that period of time, we collected data for about 200 instances of activities. We considered the same type of ADLs as for the smart home lab dataset. For the sake of this project, it was not feasible to directly observe the execution of the activities, except for limited periods of time during the setup of the system, due to obvious privacy reasons. Hence, we manually labeled most of the activities offline, based on the observation of raw sensor data; this was possible since the considered activities are relatively easy to distinguish by a human observer based on the collected sensor readings.

3.4.3 Results

We performed an extensive experimental evaluation to compare the proposed system with another hybrid method –named FABER– that was proposed in [31]. That

[1] http://secure.ewlab.di.unimi.it/

method is based on the probabilistic logic Markov Logic Network (MLN) [100], which is used to correlate windows of n consecutive sensor events with the boundaries (i.e., start and end) of activity instances. More details about that technique can be found in [31].

For both datasets, we applied those activity recognition techniques:

- Our method with the *naive aggregation* algorithm;
- Our method with the *smart aggregation* algorithm;
- The FABER method [31].

For each technique, we performed a leave-one-day-out cross-validation, evaluating the prediction's quality in terms of the standard measures of precision, recall and F_1 score (the latter is the harmonic mean of precision and recall). In the following we explain how these measures are computed.

Each predicted activity instance a_{ts}^{A} is characterized by three parameters: the class A of the activity, its start time t_s (i.e., the initial timestamp of ts), and its end-time t_e (i.e., the last timestamp of ts). For each prediction a_{ts}^{A}, we count a *true positive* (TP) when an activity instance of type A actually started in a neighborhood of t_s; i.e., between $t_s - \alpha$ and $t_s + \alpha$. For the sake of these experiments, we set α to 15 minutes. Indeed, for our application scenario (i.e., the recognition of abnormal behaviors), it is sufficient to know the approximated boundaries of activity instances. In the other case, we count the prediction as a *false positive* (FP). Finally, for each activity instance of type A that actually started at t we count a *false negative* (FN) when no prediction exists for an activity of class A having its start-time in a neighborhood of t. Similarly, we count TP, FP and FN for the end-time of predictions and actual activity instances. Precision, recall and F_1 scores are computed as follows:

$$\text{precision } p = \frac{TP}{TP + FP}$$

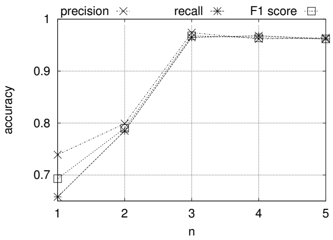
$$\text{recall } r = \frac{TP}{TP + FN}$$

$$F_1 = \frac{p \cdot r}{2 \cdot p \cdot r}$$

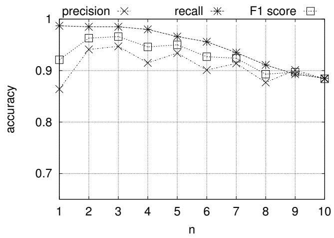
We preliminarily performed cross-validation to select the most appropriate classifier for the machine learning module. As a result, we chose to use a Random Forests classifier [101]. As a first step, we first experimentally calibrated the optimal value for the parameter n, corresponding to the length of the temporal sequence of sensor events to be used by our algorithms. The results of activity boundary recognition on the smart home lab and real home datasets are shown in Figures 3.6 and 3.7, respectively.

With the smart home lab dataset, very positive results have been achieved (with F_1 score that exceeds 0.96) with all the three considered methods. With the MLN-based technique used in FABER, the highest recognition rate is achieved with $n = 3$. This means that, with this dataset, the temporal sequence of the 3 most recent sensor events is sufficient to reliably detect the start or end of an activity. This is due to the quite repetitive way in which activities have been executed in the lab; longer sequences of sensor events may be needed when activities are executed in more variable ways. Values of n lower than 3 produce worse results, while larger values strongly increase the execution times of the learning phase, without increasing recognition rates. With our boundary detection method (i.e., SMART AGGREGATION), the highest recognition rates are achieved using with $n = 2$ or $n = 3$. The *naive aggregation* and the *smart aggregation* methods achieve similar recognition rates; however, the former produces a larger number of false positives. On the contrary, the *smart aggregation* method provides more balanced and slightly higher values of precision and recall.

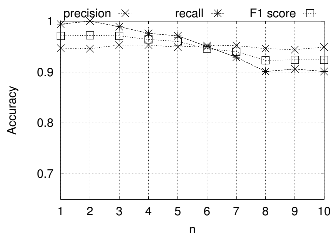
In general, with the real home dataset we achieve lower recognition rates. This is due to the intrinsic variability of activity execution in a real-world situation, with respect to the relatively stable activity execution patterns reproduced in the smart home lab. Moreover, the level of sensor noise due to missing or incorrect sensor readings is inevitably larger in a real home environment than in the lab. With the real home dataset, the MLN-based method used in FABER is the least effective among the ones that we evaluated. The highest recognition rates with MLN are achieved with $n = 5$. We were not able to test the performance with larger values of n, since the execution of the learning algorithm did not terminate in a reasonable amount of time. Independently from the value of n, the FABER method achieved particularly low values of recall. Our technique achieves better results. In particular, the *smart aggregation* method leads to the highest values of F_1 score (slightly above 0.8 with $n = 4$), and very well balanced values of



(a) MLN-based method (used in FABER)



(b) Random forests and *naive aggregation* (used in our method)



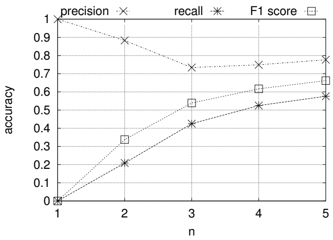
(c) Random forests and *smart aggregation* (used in our method)

Figure 3.6: Smart home lab dataset. Accuracy of activity boundary detection; n is the length of the considered temporal sequence of sensor events

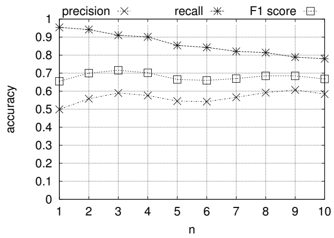
precision and recall. The *naive aggregation* method achieves lower recognition rates, producing a large number of false positives.

Summarizing, with both datasets, the *smart aggregation* algorithm of our method reduces the number of false positives with respect to the FABER method, and improves the overall accuracy.

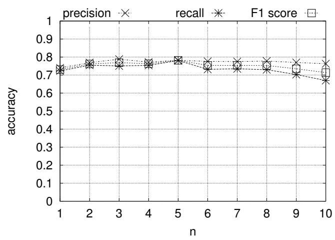
In order to compare our activity recognition method with a well-known technique, we implemented the method proposed by Van Kasteren et al. in [42]. That method is based on the usage of Hidden Markov Models (HMMs) [102]. HMMs are generative probabilistic models consisting of a temporal sequence of hidden variables and observable variables. The general structure of HMMs is shown in Figure 3.8. A hidden variable at time t, named $x(t)$, depends only on the hidden variable at time $t-1$ (named $x(t-1)$). An observable variable at time t, named $y(t)$, depends only on the hidden variable $x(t)$. In our case, the observable variable $y(t)$ corresponds to the observations of the occurrences of sensor events during t, while the hidden variable $x(t)$ corresponds to the class of the activity instance performed during t.



(a) MLN-based method (used in FABER)



(b) Random forests and *naive aggregation* (used in our method)



(c) Random forests and *smart aggregation* (used in our method)

Figure 3.7: Real home dataset. Accuracy of activity boundary detection; n is the length of the considered temporal sequence of sensor events

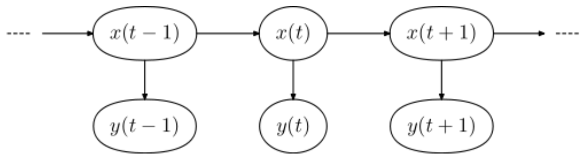


Figure 3.8: The general structure of a HMM

HMMs are specified using three probability distributions:

- The probability distribution over initial hidden states;
- The probability distribution of transitions among hidden states;
- The probability distribution of an hidden state $x(t)$ generating an observation $y(t)$.

Those distributions are estimated using an annotated dataset. Thanks to this model, the prediction of an activity at t depends not only on the current observations, but also on the activity predicted at the previous time slice $t-1$. Applying the method proposed in [42], the temporal sequence of sensed events is divided into time slices of constant length (60 seconds). The sensor events occurred during the time slice t are represented by a binary feature vector $\overline{F}_t = \langle F_{E_1}, \ldots, F_{E_m} \rangle$, in which every event type $E_i \in \mathbf{E}$ corresponds to a feature F_{E_i} (we remind that $\mathbf{E}$ is the set of all the considered event types). Among different feature extraction methods, the most effective proved to be the following:

- **Change point**: The value of the feature F_E at t is 0 if no instances of the event type E occurred during t; it is 1 otherwise.
- **Last**: The value of the feature F_E at t is 1 if the type of the most recent event occurred within or before t is E; it is 0 otherwise.

The above feature extraction methods can be combined to obtain the **Change point + last** method, in which feature vectors are built by concatenating the vectors obtained using the *change point* and the *last* representations.

Example 3.4.1. Consider three event types: E_1, E_2 and E_3. The following tables show a possible event sequence and the corresponding time slices representations.

Event type	Timestamp
E_1	10:44:31
E_2	10:44:48
E_3	10:46:23

Time slice	Change point	Last	Change point + Last
10:44	1 1 0	0 1 0	1 1 0 0 1 0
10:45	0 0 0	0 1 0	0 0 0 0 1 0
10:46	0 0 1	0 0 1	0 0 1 0 0 1

Inferencing to derive the most probable sequence of hidden states generated by the sequence of observations is performed by applying the well known Viterbi algorithm [102]. In order to identify the boundaries of the activities instances, we applied a segmentation of the time slices considering the predicted hidden states: consecutive time slices classified with the same activity class are considered as part of the same instance. A leave-one-day-out cross-validation was performed on both datasets in order to obtain the measures of precision and recall. The comparison with our method using *smart aggregation* and $n = 3$ is shown in Tables 3.4.3 and 3.4.3.

The results show that, with both datasets, our method achieves better values of precision and recall than the HMM-based technique, with all the three feature extraction methods. In particular, with the smart lab dataset, the improvement in the recall rate obtained by our method is very relevant (i.e., from 0.925 to 0.989).

Technique	Precision	Recall	F1
HMM (Change point)	0.929	0.925	0.927
HMM (Last)	0.919	0.915	0.917
HMM (Change point + Last)	0.929	0.925	0.927
Our method	**0.957**	**0.989**	**0.972**

Table 3.4: Comparison between our method and the HMM-based technique with the smart lab dataset

Technique	Precision	Recall	F1
HMM (Change point)	0.820	0.736	0.776
HMM (Last)	0.790	0.723	0.755
HMM (Change point + Last)	0.821	0.726	0.770
Our method	**0.855**	**0.771**	**0.811**

Table 3.5: Comparison between our method and the HMM-based technique with the real home dataset

The precision rate also improves consistently (i.e., from 0.929 to 0.957). This trend is confirmed with the real home dataset.

3.5 Summary

In this chapter, we presented a hybrid activity recognition method which uses knowledge-based reasoning to refine the classification of a machine learning algorithm and, most importantly, to infer the most likely activity instances. The proposed method relies on Random Forest and temporal-based feature extraction to classify, for each high-level event, the most likely activity. Then, an algorithm based on a set of knowledge-based conditions groups together those sensor events which most likely belong to the same instances. We experimentally evaluated that our method outperforms state-of-the-art solutions purely based on supervised learning. Our method addresses the research question **Q1)** presented in Section 2.5, thus proposing a novel approach to combine knowledge-based and data-driven methods to improve ADLs recognition. However, the proposed method has several limitations. First of all, the semantic conditions used by the SMART AGGREGATION algorithm requires a relevant knowledge-engineering effort. It is questionable if such system would scale with an increasing number of activities.

Second, this method requires the acquisition of an annotated dataset, which is often unfeasible. Even if a labeled dataset is available, usually it is adequate only for specific subjects and environments. Hence, the flexibility of the proposed system is questionable. In the following chapter, we introduce another hybrid ADLs recognition algorithm which overcomes these limitations. In particular, it does not require training sets, and it relies on a more flexible specification of semantic relations among activities and smart-home artifacts.

Chapter 4

Unsupervised activity recognition through ontological and probabilistic reasoning

4.1 Introduction

Even if most activity recognition systems rely on supervised learning [38, 39], its applicability to detect complex ADLs (e.g., cooking, cleaning, and dressing) is questionable. On the one side, the way in which individuals perform ADLs strongly depends on current context conditions. Hence, a large dataset of ADLs should be acquired to capture most execution patterns in different situations. On the other side, activity execution patterns are strongly coupled to the individual's characteristics and home environment, and the portability of activity datasets is an open issue [61]. As a consequence, ideally one extensive ADLs dataset should be acquired from each monitored individual. Unfortunately, acquiring ADLs datasets is very expensive in terms of annotation costs [103, 104]. Besides, activity annotation by an external observer, by means of cameras or direct observation, violates the user's privacy. Indeed, to overcome that problem many other works relied on knowledge-based activity models, manually specified through logic languages and ontologies. Those models are matched with acquired sensor data to recognize the activities [62, 64, 105]. However, the main shortcoming of that approach relies in the rigidity of specifications. For instance, complex ADLs are often specified through temporal sequences of simpler actions [98]. Nevertheless, it is unfeasible to enumerate all the possible sequences of actions describing a complex ADL.

In this chapter, we propose a method to overcome the limitations of both approaches. First, our method is unsupervised: we do not need to acquire expensive activity datasets. Note that the term *unsupervised* does not refer to unsupervised learning, but we use it to highlight that our method does not require a training set. Second, our activity model is based on general semantic relations among activities and smart-home infrastructure; hence, we can seamlessly reuse our model with different individuals and in different environments. Specifically, we defined an OWL 2 [106] ontology to formally model the smart home environment and the semantics of activities. We rely on ontological reasoning to derive necessary conditions about the sensor events that must occur during the execution of a specific activity in the current environment. This also enables to extract *semantic correlations* among fired sensor events and executed ADLs. Based on the semantic correlations, a statistical algorithm pre-processes sensor events to identify *candidate* activity instances, i.e., initial hypotheses about the start and end time of occurred activities. Finally, we translate our ontological model in a Markov Logic Network (MLN) [28], and perform probabilistic reasoning to refine candidate activity instances and check their consistency.

The combination of specification-based and probabilistic approaches has also been investigated in other fields of Artificial Intelligence [107]. However, our method supports the recognition of interleaved activities, while most existing techniques are restricted to sequential ones. We target the recognition of interleaved activities explicitly by considering this aspect in our MLN model, where sensor events can be assigned to overlapping activity instances. Some methods analyze textual descriptions of activities mined from the Web in order to obtain correlations among used objects and activities [108, 109]. In our work, we mine not only correlations, but also necessary conditions about sensor events that must be observed during the activity execution. Moreover, we derive correlations and necessary conditions considering the actual environment where activities are executed, while methods based on Web mining derive generic correlations. An unsupervised method that is close to our approach has been proposed by Ye et al. [75], where ontologies are used to derive semantic similarity between sensor events. This similarity is used to segment sensor data, obtaining sequential activities' patterns used to train a clustering model. With respect to that work, our method is totally independent of the data and it also considers interleaved activities. Further, probabilistic description logics have been used to recognize ADLs considering the variability of activity execution [98]. However, those works rely on rigid assumptions about the simpler constituents of activities. Hence, while the specification-based approach is effective for activities characterized by a few typical execution patterns, it is hardly scalable to the comprehensive specification of complex ADLs in different contexts. On the contrary, in this work we rely on general semantic relations among activities and smart-home infrastructure, which are fine-tuned to the current context.

We performed extensive experiments with real-world datasets of ADLs performed by 22 individuals in two different smart home environments. Results show that, even using a smaller number of sensors, the performance of our unsupervised method is comparable to the one of existing methods that rely on labeled activity datasets.

This chapter is structured as follows. Section 4.2 introduces some preliminary notions which are useful to understand the proposed technique. A general overview of our system is introduced in Section 4.3. The ontological and probabilistic reasoning methods are respectively in Section 4.4 and Section 4.5. The experimental evaluation of our method on two different datasets is discussed in Section 4.6. Finally, Section 4.7 concludes the chapter.

4.2 Preliminaries

In this section, we introduce preliminary notions about description logics and Markov Logic Networks.

4.2.1 Description logics and formal ontologies

In computer science, description logics (DLs) [65] have emerged as the state-of-the-art formalism to represent *ontologies*. These enable to formally define concepts of a domain of interest, their properties, and the relationships among concepts. In this work, we use an ontology to formally define the semantics of activities, sensor events, and context data. Moreover, DLs support ontological reasoning, which allows to verify the consistency of the knowledge base, and to infer additional information from existing facts. The formalism of choice is typically OWL 2 [106]. A knowledge engineer can model the domain of interest by means of classes, individuals, properties of individuals, and relationships among individuals. Several operators can be used to declare complex definitions based on simpler ones, including operators for conjunction, disjunction, negation, and universal and existential quantification. For instance, the activity PREPARINGHOTMEAL can be defined based on the definitions of PREPARINGMEAL and PREPARINGCOLDMEAL:

$$\textsc{PreparingHotMeal} \equiv \textsc{PreparingMeal} \sqcap \\ \neg\textsc{PreparingColdMeal}$$

In this work, we also exploit the following operators:

- *Qualified cardinality restriction* restricts the class membership to those instances that are in a given relation with a minimum or maximum number of other individuals of a given class. For instance, the following axiom states that the activity *"Preparing hot meal"* requires the use of at least one instrument to cook food:

$$\textsc{PreparingHotMeal} \sqsubseteq \textsc{Activity} \sqcap \\ \geq 1\, \textsc{requiresUsageOf}.\textsc{CookingInstrument}$$

- *Composition of properties.* OWL 2 supports a restricted form of property composition $\circ$. For instance, the following axiom states that if a person is in a given apartment, and she is executing a given activity, then that activity is executed in that apartment:

$$\textsc{executesAct}^{-} \circ \textsc{isInLocation} \sqsubseteq \textsc{actIsExecutedInLocation}$$

  Note that EXECUTESACTIVITY^{-} denotes the inverse of EXECUTESACTIVITY.

Formally, a DLs knowledge base is composed by a pair $\langle \mathcal{T}, \mathcal{A} \rangle$. The *TBox* $\mathcal{T}$ constitutes the terminological part of the knowledge base. The TBox is composed of a set of axioms $C \sqsubseteq D$ or $P \sqsubseteq R$ (*inclusions*) and $C \equiv D$ or $P \equiv R$ (*equality*), where C and D are classes, and P and R are object properties. An axiom $C \sqsubseteq D$ is satisfied by an interpretation $\mathcal{I}$ when $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. An interpretation $\mathcal{I}$ satisfies a TBox $\mathcal{T}$ when $\mathcal{I}$ satisfies all the axioms of $\mathcal{T}$.

The *ABox* $\mathcal{A}$ is the assertional part of the knowledge base. The ABox is composed of a set of axioms of the form $x : C$ and $\langle x, y \rangle : R$, where x and y are individuals, C is a class, and R is an object property. For instance,

$$\text{MARY} : \text{ELDERLYPERSON}$$

denotes that Mary is an elderly person and

$$\langle \text{MARY}, \text{APARTMENT}23 \rangle : \text{LIVESIN}$$

represents that Mary lives in Apartment23. Axioms $x : C$ and $\langle x, y \rangle : P$ are satisfied by an interpretation $\mathcal{I}$ when $x^{\mathcal{I}} \in C^{\mathcal{I}}$ and $\langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in P^{\mathcal{I}}$, respectively. An interpretation $\mathcal{I}$ satisfies an ABox $\mathcal{A}$ when $\mathcal{I}$ satisfies all the axioms of $\mathcal{A}$. An interpretation $\mathcal{I}$ that satisfies both the TBox $\mathcal{T}$ and the ABox $\mathcal{A}$ is called a *model* of $\langle \mathcal{T}, \mathcal{A} \rangle$. DLs support several reasoning tasks. In particular, we rely on the following ones:

- *Satisfiability:* a class C is satisfiable with respect to a TBox $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}$ is non empty. We execute this reasoning task to check the consistency of our ontological model.
- *Property fillers retrieval:* retrieving all the instances in $\mathcal{A}$ that are related to a given individual with respect to a given property. We execute this reasoning task to derive semantic correlations among activities and sensor events.

4.2.2 Markov Logic with numerical Constraints

In addition to purely logical or probabilistic approaches, a Markov Logic Network provides many benefits and allows to handle uncertainty, imperfection, and contradictory knowledge. These characteristics make it an appealing tool to reason with sensor data and ADLs. Technically, a Markov Logic Network (MLN) $\mathcal{M}$ is a finite set of pairs $(F_i, w_i), 1 \leq i \leq n$, where each F_i is an axiom in function-free first-order logic and $w_i \in \mathbb{R}$ [28]. Together with a finite set of constants $C = \{c_1, ..., c_n\}$ it defines the *ground* MLN $\mathcal{M}_C$, i.e., the MLN in which axioms do not contain any free variables. This comprises one binary variable for each grounding of F_i with weight w_i. Hence, a MLN defines a log-linear probability distribution over Herbrand interpretations (possible worlds)

$$P(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(\mathbf{x})\right) \tag{4.1}$$

where $n_i(\mathbf{x})$ is the number of satisfied groundings of F_i in the possible world $\mathbf{x}$ and Z is a normalization constant.

In this work, we rely on an extended version of MLN which includes numerical constraints, also denoted as $\mathsf{MLN_{NC}}$ [110, 111]. We use this extension to reason on the temporal domain of activities and sensor events. The constraints are predicates of the form $\theta \bowtie \psi$, where θ and ψ denote variables, numerical constants, or algebraic expressions (that might contain elementary operators). In this context, the binary operator $\bowtie$ returns a truth value under a particular grounding.

Definition 4.2.1 ($\mathsf{MLN_{NC}}$)**.** A numerical constraint NC is composed of numerical constants (e.g., elements of $\mathbb{N}$, $\mathbb{I}$), variables, elementary operators or functions ($+$, $*$, $-$, $\%$, $\sqrt{\ }$), standard relations ($>, <, =, \neq, \geq, \leq$), and Boolean operators ($\wedge, \vee$). An $\mathsf{MLN_{NC}}$ is a set of pairs (FC_i, w_i) where FC_i is a formula in first-order logic that may contain a NC and w_i is a real number representing the weight of FC_i.

Example 4.2.1. Using $\mathsf{MLN_{NC}}$ it is possible to represent the axiom: two events "turning on the oven" cannot belong to the same instance of meal preparation if their temporal distance is more than two hours:
$\{\forall\ se_1,\ se_2,\ ai_1,\ ai_2,\ t_1,\ t_2 :\ event(se_1,\ 'oven',\ t_1) \wedge$
$event(se_2, 'oven', t_2) \wedge occursIn(se_1, ai_1) \wedge occursIn(se_2, ai_2)$
$\wedge\ \mathsf{NC}(t_1, t_2) \Rightarrow ai_1 \neq ai_2, \mathsf{NC}(t_1, t_2) = |t_1 - t_2| > 120\}$.

Maximum a posteriori (MAP) inference is the task of finding the most probable world given some observations also referred to as evidence. Given the observed variables $E = e$, the MAP problem aims to find an assignment of all non-evidence (hidden) variables $X = x$ such that $\mathbf{I} = \underset{x}{\operatorname{argmax}}\ P(X = x \mid E = e)$. Based on the MLN of sensor events and semantic constraints, we apply MAP inference to derive the most probable activities. In the following, we outline our model, architecture, and the individual components.

4.3 Model and system overview

4.3.1 Notation

In this chapter we adopt a similar but slightly different notation with respect to the one presented in Chapter 3. We consider $\mathbf{A} = \{ac_1, ac_2, \ldots, ac_k\}$ as the set of activity classes. Further, an instance ai_i of an activity class $ac_j \in \mathbf{A}$ represents the occurrence of ac_j during a given timespan. The activity instance is associated to the operations executed to perform it, where the start and end time of instances of different activities can overlap. We denote $\mathbf{E}$ as the set of pre-processed event types that correspond to the set of monitored operations (e.g., $\mathbf{E} = \{$ *opening_the_fridge*, *closing_the_fridge* $\}$). In addition, $\mathbf{T}$ describes the set of all possible event timestamps. An event is defined as $ev(se, et, t)$, where se is the identifier of the instance of an event with type $et_i \in \mathbf{E}$ occurred at timestamp $t_i \in \mathbf{T}$.

4.3.2 Ontological model

Figure 4.1 illustrates an excerpt of our ontology, which models a complete home environment. In addition, it also covers axioms for each activity class that describe

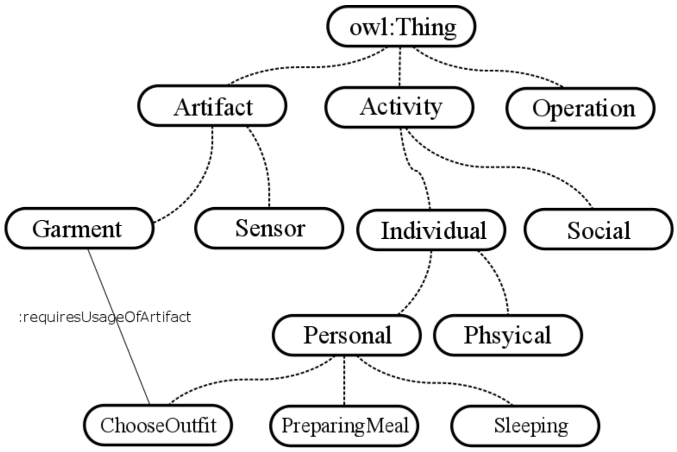


Figure 4.1: Excerpt of our ontology. The dashed lines represent a *subClassOf* relation where the upper is the parent of the lower class. In addition, the individual classes have relations that describe dependencies.

dependencies and conditions. In particular, we express necessary conditions for a set of operations to be generated by an instance of that class, according to the activity semantics. For example, the operations generated by an instance of *preparing hot meal* must include an operation *using a cooking instrument*. The ontology also models sensors and the operation that they detect; e.g., a power sensor attached to the electric stove detects the operation *turning on the stove*. In turn, this operation is a subclass of *using a cooking instrument*. The ontology carefully describes these kind of relations and, through ontological reasoning, we can derive constraints like the following: "since the stove is the only cooking instrument in the home, and a sensor is available that detects the usage of the stove, then each instance of *preparing hot meal* executed in the home must necessarily generate an event from that sensor".

Besides, other necessary conditions regard time and location. This includes constraints on the duration of the activity instance, and dependencies between activity and location. As explained in the next section, ontological reasoning is also used to infer probabilistic dependencies among sensor event types and classes of executed activities; we denote them as *semantic correlations*. Our ontology is publicly available[1].

4.3.3 Architecture

Figure 4.2 shows an overview of our system. The smart-home monitoring system

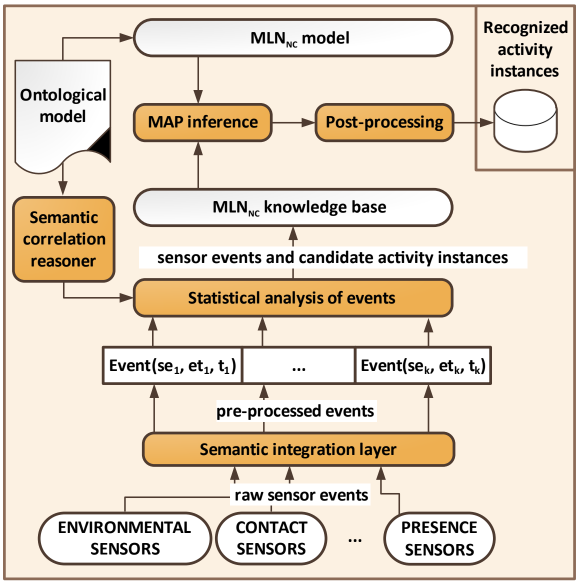


Figure 4.2: System overview.

collects raw events data from the sensor network, including environmental, presence, and contact sensors. The SEMANTIC CORRELATION REASONER performs ontological reasoning to derive semantic correlations among event types and activity classes; e.g., "the event type *UseStove* is strongly related to *PreparingHotMeal* and unrelated to *PreparingColdMeal*". Those correlations are used by the module for STATISTICAL ANALYSIS OF EVENTS to identify *candidate* activity instances,

[1] `http://sensor.informatik.uni-mannheim.de/#results2016unsupervised`

which are then refined by the MLN_{NC} reasoner. In particular, the events as well as the candidate activity instances are used to populate the assertional part of the MLN_{NC} knowledge base. The ontological model of considered activities and events is translated into the MLN_{NC} model. Periodically (e.g., at the end of each day), MAP inference is performed to assign each event to the candidate activity instance that most probably generated it, according to semantic correlations and ontological constraints. Finally, the output of MAP inference is post-processed to detect the exact start and end time of occurred activity instances.

4.4 Ontological reasoning

In the following, we introduce a simple running example to illustrate our approach.

Example 4.4.1. Suppose to monitor three activities in a smart home: preparing hot meal, preparing cold meal, and preparing tea. The home contains: one silverware drawer, one stove, and one freezer, each equipped with a sensor to detect its usage. No training set of activities is available. How can we exploit semantic reasoning to recognize the activities?

In the following of this section, we explain how we answer the above question.

4.4.1 Semantic correlation reasoner

The specific objective of this reasoner is to compute the degree of correlation among sensor events and the activities performed in the home. As illustrated in the axioms below, in our ontology, artifacts are organized in a hierarchy. The class STOVE is a sub-class of cooking instruments, used in the apartment to prepare hot meal or tea, where FREEZER is a DEVICE used to prepare hot or cold meal. SILVERWAREDRAWER belongs to FOODPREPFURNITURE and is used for the three activities. The instance $\{\text{APT}\}$ represents the current apartment. For clarity, we represent the name of ontological instances within curly brackets.

$$\begin{aligned} \textsc{Stove} \sqsubseteq \textsc{CookingInstrument} \sqcap \\ \Big(\exists\, \textsc{usedFor}.\big((\textsc{PrepHotMeal} \sqcup \textsc{PrepTea}) \sqcap \\ (\exists\, \textsc{occursIn}.\{\textsc{apt}\})\big)\Big). \end{aligned}$$

$$\textsc{Freezer} \sqsubseteq \textsc{Device} \sqcap \Big(\exists\, \textsc{usedFor}.\big((\textsc{PrepHotMeal} \sqcup \textsc{PrepColdMeal}) \sqcap (\exists\, \textsc{occursIn}.\{\textsc{apt}\})\big)\Big).$$

$$\textsc{SilverwareDrawer} \sqsubseteq \textsc{FoodPrepFurniture}.$$

$$\textsc{FoodPrepFurniture} \sqsubseteq \textsc{Furniture} \sqcap \Big(\exists\, \textsc{usedFor}.\,\big((\textsc{PrepTea} \sqcup \textsc{PrepColdMeal} \sqcup \textsc{PrepHotMeal}) \sqcap (\exists\, \textsc{occursIn}.\{\textsc{apt}\})\big)\Big).$$

Based on the smart home setup, we instantiate the ontology with the sensors and artifacts in the apartment, and we specify which activities we want to monitor.

Example 4.4.2. The activities that we want to monitor are $\{\textsc{ac\_prep\_cold\_meal}\}$, $\{\textsc{ac\_prep\_hot\_meal}\}$ and $\{\textsc{ac\_ prep\_tea}\}$. They are instances representing the generic occurrences of $\textsc{PrepColdMeal}$, $\textsc{PrepHotMeal}$, and $\textsc{PrepTea}$, respectively. Lines 5 and 6 state that at most one instance of each activity type can be monitored at a time. Further, lines 7 and 8 represent that the $\{\textsc{apt}\}$ contains exactly one cooking instrument, one silverware drawer, and a freezer:

$$\{\textsc{apt}\} = \textsc{Apartment} \tag{4.1}$$
$$\sqcap\, (\exists\, \textsc{monitAct}.(\{\textsc{ac\_prep\_cold\_meal}\})) \tag{4.2}$$
$$\sqcap\, (\exists\, \textsc{monitAct}.(\{\textsc{ac\_prep\_hot\_meal}\})) \tag{4.3}$$
$$\sqcap\, (\exists\, \textsc{monitAct}.(\{\textsc{ac\_prep\_tea}\})) \tag{4.4}$$
$$\sqcap\, (\leq\, 1\textsc{monitAct}.\textsc{PrepColdMeal}) \tag{4.5}$$
$$\sqcap\, (\leq\, 1\textsc{monitAct}.\textsc{PrepHotMeal}) \sqcap (\leq\, 1\textsc{monitAct}.\textsc{PrepTea}) \tag{4.6}$$
$$\sqcap\, (=\, 1(\textsc{isIn})^{-}.\textsc{CookingInstrument}) \tag{4.7}$$
$$\sqcap\, (=\, 1(\textsc{isIn})^{-}.\textsc{SilverwareDrawer}) \sqcap (=\, 1(\textsc{isIn})^{-}.\textsc{Freezer}). \tag{4.8}$$

Subsequently, we introduce an instance in the ontology for each artifact in the apartment:

$$\{\textsc{stove}\} \equiv \textsc{Stove} \sqcap \exists\, \textsc{isIn}.\{\textsc{apt}\}.$$

$$\{\textsc{freezer}\} \equiv \textsc{Freezer} \sqcap \exists\, \textsc{isIn}.\{\textsc{apt}\}.$$

$$\{\textsc{silverware\_drawer}\} \equiv \textsc{SilverwareDrawer} \sqcap \exists\, \textsc{isIn}.\{\textsc{apt}\}.$$

We also instantiate each sensor that occurs in our apartment:

$$\{\textsc{s\_stove}\} \equiv \textsc{PowerSensor} \sqcap (\exists\, \textsc{sensesUsageOf}.\{\textsc{stove}\}) \sqcap (\exists\, \textsc{producesEvent}.\{\textsc{et\_stove}\}).$$

$$\{\text{S\_SILVERWARE\_DRAWER}\} \equiv \text{CONTACTSENSOR}$$
$$\sqcap (\exists\, \text{SENSESUSAGEOF}.\{\text{SILVERWARE\_DRAWER}\})$$
$$\sqcap (\exists\, \text{PRODUCESEVENT}.\{\text{ET\_SILVERWARE\_DRAWER}\}).$$

$$\{\text{S\_FREEZER}\} \equiv \text{CONTACTSENSOR}$$
$$\sqcap (\exists\, \text{SENSESUSAGEOF}.\{\text{FREEZER}\})$$
$$\sqcap (\exists\, \text{PRODUCESEVENT}.\{\text{ET\_FREEZER}\}).$$

According to the introduced axioms, {S_STOVE} is an instance of POWERSENSOR which senses the usage of {STOVE} and produces a generic event of type {ET_STOVE}. Similarly, the last two axioms define sensors and events for the silverware drawer and the freezer, respectively.

We exploit the property composition operator to infer the semantic correlations between sensor events and activity types. In particular, we use the following axiom, which states that: "if an event of type et is produced by a sensor that detects the usage of an artifact possibly used for an activity of class ac, then et is a *predictive sensor event type* for ac":

$$\text{PRODUCESEVENT}^{-} \circ \text{SENSESUSAGEOF} \circ$$
$$\text{USEDFOR} \rightarrow \text{PREDICTIVESENSOREVENTFOR}$$

Then, we perform ontological reasoning to infer the fillers of property PREDICTIVESENSOREVENTFOR, and use them to compute semantic correlations.

Example 4.4.3. Considering all of the introduced axioms, the OWL 2 reasoner infers that:

- {ET_STOVE} *is a predictive sensor event type for* {AC_ PREPARE_HOT_MEAL} *and* {AC_PREP_TEA}.
- {ET_SILVERWARE_DRAWER} *is a predictive sensor event type for* {AC_PREP_HOT_MEAL}, {AC_PREP_COLD_MEAL } *and* {AC_PREP_TEA}.
- {ET_FREEZER} *is a predictive sensor event type for* {AC_PREP_HOT_MEAL} *and* {AC_PREP_COLD_ MEAL}.

We represent semantic correlations using a *prior probability matrix* (PPM). The rows correspond to the activity classes, where the columns to the sensor event types. Hence, $PPM(ac, et)$ stores the probability of an event of type et being generated by an activity of class ac. If a given sensor event type is predictive of a single activity class, the value of the corresponding entry is 1; if it is predictive of multiple activity classes, the value is uniformly distributed among them. The prior probability matrix resulting from our running example is shown in Table 4.1. The PPM is given as input to the STATISTICAL ANALYSIS OF EVENTS module.

	{**et_stove**}	{**et_silverware_drawer**}	{**et_freezer**}
{**ac_prep_hot_meal**}	0.5	$0.\overline{33}$	0.5
{**ac_prep_cold_meal**}	0.0	$0.\overline{33}$	0.5
{**ac_prep_tea**}	0.5	$0.\overline{33}$	0.0

Table 4.1: Prior probability matrix of our running example.

4.4.2 Deriving necessary sensor observations

Our ontology includes a property REQUIRESUSAGEOFARTIFACT, which associates artifacts in the apartment with activities for which they are necessary.

Example 4.4.4. Continuing our running example, the axiom below defines PREPHOTMEAL as a subclass of PREPAREMEAL that requires the usage of a cooking instrument:

$$\text{PrepHotMeal} \sqsubseteq \text{PrepareMeal} \sqcap \exists \text{requiresUsageOfArtifact}.\\ (\text{CookingInstrument} \sqcap (\exists \text{isIn}.\{\text{apt}\})).$$

Thus, we infer which sensor events must necessarily be observed during the execution of an activity. The following axiom states that: "if an event of type et is produced by a sensor that detects the usage of an artifact required for executing an activity of class ac, then et is a *necessary sensor event type* for each activity instance of class ac".

$$\text{producesEvent}^{-} \circ \text{sensesUsageOf} \circ\\ \text{requiresUsageOf}^{-} \rightarrow \text{necessaryEventFor}.$$

Then, we infer the fillers of property NECESSARYEVENTFOR through ontological reasoning, translate them in MLN_{NC} axioms, and add them to the MLN_{NC} model.

Example 4.4.5. Given the introduced axioms, in this case the OWL 2 reasoner infers that $\{$ET_STOVE$\}$ is a necessary sensor event type for $\{$AC_PREP_HOT_MEAL$\}$. Indeed, ET_STOVE is produced by usage of STOVE, which is the only instance of COOKINGINSTRUMENT available in the home.

4.5 Recognizing activity instances

At first, we identify activity instance *candidates* and consider them as part of our MLN_{NC} knowledge base (KB). The KB also includes observed sensor events and computed semantic correlations. Then, MAP inference enables us to assign each activity instance to its most probable class, and each event to its most probable activity instance. Figure 4.3 depicts our MLN_{NC} model, where we distinguish between *observed* (star symbol) and *hidden* predicates.

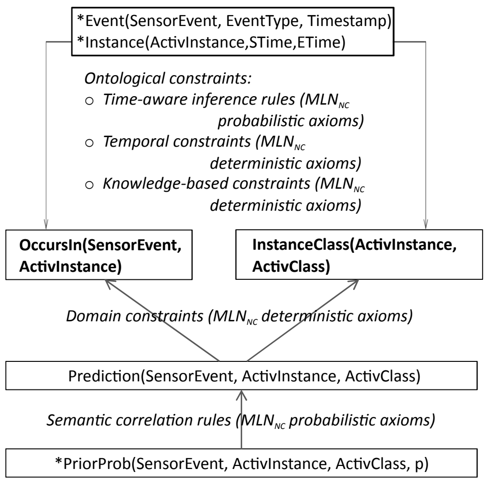


Figure 4.3: Probabilistic activity recognition framework. The arrows indicate the relations and dependencies between the depicted observed and hidden predicates.

Observed predicates represent knowledge facts, where the instances of hidden predicates are computed by MAP inference. In the following, we explain the different components of our framework in detail.

4.5.1 Statistical analysis of events

Candidate activity instances are computed by a heuristic algorithm, shown in Algorithm 3, which implements the STATISTICAL ANALYSIS OF EVENTS module. The algorithm iterates over all temporally ordered events provided by the SEMANTIC INTEGRATION layer. It considers the PPM matrix of semantic correlations to infer, for each sensor event se, the most probable activity class ac generating it. The corresponding timestamp of the event and the resulting activity class enables us to formulate initial hypotheses about the occurred activity instances. If an activity instance ai of class ac exists, whose boundaries (start and end time) are temporally close to se according to an activity-dependent threshold $maxGap_{ac}$, then se is assigned to ai. Otherwise, a new instance of class ac is created, and se is assigned to it. The boundaries of each instance are respectively represented by the first and the last event of the instance.

ALGORITHM 3: Statistical analysis of events

Input: Sensor events $\{event(se_0, et_0, t_0), \ldots, event(se_n, et_n, t_n)\}$,
prior probability matrix PPM
Output: Candidate activity instances $\{i_0, i_1, \ldots, i_{m-1}\}$
$instances \leftarrow \emptyset$
for each $event(se, et, t) \in X$ **do**
  $ac \leftarrow$ activity class with max correlation with et according to PPM
  $ai \leftarrow$ activity instance in $instances$ of class ac closest to se
  if ai exists **and** t is temporally close to ai according to $maxGap_{ac}$ **then**
    assign $event(se, et, t)$ to ai
  else
    $ai \leftarrow$ a new instance of class ac
    assign $event(se, et, t)$ to ai
    $instances \leftarrow instances \bigcup \{ai\}$
  end if
end for
return $instances$

4.5.2 MLN modeling

Semantic correlations are modeled through predicates *PriorProb*, *Event*, and *Instance*. The *PriorProb* predicate represents correlations among sensor events and activities:

$$*PriorProb(SensorEvent, ActivInstance, ActivClass, p)$$

Hence, it describes the probability p that a given sensor event se corresponds to a given activity instance ai of an activity class ac. The probability relies on the semantic correlation between the event type et and the activity class ac (PPM), and also depends on the temporal distance between the sensor event se and the boundaries of the activity instance ai.

Formally, given an activity instance ai of class ac with start time t_{st} and end time t_{ed}, and a sensor event se of type et and timestamp t, the probability p of **PriorProb*(se, ai, ac, p) is computed by the following function:

$$p = \begin{cases} PPM(ac, et) & \text{if } t_{ed}\text{-}MaxGap_{ac} \leq t \leq t_{st}\text{+}MaxGap_{ac} \\ 0 & \text{otherwise} \end{cases}$$

Each sensor event is represented by an instance of the predicate *Event*, which represents the sensor event, its type, and its timestamp:

$$*Event(SensorEvent, EventType, Timestamp)$$

Candidate activity instances computed by Algorithm 3 are represented by the predicate *Instance* which models the relation between the activity instance, its start time, and end time:

$$*Instance(ActivInstance, STime, ETime)$$

The instantiated predicates, derived from the activity instances and the recorded sensor events, are added as facts to our MLN_{NC} knowledge base.

4.5.3 Hidden predicates and domain constraints

Beside the observed predicates, the model also comprises a set of hidden predicates, which can be considered our target classes: *Prediction*, *OccursIn*, and *InstanceClass*. The predicate *Prediction* represents the predicted assignment of a sensor event to an activity instance of a given class:

$$Prediction(SensorEvent, ActivInstance, ActivClass)$$

In addition, the other two predicates are used to express domain constraints about the consistency of inferred activity instances:

$$OccursIn(SensorEvent, ActivInstance)$$
$$InstanceClass(ActivInstance, ActivClass)$$

In particular, the following domain constraint states that each sensor event occurs in exactly one activity instance:

$$|ai|OccursIn(se, ai) = 1,$$

while the following one states that each activity instance belongs to exactly one activity type:

$$|ac|InstanceClass(ai, ac) = 1.$$

4.5.4 Semantic correlation rules

The relations between the observed and hidden predicates are modeled by probabilistic axioms. As illustrated in Figure 4.3, the hidden predicate *Prediction* is derived from *PriorProb*:

$$conf : *PriorProb(se, ai, ac, conf) \Rightarrow Prediction(se, ai, ac).$$

Thus, the confidence value describes the probability that a sensor event is assigned to an activity instance of a given class. In turn, the remaining hidden predicates are derived from the hidden *Prediction* predicate. The corresponding probabilistic axioms are the following:

$$Prediction(se, ai, ac) \Rightarrow OccursIn(se, ai),$$
$$Prediction(se, ai, ac) \Rightarrow InstanceClass(ai, ac).$$

Note that the above rules are subject to the domain constraints introduced before.

4.5.5 Knowledge-based constraints

Knowledge-based constraints enable us to express conditions about the occurrence (or non-occurrence) of sensor events of a given type during the occurrence of an activity instance.

Example 4.5.1. The constraint "each activity instance of type 'preparing hot meal' must be associated to an event of type 'UseStove' " is logically expressed by the rule:

$$InstanceClass(ai,\ \text{"}PreparingHotMeal\text{"}) \Rightarrow \exists\ se,\ t\ :$$
$$OccursIn(se,\ ai) \wedge *Event(se,\ \text{"}UseStove\text{"},\ t).$$

Knowledge-based constraints are automatically derived from the fillers of the NECESSARYEVENTFOR OWL 2 property obtained from ontological reasoning as already mentioned.

4.5.6 Temporal constraints

We model MLN_{NC} temporal constraints regarding the duration and the distance of events or activities. We consider two kinds of temporal constraints:

1) *Temporally close events (e.g., whose temporal distance is below Δ seconds) likely belong to the same activity instance.* We express this soft constraint through these axioms:

$$\forall\, t_1,\, t_2\, :\, (|t_1 - t_2| < \Delta) \,\Rightarrow\, tClose(t_1,\, t_2)$$

$$\begin{gathered} w\;Event(se_1,\, et_1,\, t_1) \,\wedge\, Event(se_2,\, et_2,\, t_2) \wedge \\ tClose(t_1,\, t_2) \wedge OccursIn(se_1,\, ai) \,\Rightarrow\, OccursIn(se_2,\, ai) \end{gathered}$$

The latter is a probabilistic axiom whose weight w is chosen experimentally.

2) *Constraints on typical duration of each activity (e.g., "showering cannot last more than Δ' minutes").* We express these constraints either through probabilistic or deterministic axioms, according to the characteristics of the considered activity. Indeed, the variance of the duration of certain activities (e.g., showering) is relatively small, while it is larger for other activities (e.g., preparing dinner). The duration of the former is modeled with deterministic axioms where probabilistic ones are used for the latter. The axioms below state that an instance of "showering" cannot last more than Δ' minutes:

$$\forall\, t_1,\, t_2\, :\, (|t_1 - t_2| < \Delta') \,\Rightarrow\, tclose\_showering(t_1,\, t_2)$$

$$\begin{gathered} InstanceClass(ai, \text{“}Showering\text{”}) \,\wedge\, OccursIn(se_1, ai) \wedge \\ OccursIn(se_2, ai) \,\wedge\, Event(se_1, et_1, t_1) \wedge \\ Event(se_2, et_2, t_2) \,\Rightarrow\, tclose\_showering(t_1,\, t_2) \end{gathered}$$

4.5.7 Time-aware inference rules

Finally, as explained before, the semantics of some simple activities is naturally expressed in our ontology based on the typical actions composing them. Hence, we apply rules that express the relation of specific operations derived from sensor events in context of time. Consider the following example:

Example 4.5.2. A typical pattern of operations for watering plants consists in (1) "getting water" and (2) "moving to the plants" shortly after. We express this activity inference pattern through the MLN_{NC} axioms below:

$$\begin{aligned}
&Event(se_1,\, \text{“}water_sensor\text{”},\, t_1)\\
&\wedge Event(se_2,\, \text{“}plant_presence_sensor\text{”},\, t_2) \,\wedge\, t_1 < t_2\\
&\wedge tclose_waterplants(t_1,\, t_2) \,\Rightarrow \exists\, ai\, :\\
&InstanceClass(ai,\, \text{“}WaterPlants\text{”})\\
&\wedge occursIn(se_1,\, ai) \,\wedge\, occursIn(se_2,\, ai).
\end{aligned}$$

4.5.8 Inference of activity instances and temporal boundaries

In order to infer activity instances, their class, and corresponding sensor events, we execute MAP inference on the presented MLN_{NC} model. The output of MAP inference is the most probable assignment of (i) sensor events to activity instances (i.e., fillers of the OccursIn predicate), and (ii) activity classes to activity instances (i.e., fillers of the InstanceClass predicate). Since computing the start and end time of activity instances within MLN_{NC} reasoning would be unnecessarily complicated, we post-process the result of MAP inference to detect the temporal boundaries of each activity instance ai:

$$\begin{aligned}
STime(ai) &= \min\{t \,:\, \exists Event(se, et, t) \wedge OccursIn(se, ai)\},\\
ETime(ai) &= \max\{t \,:\, \exists Event(se, et, t) \wedge OccursIn(se, ai)\}.
\end{aligned}$$

4.6 Experimental evaluation

In the following, we present our experimental setup and results. Unless otherwise specified, the presented results rely on the introduced unsupervised approach, where the semantic correlations (PPM matrix) were derived from ontological reasoning. To evaluate the effectiveness of semantic correlations extracted with our method, we also performed experiments computing the PPM from the dataset; more precisely, based on the frequency of the sensors types produced by the different activities. We denote by MLN_{NC} (Ontology) the former method, and by

MLN_{NC} (Dataset) the latter. We use the well-known dataset of Cook et al. [112, 113], named *CASAS*, and the real-home dataset introduced in Chapter 3. Both datasets include interleaved activities in a smart-home environment. To provide the possibility to reconstruct our approaches and experiments, we provide a REST API and web interface which is publicly available[2] and supports the MLN_{NC} solver.

4.6.1 CASAS Dataset

The CASAS dataset covers interleaved ADLs of twenty-one subjects acquired in a smart home laboratory. Sensors collected data about movement, temperature, use of water, interaction with objects, doors, phone; 70 sensors were used in total. Eight activities were considered: *fill medication dispenser* (ac_1), *watch DVD* (ac_2), *water plants* (ac_3), *answer the phone* (ac_4), *prepare birthday card* (ac_5), *prepare soup* (ac_6), *clean* (ac_7), and *choose outfit* (ac_8). The order and expenditure of time were up to the subject and it was allowed to perform the activities in parallel. During the data collection only one single person was present in the smart home. With our MLN_{NC} (Ontology) method, only 25 out of 70 sensors were used. Indeed, the semantic correlation reasoner excluded the remaining 45 (mostly movement sensors), since they had no significant correlation with the considered activities.

During this experiment, we evaluated how well the considered sensor events could be assigned to the corresponding activity instance, but also the quality of detected activity boundaries. Table 4.2 shows that our method outperforms the HMM approach used in [112] in assigning each sensor event to the activity instance that generated it. We observe that we recognize each activity at least equal or better than HMM, except *Clean*. The poor performance in recognizing *Clean* is due to the fact that, in the CASAS dataset, it is characterized by different movement patterns that are only partially captured by our method, especially when semantic correlations are extracted from the ontology. Considering the other activities, the PPM generated by ontological reasoning obtains essentially the same performance of the one extracted from the dataset, confirming the effectiveness of our semantic correlation reasoner.

Focusing on the other activities, the experiments show that the interactions with

[2] `http://executor.informatik.uni-mannheim.de`

Table 4.2: CASAS dataset: Results ($\mathbf{F_1}$ measure) of the proposed activity recognition method compared to related work for interleaved activities.

Class	**HMM** [112] **(time-shifted)**	$\mathbf{MLN_{NC}}$ **(Dataset)**	$\mathbf{MLN_{NC}}$ **(Ontology)**
ac_1	0.656	0.803	**0.848**
ac_2	0.862	**0.882**	0.811
ac_3	0.285	**0.740**	0.720
ac_4	0.589	0.688	**0.723**
ac_5	**0.828**	0.807	0.808
ac_6	0.826	0.873	**0.882**
ac_7	**0.881**	0.781	0.574
ac_8	0.673	**0.904**	0.882
$avg.$	0.700	**0.810**	0.781

objects are strong indicators of the performed activities. However, inspecting the recognition result in detail, we noticed a few cases in which subjects exhibited strange behaviors; e.g., prepared soup without water or took the phone but did not place a phone call. Especially the latter case is hard to recognize without further information. The former case is probably related to sensor errors. Figure 4.4 illustrates the individual results in more detail. It highlights that there are cases where

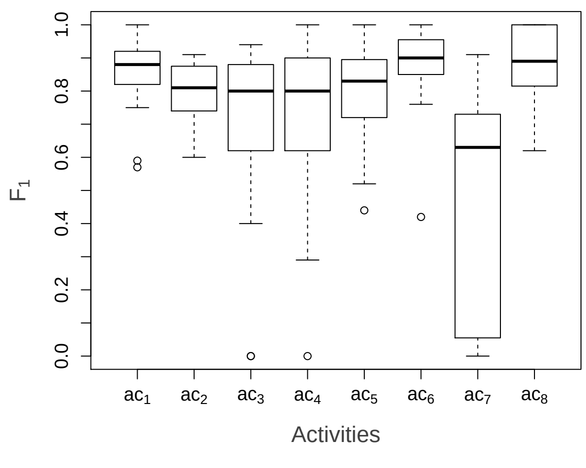


Figure 4.4: CASAS dataset: Detailed recognition results for each activity, aggregated over all subjects and represented by a box plot.

we could not recognize the activities *Answer the phone* and *Clean* at all, but in general the distribution is very similar and narrowed.

Considering the boundary detection method, the experiments show that preceding results and the quality of the detected boundaries for the individual activities are weakly related. Table 4.3 describes the deviation from the actual boundaries in detail. ΔStart is the average difference between the actual and predicted start of an activity instance in minutes. ΔDur is the average difference of actual and predicted duration. In context of the typical duration of each activity, the boundaries are well detected. Hence, the highest deviations are associated with the longest activities, and the overall results are acceptable for most applications.

Table 4.3: CASAS dataset: Results of boundary detection with MLN_{NC}. It shows the average deviation [min] of the candidate compared to the refined instances.

Class	**ΔStart (Candidate)**	**ΔStart (Refined)**	**ΔDur (Candidate)**	**ΔDur (Refined)**
ac_1	**0.670**	0.765	1.436	**0.890**
ac_2	**0.592**	**0.592**	**2.974**	3.140
ac_3	**0.075**	0.081	0.930	**0.829**
ac_4	**0.079**	**0.079**	**0.341**	0.422
ac_5	1.300	**1.079**	5.810	**4.642**
ac_6	1.617	**0.109**	4.077	**0.803**
ac_7	1.311	**0.692**	2.390	**2.249**
ac_8	**0.079**	0.097	1.300	**0.521**
$avg.$	0.727	**0.456**	2.424	**1.701**

When we compare the candidate instances and the refined (final) results obtained through MLN_{NC} reasoning, it strikes that our method refines the candidates reliably. Regarding *watch DVD* (ac_2) and *answer the phone* (ac_4), the refined duration increased slightly, because in some cases subjects took the phone well before using it, or turned on the DVD player well before watching a DVD. Besides, the low numbers clearly show that the duration of the different activities was in general short.

4.6.2 Real-home Dataset

We also considered the real-home dataset presented in Chapter 3. This dataset was acquired from an elderly woman diagnosed with Mild Cognitive Impairment, living alone in her apartment. Different environmental sensors (magnetic sensors, motion sensors, temperature sensors) have been used to monitor three ADLs for 55 days:

Taking medicines (ac_9), *Cooking* (ac_{10}), *Eating*. Moreover, activity *Others* (ac_{11}) was also labeled. Totally, 11 sensors were deployed. Our semantic correlation reasoner discarded 2 sensors among them, because they had no significant correlation with the considered activities. Compared to CASAS, this dataset was acquired in a fully naturalistic environment. Due to the cognitive decline of the subject, activities have been performed in many different and sometimes unexpected ways. Besides, the acquired data is also affected by noise due to various technical issues encountered during data acquisition [54]. Hence, the recognition of ADLs in this scenario is challenging, even if the number of considered activities is limited.

In order to be comparable with the results presented in Chapter 3 on the same dataset, we focused on activity instance classification. Table 4.4 shows the corresponding results and indicates that the accuracy achieved by our unsupervised method is comparable to the one achieved by the supervised method proposed in Chapter 3. However, we were unable to recognize *Eating*, because in the dataset it

Table 4.4: Real-home dataset: Results ($\mathbf{F_1}$ measure) of the proposed activity recognition method compared to our supervised approach.

Class	**Supervised** (Chapter 3)	$\mathbf{MLN_{NC}}$ **(Dataset)**	$\mathbf{MLN_{NC}}$ **(Ontology)**
ac_9	**0.946**	0.837	0.831
ac_{10}	**0.757**	0.669	0.752
ac_{11}	-	0.665	**0.702**

was only characterized by a single presence sensor close to the table, that was also used in context of the other activities. Hence, our semantic correlation reasoner did not find any sensor significantly correlated to *Eating*. Therefore, we decided to exclude that activity from the evaluation. On the other side, we were able to recognize *Others*, which was not previously considered. In particular, we consider as part of an *Others* activity instance each event in the dataset which is not annotated with a specific activity.

Inspecting the results, we notice that, with *Cooking*, our unsupervised method achieves essentially the same recognition rate of the supervised technique. With *Taking medicines*, the accuracy of our method is lower, mainly due to the absence of sensors strongly correlated to that activity. The accuracy of recognizing *Others* is in line with the one of the other activities. Note that we did not specify any rule for the *Others* activity, while the ontology derived that each event has a certain semantic correlation with that activity (i.e., any event can potentially occur outside the targeted activities). Considering the corresponding instance boundary results, Table 4.5 shows that, also with this dataset, MLN_{NC} refinement significantly improves the accuracy of predicted activity instances.

Table 4.5: Real-home dataset: Results of the boundary detection method. It shows the average deviation [min] of the candidate compared to the refined instances.

Class	**ΔStart (Candidate)**	**ΔStart (Refined)**	**ΔDur (Candidate)**	**ΔDur (Refined)**
ac_9	**2.199**	2.533	**1.084**	**1.084**
ac_{10}	14.437	**8.954**	25.833	**21.133**
ac_{11}	7.559	**3.255**	34.170	**16.590**
$avg.$	8.065	**4.914**	20.362	**12.936**

4.7 Summary

In this chapter, we proposed an unsupervised method to recognize complex ADLs through ontological and probabilistic reasoning. An ontology is in charge of inferring semantic correlations between sensor events and ADLs. Those correlations are then combined with sensor events collected in the home and processed by a Markov Logic Network to infer the most likely ADLs. The advantages of our method are: a) it does not require any training set, and b) the activity model is based on general relations among activities and smart-home infrastructure. Indeed, our model can be seamlessly reused with different individuals and in different environments. Moreover, our method can recognize activities performed in an interleaved fashion.

Extensive experiments with real-world datasets showed that the accuracy of our unsupervised method is comparable to the one of supervised approaches, even using a smaller number of sensors. Our method addresses the research question **Q2)** presented in Section 2.5, thus recognizing ADLs with a scalable method which avoids the acquisition of an annotated dataset

On the negative side, our technique requires a relevant knowledge engineering effort to define a comprehensive ontology of activities, home environment, and sensor events. For instance, our ontology includes 235 classes and 59 properties. One could argue that exploiting ontological reasoning is not worth the effort, since it would be easy to manually estimate correlations among activities and sensor events based on common sense. However, consider the CASAS setup used in our experiments: it involves 70 sensors and 8 activities, resulting in 560 combinations of activities and sensor events. Other real-world deployments are much more complex. Hence, manual modeling would be unfeasible in a realistic scenario.

We point out that the knowledge engineering effort can be reduced by reusing existing ontologies. In particular, the ontology used in this work is an extension of the COSAR ontology [71], which was originally intended to model context data and human activities. The extension mainly regarded the definition of a few classes of activities and artifacts that were not considered before, and a few additional properties used by our reasoning method. Developing the extension required one day of work by a researcher with good skills in OWL 2 modeling. Moreover, we were able to use the same ontology for both apartments involved in our experimentation, which had very different characteristics. However, it is questionable whether in larger scale implementations the same ontology can be adequate to cover every possible home environment and individuals' mode of activity execution. We intend to perform extensive experiments in real-world environments to answer this question. Moreover, as a future research direction, we want to exploit active learning to fine-tune the probabilistic model according to the user's environment and personal habits, and to automatically evolve the ontology according to the current context.

Chapter 5

From macro- to micro-activity recognition: Unobtrusive detection of object manipulations

5.1 Introduction

In the previous chapters we presented two techniques to recognize high-level activities. However, complex ADLs often consist of a sequence of simpler activities, also called *micro-activities*. For instance, the activity *taking medicines* can be composed of the following sequence of *micro-activities*: *extracting the medicine box from the drawer*, *moving the medicine box on the table*, *fill a glass with water* and *take the medicine*. The detection of those *micro-activities* can be adopted as an intermediate step to recognize more complex ADLs. In this chapter we focus on a novel method to recognize a specific set of *micro-activities*: objects manipulations. Indeed, ADLs recognition proved to be very effective when the interaction of the inhabitant with household items is considered. Analyzing how objects are manipulated can be particularly useful, in combination with other sensor data, to detect anomalies in performing ADLs, and hence to support early diagnosis of cognitive impairments for elderly people. We propose an unobtrusive solution which shifts all the monitoring burden at the objects' side. In particular, we investigate the effectiveness of using tiny BLE beacons equipped with accelerometer and temperature sensors attached to everyday objects. We adopt statistical methods to analyze in real-time the accelerometer data coming from the objects, with the purpose of detecting specific manipulations performed by seniors in their homes.

The advantages of having sensors on everyday artifacts for ADL recognition have been identified long ago, exploring solutions mainly based on accelerometers and RFID [114, 115], however the technology has not been sufficiently reliable and cost-effective for a wide-scale deployment. A common argument against using sensor-augmented objects as opposed to wearables for ADL recognition, in addition to technological issues, has been the difficulty in identifying the subject that is performing the activity in case of multiple inhabitants of the same space [116]. On this respect, there has been some progress on this issue both on the technological side (miniaturization of identifying beacons) and on wearable-free solutions based on data analysis [117]. Another approach to recognize specific object manipulations without neither sensors on objects nor wearables takes advantage of audio and/or video recording [97], but this solution is often perceived as too obtrusive.

Our investigation is driven by a specific application domain: the recognition of fine-grained anomalies in performing instrumented activities of daily living by elders at risk of cognitive impairment [3]. Clinicians need to identify manipulations of specific objects in a home environment including omissions, substitutions and improper manipulations. For example, these include reaching and opening a wrong medicine box, using the wrong tool to perform an action or unnecessarily repeating a given manipulation. The system described in this chapter is not intended by itself to support early diagnosis based on improper object manipulations. However, reliable object manipulation monitoring is an essential subsystem of a more complex monitoring environment. In particular, what we describe is intended to substitute the RFID-based subsystem used in [3] to monitor the use of items in preparing and consuming meals as well as taking medicines. As shown in Figure 5.1, in order to recognize anomalies in performing these high-level activities other sensor subsystems are used, including sensors revealing presence, pressure, temperature, power consumption and more.

In our experience on deployments in the real homes of the elderly for continuous monitoring, solutions based on wearables are critical: there is no guarantee that wristbands or pendants are constantly worn, not to mention smart-phone or RFID readers that have been proposed for the advantage of identifying the specific manipulated object. There are also indications of a general adversity or disaffection of users to wearables targeted to healthcare related applications [41]. Similarly, cameras and microphones are sometimes tolerated in retirement residences, but much less in private homes.

Our major contributions are experimental results on the effectiveness of unobtrusive object manipulation recognition, using current commercial low cost and low energy consumption multi-sensor devices that can be attached to everyday objects. A closely related work is [118], which uses acceleration data acquired from sensors on items to evaluate surgeons' skill in manipulating precision tools. With respect to that work, we monitor manipulations relevant to our application domain, which are more coarse-grained and of a different nature. We collected a dataset of more than two thousands labeled manipulations, and we report encouraging preliminary results on their recognition through machine learning techniques applied to accelerometer data collected from the objects. We believe that our study contributes to the design of a sensing subsystem that could be effectively integrated into the smart-home environments used in several previous works on monitoring complex activities at home [119, 120, 121], independently from the algorithmic method being used, since object manipulations may be considered as simple events.

This chapter is structured as follows. In Section 5.2 we introduce the notation to formalize object manipulations. Our framework based on machine learning to detect fine-grained object manipulations is explained in Section 5.3. In Section 5.4 we provide a detailed description of our experimental evaluation. Finally, Section 5.6 concludes the chapter.

5.2 Modeling objects manipulations

We define as *object manipulation* the interaction of an individual with an object of interest with the objective of achieving some task within the execution of a particular ADL. More formally, we define a manipulation instance as $m = \langle o, M, t_s, t_e \rangle$, where o is the object manipulated, M is the manipulation type, t_s and t_e are respectively the start and end time of the manipulation. Given ai_A an instance of an ADL A, we say that a manipulation $m \in ai_A$ if m is performed during ai_A.

Example 5.2.1. Considering as object of interest a glass, some possible types of manipulation of that object could be: using the glass to drink while eating a meal, moving the glass on the table while preparing the table, emptying the glass in the sink, inserting the glass in the dishwasher, and so on. A manipulation instance could be: $m = \langle$*glass*, *drinking*, 12:45:32, 12:45:45$\rangle$ where $m \in ai_{eating}$ (a manipulation which consists in using the glass to drink during the consumption of a meal).

We point out that not every type of object manipulation is interesting for monitoring ADL execution, hence we divide the manipulation types in two categories: *relevant* ad *irrelevant*. We consider a manipulation *relevant* if the task that is achieved by performing the manipulation is crucial to monitor a particular ADL; *irrelevant* otherwise. Of course, classification of manipulation types in *relevant* and *irrelevant* has to be decided accurately by domain experts. Manipulations considered as *relevant* are further classified in specific sub-classes.

Example 5.2.2. Suppose that we're interested in monitoring the activity of taking medicines. In this scenario, we consider a manipulation relevant if a medicine package is extracted from its repository, or if it is opened; while it is considered irrelevant if a medicine package is just slightly displaced inside the repository while searching.

5.3 The technique

In this section, we illustrate our technique to analyze the data coming from accelerometer positioned on objects, in order to recognize specific manipulations.

5.3.1 Recognition Framework

The system is considered as part of a smart home environment instrumented with several environmental sensors. The general architecture is shown in Figure 5.1. Each object of interest has attached a wireless device which incorporates a 3-axis accelerometer sensor. Each device communicates periodically the raw sensor data to the *Smart-object data processing* module, along with the device's unique identifier. This module is in charge of: a) segmenting the accelerometer data in order to identify the manipulation occurrences, b) extracting several features and c) applying machine learning techniques in order to recognize the specific manipulation performed. Since each type of object has an associated set of specific manipulations (e.g., a bottle of water is used to pour/drink water, differently from a medicine box that is used to extract pills), we built a specialized classifier for each object

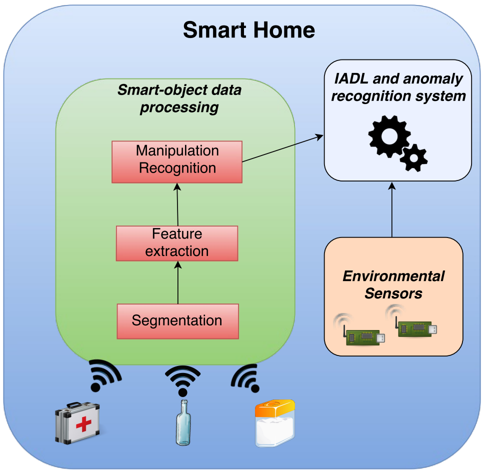


Figure 5.1: General architecture

type. Of course, this does not mean using a different classifier for each object: for instance, a bottle of water and a milk box can be manipulated similarly and a single classifier is in charge of recognizing the manipulations of both objects. Detected manipulations, along with measurements acquired from smart-home environmental sensors, are transmitted to a system which is in charge of recognizing ADLs performed by the monitored subject and the possible abnormal behaviors.

5.3.2 Segmentation and feature extraction

We pre-process data transmitted from the objects in order to identify the manipulation occurrences. To do this, we analyze 3-axis accelerometer data in order to detect whether an object is in motion. This is done by using a straightforward threshold based method on accelerometer data which detects when the object starts and stop moving. Each manipulation occurrence $occ_i = \langle o, t_s, t_e, \vec{x}, \vec{y}, \vec{z} \rangle$ is represented by: the object o manipulated, the start time t_s (i.e. the time instant where the object started moving), the end time t_e (i.e. the time instant where the object stopped moving) and the accelerometer data on the three axis. The output of segmentation module is a set of n manipulation occurrences $O = \{occ_1, occ_2, \ldots, occ_n\}$.

From each manipulation occurrence, we build a feature vector which comprises more than 40 different features regarding statistics on accelerometer data and the duration of the manipulation. In particular, for each axis ($\vec{x}, \vec{y}$ and $\vec{z}$) we consider the following statistics:

- Maximum
- Minimum
- The difference between maximum and minimum
- Mean
- Variance
- Standard Deviation
- Zero-crossing rate
- Root mean square
- First quartile
- Third quartile
- Energy
- Kurtosis
- The average difference between consecutive acceleration values

Moreover, for each pair of axis we also compute the Pearson Correlation and the covariance. We selected these features considering the literature on activity recognition from acceleration data [38, 122, 123].

5.3.3 Manipulation recognition

The next step is to infer, for each feature vector, the specific manipulation performed with the related object. As previously described, for each type of object we're interested in distinguishing between *irrelevant* manipulations and a set of specific *relevant* manipulations. Since we're not interested in detecting the fine-grained types of *irrelevant* manipulations, they're grouped together into a single class called *Irrelevant*. We adopt a supervised approach, using state-of-the-art classifiers like *Random Forest* [101] and *AdaBoost* [124] depending on the specific object.

We adopted two different classification approaches:

- Direct classification
- Multi-layer classification

In the following we describe these techniques.

Direct classification

Our first straightforward approach consists in directly distinguishing the fine-grained manipulations using a multi-class classifier. This method is shown in Figure 5.2. Hence, depending on the type of object from which the manipulation comes, a

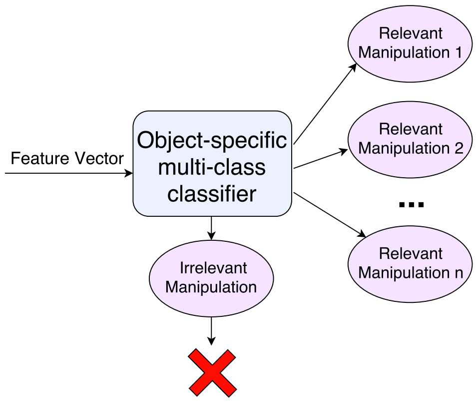


Figure 5.2: Direct classification schema for a specific object

specific classifier is used. Every single classifier is trained with a set of relevant manipulations and irrelevant manipulations of the specific object type.

Multi-layer classification

With the objective of improving the above mentioned method, we also propose a different approach, which is represented in Figure 5.3. Instead of directly detecting the manipulation type from the feature vector, we use two layers. In the first layer a binary classifier is in charge of distinguishing, for a specific object, *relevant* manipulations from the *irrelevant* ones. This classifier is trained with relevant manipulations (all grouped together in the same class) and irrelevant manipulations of the specific object. Only the manipulations which are classified as *relevant* are forwarded to the second layer, while the others are discarded. In the second layer, a multi-class classifier is in charge of recognizing the specific relevant manipulation performed on the object. Hence, this classifier is trained only with the corresponding relevant manipulations.

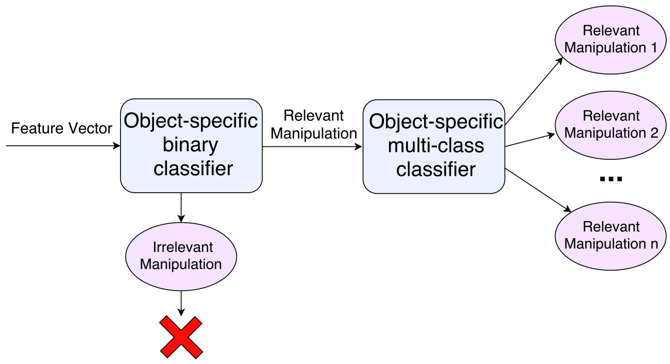


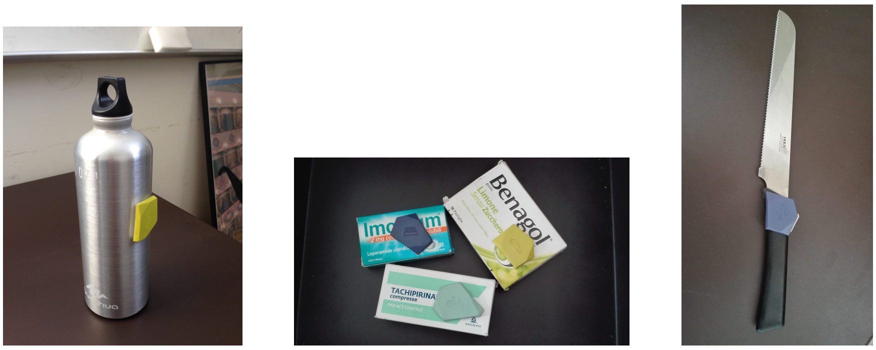
Figure 5.3: Multi-layer classification schema for a specific object

5.4 Experimental evaluation

In this section we describe our experimental setup, how we acquired a dataset of manipulations, and finally we present our preliminary results.

5.4.1 Experimental setup

Driven by requirements from clinicians, we currently focus on fine-grain monitoring of three complex activities: *preparing a meal*, *consuming a meal* and *taking medicines*. While the final deployment of stable versions of the system is in real homes, our experimental activity is conducted in a smart-room lab. Activity recognition is performed by processing data coming from a wide variety of environmental sensors, including pressure pads, temperature sensors, power meters, magnetic switches, presence sensors and more. The experiment reported in this work is intended to verify the viability of substituting our RFID based solution for recognizing manipulations of specific items. For this purpose we selected specific objects: a) medicine boxes as they have a key role in monitoring adherence to prescription and their improper manipulation can also be a useful indicator, b) a liquid bottle as it is an example of an item used in meal consumption, may have to be refrigerated, and may also play a role in monitoring water consumption, and c) a kitchen tool, a knife in particular, as a tool being used both in meal preparation, and in meal consumption. These objects are shown in Figure 5.4 with their sensing device attached.



(a) Liquid bottle (b) Medicine boxes (c) Knife

Figure 5.4: The monitored objects

5.4.2 The sensing devices

In order to monitor objects manipulation, we take advantage of current off-the-shelf devices: Estimote's Stickers. A sticker is a packaged PCB with a battery-powered ARM CPU equipped with 3-axis accelerometer, temperature sensor, and a Bluetooth Smart radio able to periodically broadcast its sensed data in a short range (a few meters). Their tiny packaging makes it easy to attach them on objects as shown in Figure 5.4. Each sticker can be easily distinguished by a unique identifier which is particularly useful to improve manipulation detection by exactly knowing which kind of object is manipulated. Estimote Stickers adopt a proprietary communication protocol called Nearables; Table 5.1 reports the data frame of this protocol. In our setup, each sticker broadcasts a packet every 100 milliseconds while it is moving; every 200 milliseconds otherwise. A BLE scanner is in charge of collecting the data coming from each sticker.

5.4.3 Sensor data analysis

We perform data acquisition by scanning the BLE signal through a mobile device. In order to perform segmentation, we exploit the value of the *Motion* field transmitted in every packet by the stickers. This field is set to *true* when the sticker is in

Field	Description
Identifier	Unique identifier
Motion	Whether the sticker is moving (boolean)
xAcceleration	X-Axis acceleration
yAcceleration	Y-Axis acceleration
zAcceleration	Z-Axis acceleration
Temperature	Stickers temperature value (in Celsius)
Orientation	Physical orientation of the sticker
RSSI	Signal strength
Power	Signal strength at 0 meters
Battery Level	Sticker's battery level

Table 5.1: Nearables data frame

motion. Our experiments revealed that this value provides sufficient accuracy for determining begin and end of our manipulations. Hence, for a specific sticker we consider all the consecutive data packets with the *Motion* field set to *true* as part of the same manipulation occurrence.

A labeled dataset is used to construct the predictive model. We performed experiments with different type of models for each of our considered objects, and we selected Random Forest for the liquid bottle and the medicine boxes, and AdaBoost for the knife manipulations.

Segmentation, feature extraction and classification are performed in real time on the mobile platform. The output serialized in JSON format is sent to a REST server for integration with events detected by processing data coming from other sensors as illustrated in Figure 5.1.

5.4.4 Dataset collection

Since our recognition technique is based on supervised machine learning, a critical task is the acquisition of a sufficiently large and significant dataset of object manipulations. The dataset must also be annotated with the ground truth related to each manipulation. In order to facilitate this task, we developed a mobile application. The application starts with a simple screen consisting in only one button. When that button is clicked, the Bluetooth scanner starts acquiring Nearables data packets, which are internally stored. After a few manipulations we conclude the experiment, and the app performs segmentation, it creates a set of three axis acceleration data for each manipulation and then allows the user to label each one with the ground truth (Figure 5.5).

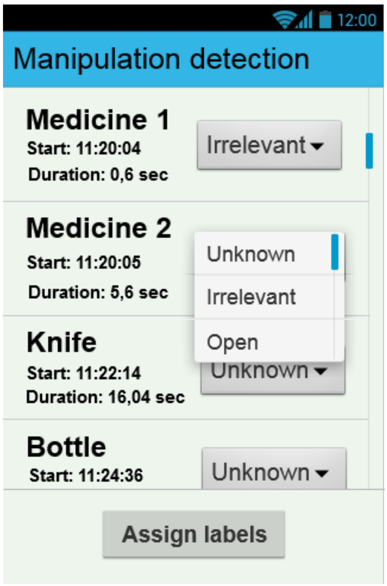


Figure 5.5: Application layout

As already mentioned, in this experiment, we focus on manipulations of a liquid bottle, a medicine box, and a knife. For the purpose of this first assessment of our system we collected manipulations performed by six different adults without physical impairments. They executed those manipulations spontaneously within realistic scenarios of activities of daily living executed in a smart room lab (e.g. cooking, taking medicines, . . .). The total number of manipulations is 2058, with 887 manipulations involving the liquid bottle, 656 the medicine boxes and 515 the knife. Out of the total, 1365 manipulations are considered relevant, while the rest are considered irrelevant. This distinction is clearly application dependent, and in our case it has been driven by the scenarios of our e-health domain and by the interest in specific manipulations by the clinicians.

It is important to consider that the specific way in which we perform segmentation can lead to group more than one manipulation into a single one; for example, if a subject extracts the water bottle from the fridge and pours the water in a glass as a single action without interruption, the motion value of Nearable remains true and the whole action will be segmented as a single manipulation. On the contrary, if the bottle is moved from the fridge to the table, and then used to fill a glass, the system will identify two manipulations. We considered alternative segmentation methods, but actually observed that the presence of these 'composed' manipulations is sometimes a benefit for our specific application, considering the final recognition accuracy.

Liquid bottle's manipulations

The total number of manipulations of the bottle that we acquired is 887. We consider 500 of them as relevant because their detection is useful to monitor the activity of meal consumption or even just *drinking* (e.g., "extract from the fridge" or "pour water"). Table 5.2 shows how we classify manipulations of the bottle.

Class	**Include**	**Description**
Irrelevant	Minor displacement	Bottle is displaced in the same place
	Irrelevant	Bottle is moved, but not by a person (e.g., movements of the fridge)
	Displacing in the fridge	Bottle is displaced inside the fridge
	Opening/closing fridge door	When the bottle is in the fridge door and it is opened, bottle moves
Relevant displacement	Displaced	Bottle is displaced from a place to another which is not a fridge
	Inserted	Bottle is displaced from a place to the fridge
	Extracted	Bottle is displaced from the fridge to a place
Drinking/Pouring	Drink	Bottle is taken from a place and is brought to lips and tilted
	Pour	Bottle is taken from a place and liquid is poured in a glass

Table 5.2: Liquid bottle's manipulations

Medicine box's manipulations

The total number of these manipulations is 656. We consider 474 of them as relevant, because their detection is useful to monitor the activities "taking medicine" (e.g. "extract from the repository" or "open medicine box"). Table 5.3 shows how we classify manipulations of medicine boxes. Note that distinguishing manipulations like "displacing the medicine M box" and "accessing the content of medicine M box" is very important in our domain, since the first if not followed by the second may be an indication that the patient prepared the medicine but in the end forgot to take it.

Class	Include	Description
Irrelevant	Handle	Medicine box is taken and handled
	Irrelevant	Medicine box is moved, but not by a person (e.g., hit the repository)
	Displacing in the repository	Medicine box's manipulations when someone searches for the correct one
	Opening/closing repository drawer	When the medicine box is in the repository and it is opened, medicine box moves
Relevant displacement	Displaced	Medicine box is displaced from a place to another which is not the medicine repository
	Inserted	Medicine box is displaced from a place to the correct repository
	Extracted	Medicine box is displaced from the repository to a place
Accessing content	Opened	Medicine box is taken from a place and a blister pack is extracted in the same place or in another

Table 5.3: Medicine box's manipulations

Knife's manipulations

The total number of manipulations involving the knife is 515. We consider 391 of them as relevant because their detection is useful to monitor the activities "preparing meal" (e.g. "extract from the drawer" or "cut something"). Table 5.4 shows how we classify manipulations involving the knife.

5.4.5 Results

Table 5.5 summarizes our results on the recognition of object manipulations. We use a 10-folds cross-validation method. The table shows both the results using the direct classification approach and the ones using the layered approach. Despite several extensions will be required, we considered these results encouraging since they show that direct classification with a simple segmentation strategy and state-of-the-art machine learning already provides quite adequate accuracy for our

Class	Include	Description
Irrelevant	Irrelevant	Knife is moved but not by a person (e.g., the repository is shaked)
	Displacing in the repository	Knife's manipulations when someone searches for a tool or silverware
	Opening/closing repository drawer	When the knife is in the repository and it is opened, the knife is moved
Relevant displacement	Displaced	Knife is displaced from a place to another which is not the knife repository
	Inserted	Knife is displaced from a place to the correct repository
	Extracted	Knife is displaced from the repository to a place
Cutting	Cut	Knife is taken from a place and something is cut in the same place or in another

Table 5.4: Bread knife's manipulations

application requirements. We expected more from the layered approach that shows improvements only on specific object manipulations.

5.5 Limitations of current BLE technology

The use of BLE accelerometers attached to objects addresses important drawbacks of different technological solutions proposed in the literature. However, it currently has several limitations. First of all, energy consumption, since we observed that the high transmission rate we used reduced the battery life to levels not acceptable for a real home deployment. Energy consumption was also affected by the need to increase the standard transmission power in order to cover at least the whole room. A second problem we found is interference when these devices are close to metal objects. Other problems arise when the monitored objects are dipped in water or exposed to high temperatures, since the devices would be damaged. However, we are confident that technological evolution will soon solve these limitations, while the ones affecting other approaches are not only technological, but involve user acceptance and privacy issues that may be more difficult to overcome.

	Accuracy (%)		
	Multi-layer classification	Direct classification	Total occurrences
Bottle			
Total	90,41	**91,54**	887
Irrelevant	91,73	**94,83**	387
Rel. displacement	85,28	**85,71**	231
Drink/Pour	**92**	91,82	269
Medicine box			
Total	92,07	**92,98**	656
Irrelevant	88,46	**91,75**	182
Rel. displacement	84,67	**85,40**	137
Accessing content	**97,03**	96,73	337
Knife			
Total	**96,88**	96,11	515
Irrelevant	**97,58**	97,56	124
Rel. displacement	**95,51**	93,58	156
Cut	**97,44**	97,02	235
Total			
Total	92,56	**93,14**	2058
Irrelevant	91,91	**94,51**	693
Relevant	**93,51**	93,06	1356

Table 5.5: Results

5.6 Summary

In this chapter, we proposed a method to detect fine-grained manipulations performed on everyday objects exploiting small BLE accelerometers attached to the objects of interest. A simple segmentation strategy and standard machine learning techniques are applied to the continuous stream of accelerometer data to classify in real-time the most likely manipulation that the subject performed on the monitored object. Extensive experiments with a dataset consisting of more than two thousands manipulations show that our approach can obtain encouraging results. Our method addresses the research question **Q3)** presented in Section 2.5, thus unobtrusively recognizing fine-grained manipulations performed by the inhabitant on household objects. We intend to extend our work in several directions. First of all, we want to combine accelerometer data with temperature information of the object in order to identify displacement of items to and from refrigerated repositories, as well as recognizing when an item which needs to be refrigerated has been forgotten somewhere else the kitchen. Acceleration data can also be usefully combined with fine-grained indoor positioning data, as well as other sensor data to refine manipulation detection. We also intend to investigate and evaluate different recognition techniques. Finally, we aim to consider a richer set of manipulations (and objects involved) acquiring the dataset from our target users, the elderly, in their homes.

Chapter 6

A health-care use case: Fine-grained and long-term anomalies recognition

6.1 Introduction

In this chapter we present a specific use case scenario for activity recognition in the health-care domain. In particular, we investigate one of the most frequent threats to independent living: cognitive decline. Indeed, its first early symptoms often lead to a Mild Cognitive Impairment (MCI) diagnosis. According to the International Working Group on MCI, there is evidence of subtle differences in performing activities of daily living (ADLs) among MCI patients compared to both healthy older adults and individuals with dementia [77]. Other studies [125, 126] observed how a closer examination of functional skills in individuals with MCI may enhance our understanding of the natural history and cognitive correlates of functional deterioration associated with dementia. They pointed out the limits of informant-based reports on subject abilities and proposed to extend well-known performance evaluation tests (e.g., NAT [127]) with *subtle errors* recognition. Hence, from a medical point of view there is a clear interest in methods to monitor the elderly with the goal of identifying specific abnormal behaviors as indicators of cognitive decline. Indeed, several research projects, and numerous research papers have tried to detect behavioral markers of MCI onset through ubiquitous computing technologies, obtaining a correlation between the predicted and actual cognitive status of the patient. A general approach is to build a model of the "regular" behavior in order to identify those activity patterns which diverge from the expected ones [120]. The main drawback of this type of approaches is that behavioral changes are detected without giving specific explanations of what happened. Other research groups tried to refine the identification by recognizing the general anomaly's category (e.g. omission, substitution, replacement, ...) using statistical methods [121]. However, the results show a high rate of false positives. Moreover, some of these approaches require the execution of ability tests about the performance of ADLs in an instrumented smart home of a medical institution; hence, they incur in high costs and cannot be applied on a continuous basis. Some of them deploy cameras and sensor networks in controlled environments and use video and audio for activity recognition: these systems are often perceived as too invasive for the elderly's privacy. Other works rely on continuous monitoring of low-level behavioral markers (steps taken, walking speed, ...) and trigger alarms whenever they detect situations sufficiently distant from the expected (modeled) behavior.

We have joined this research effort by designing and implementing a pervasive system for fine-grained abnormal behavior recognition [31, 3, 17]. We propose a tool for clinicians for analyzing the decline of functional abilities, supporting the diagnosis of MCI or even distinguishing between different MCI subtypes. Our system has a sensor network component intended to be installed in the home of the senior and continuously acquiring data. Video and audio acquisition are excluded as too intrusive, while sensors are used to detect the presence in particular locations, opening and closing of drawers, fridge and cabinet doors, use of appliances, as well as the use of specific tools and food items.

Our system identifies the *anomalies* that can be observed in carrying out the activity (e.g., inappropriate timing in assuming food or medicine intake, improper use of equipment, unnecessary repetitions of actions). This is a challenging task for at least two reasons: a) only certain anomalies or patterns of anomalies are relevant indicators for clinicians and they need to be properly modeled based on cognitive neuroscience expertise; b) most approaches to activity recognition lack the ability to identify the fine-grained anomalies that are of interest to clinicians.

Preliminary and encouraging results of fine-grained anomalies recognition have been presented in a related Ph.D. thesis [128]. In this work we extend that system in order to consider new fine-grained abnormal behaviors based on objects manipulations. To do so, we exploit our results presented in Chapter 5 on using Bluetooth Low Energy (BLE) accelerometers attached to everyday objects in order to recognize performed manipulations. We present preliminary results on a new dataset consisting of hundreds of complex/interleaved ADLs and anomalies.

We also propose a novel long-term analysis method to detect significant changes in the trend of performing activities and to avoid raising alerts for isolated abnormal activities [32]. In particular, we introduce a novel system to automatically recognize long-term abnormal behaviors (e.g., changes in habits regarding the timing of meal consumption).

The definitions of fine-grained and long-term abnormal behaviors are provided in Section 6.2. In Section 6.3 we introduce our framework to detect fine-grained abnormal behaviors based on objects manipulations. Finally, our system to detect long-term abnormal behaviors is presented in Section 6.4.

6.2 Fine-grained and long-term abnormal behaviors

6.2.1 Fine-grained abnormal behaviors

By *fine-grained abnormal behaviors* (also called *anomalies* for short) we define those behaviors, observed during the execution of everyday tasks by a subject, which diverge from the expected ones, according to a given model provided by clinicians. In particular, we consider models of abnormal behaviors that may indicate the onset of MCI, and more generally of a cognitive decline. In order to formally specify those models, we considered previous studies on these indicators [125, 126] as well as medical practice results [129], and we collaborated with cognitive neuroscience experts from the Institute Fatebenefratelli[1], Lombardy –a leading center in the field of mental health research and research on neurodegenerative disorders– within the SECURE[2] research project funded by Lombardy region

[1]IRCCS (Research and Care Institute) St John of God Clinical Research Centre, Brescia – `http://www.irccs-fatebenefratelli.it`

[2]SECURE: Intelligent System for Early Diagnosis and Follow-up at Home, `http://secure.ewlab.di.unimi.it/`

and MIUR Italian ministry.

Our term *fine-grained* refers to the ability of distinguishing each type of anomaly through the identification of single actions and on the analysis of their sequence, frequency and relation to specific activities. We also believe that this is related to the notion of *subtle errors* investigated in [126] that proved to be important indicators of early phases of cognitive decline. These errors include specific gestures or manipulation of objects. For example, "Picks up and puts down sugar bowl without using it".

Neuropsychology researchers characterized several functional difficulties in achieving everyday tasks that may be predictive of serious cognitive disorders like MCI or Dementia [125]. Each category corresponds to several different types of abnormal behaviors. The categories that we considered in this work are presented in Table 6.1.

Table 6.1: List of considered abnormal behaviors categories

Type of anomaly	**Description**	**Example**
Omission	An important step within an ADL is not performed	The medicine box has been retrieved but no medicine is taken
Substitution	A different object than appropriate is used or a different component action than expected is performed	Pouring sugar instead of salt to prepare pasta
Replacement	The subject replaces a correct action with a wrong one	Putting the medicine box in the fridge
Wrong activity	The subject performs an activity that should not be done	The subject takes a not prescribed medicine
Inefficient execution	The subject performs actions which slow down/compromise the execution of the ADL	The subject takes double of the usual time in watering plants
Repetition	The subject repeats an ADL that he/she already performed forgetting it already took place	A medicine which is prescribed once is taken twice
Searching	The subject actively searches through home's repository for an item	The subject forgets where he/she put the salt and he/she searches it in all the repositories

Activity-dependent anomalies

Abnormal behaviors may depend on how ADLs are executed (or non-executed), while others are more generic and not tight to a specific ADL. In our model we consider both types of anomalies, distinguishing the two categories:

- *Activity-dependent*: if the anomaly is contextual to the occurrence (or non-occurrence) of one or more activity instances (e.g., the elderly executes the activity of taking medicines but takes the wrong medicine)

- *Activity-independent*: if the anomaly is not contextual to the occurrence (or non-occurrence) of one or more activity instances (e.g., the elderly just keeps on wandering around, searching for something for an unusually long time)

In the case of *activity-dependent* anomalies, they could be related to one or multiple activity instances. For instance, the omission of the activity *Preparing the table* can be considered as anomalous only if the ADL is not performed before the activities *Eating lunch* or *Eating dinner*, while it is not anomalous to omit it before the *Preparing breakfast* ADL. Another example is the repetition of medicines intake, where the same medicine is taken twice in two different *taking medicines* instances within the same prescription time.

Activity-independent anomalies, on the other hand, only rely on the sensed information. For instance, a substitution like *the butter is inserted in a non-refrigerated repository* should be fired independently with respect of recognized ADLs.

Subject-dependent anomalies

Orthogonally with respect to activity dependency, we also consider subject-dependent abnormal behaviors. Indeed, a behavior which is abnormal for an individual could be normal for another one. First of all, this may depend on medical prescriptions (e.g. medicines to be taken, diet, ...). Considering this type of information it is possible to provide detailed anomalies like "*the patient forgot to take his/her morning medicine*".

In addition to medical prescriptions, another important aspect of personalized abnormal behaviors is personal habits. For instance, the usual time and duration of execution of an ADL may vary for each person. Whenever a subject changes significantly his/her habits (e.g. taking longer to perform ADLs), it may be a symptom of a cognitive decline. Hence, we define some rules which capture the deviation from a past "normal behavior", mining statistics on the normal execution of ADLs.

In general, subject-dependent rules are dynamically generated by considering medical prescriptions and personal habits.

Example 6.2.1. Consider the case where the subject is searching for salt to prepare pasta. It may be a normal habit to open two or three repositories in order to effectively find and retrieve the salt shaker. Hence, until the subject keeps on behaving as usual, no anomaly is detected. However, if the subject wanders around the home opening several times different repositories (much more than the usual two or three times), it may be considered as an abnormal behavior. Hence, the anomaly related to *searching for an item* is dynamically defined based on the past normal behavior of the subject.

6.2.2 Long-term abnormal behaviors

Human behaviors are characterized by wide variability; factors such as contextual conditions, individual habits and personality traits may determine the execution of various anomalies that are not necessarily due to cognitive impairment. Consider, for instance, the anomaly of leaving repositories open. This may be normally done by cognitively healthy people for negligence or hastiness. Hence, when considered in isolation, fine-grained abnormal behaviors are only weak indicators of possible cognitive issues. On the contrary, the frequency of anomalies detected over long periods of time and their temporal trend are much stronger indicators.

We define as *long-term abnormal behaviors* those groups of activities and anomalies, observed over relatively long periods of time (from one week to several months), showing significant changes from the normal trend observed in the past, and which may indicate the onset of cognitive impairment or the progression of MCI. Long-term abnormal behaviors are better indicators when *personalized*, i.e., when specified with respect to trends observed as 'normal' for a specific patient or patient-profile. In particular, we focus on abnormal behaviors that emerge from a personalized long-term temporal analysis of performed activities (e.g., considering the timing of meal consumption, duration of meal preparation), since time-related difficulties in task executions are known to be associated with MCI onset [130].

6.3 Recognition of fine-grained abnormal behaviors based on objects manipulations

6.3.1 General architecture

The architecture of our recognition framework is shown in Figure 6.1. Our frame-

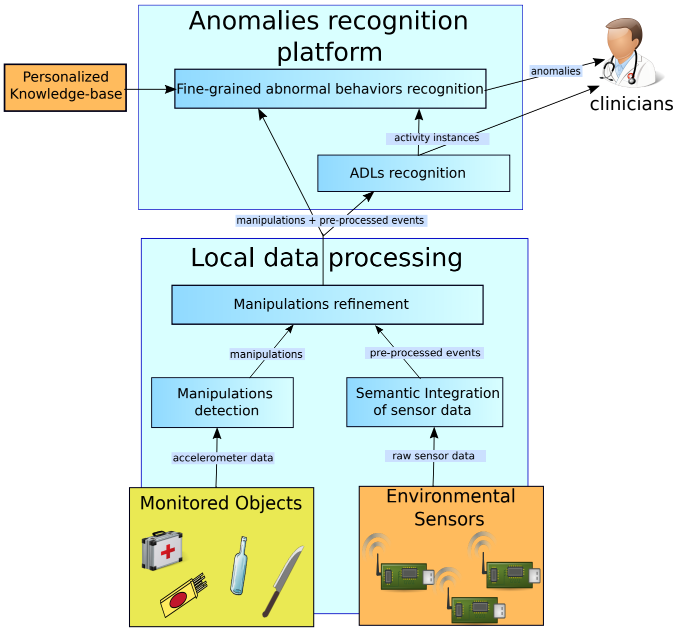


Figure 6.1: Overall anomaly recognition framework

work consists of two main components. The LOCAL DATA PROCESSING part is in charge of continuously collecting and pre-processing raw data from sensing devices. It runs within the smart-home environment. The ANOMALIES RECOGNITION PLATFORM component runs recognition algorithms on data provided by the LOCAL DATA PROCESSING component. Currently, the recognition algorithms run periodically (e.g., on all the data collected in each day). This component could be deployed both in the smart-home environment or as a cloud service. We consider a smart-home environment instrumented with two kinds of sensing devices: a) environmental sensors to monitor the inhabitant's interaction with the home environment, b) wireless accelerometers attached to a set of everyday objects in order to recognize the performed manipulations. Raw data from environmental sensors are preprocessed by SEMANTIC INTEGRATION OF SENSOR DATA module, which applies simple inference rules to derive high-level events. The MANIPULATIONS

DETECTION module applies the manipulation recognition technique presented in Chapter 5. The MANIPULATIONS REFINEMENT module combines the detected manipulations with information derived from environmental sensors to derive more precisely characterized manipulations. This process is done by a set of rules taking into account temporal and semantic relationships.

Example 6.3.1. Suppose that the system detected the manipulation of a medicine box and classified it as *significant displacement*. This can happen if the medicine box has been moved from a place to another[3]. If the system also detects that the medicine repository has been opened just before the start of the manipulation, the system can infer that the medicine box has been retrieved from the medicine repository. Hence, the manipulation class provided by the MANIPULATION DETECTION for that specific box in that timespan will be converted from *significant displacement* to *retrieved from repository*.

Refined manipulations along with pre-processed events are then temporally totally ordered. Aggregated and refined sensing data is used by the ADLS RECOGNITION module to detect activity instances with their timespans. Detected ADLs along with sensed data are transmitted to the FINE-GRAINED ABNORMAL BEHAVIORS RECOGNITION module, which applies knowledge-based reasoning to infer the occurred fine-grained abnormal-behaviors.

6.3.2 Fine-grained anomalies recognition

The abnormal behaviors are usually described in natural language by domain experts (i.e., clinicians). We use a first-order logic knowledge-base to model those descriptions in terms of temporal relations between detected ADLs, high-level events, manipulations and personalized knowledge of the monitored individual. Then anomaly recognition is performed using a logic programming engine. In our model, an anomaly is represented with the predicate

$$anomaly(an, aid, obj, t_s, t_e)$$

where an is the anomaly's type, aid is the identifier of the activity instance related to the anomaly (if any), obj is the object related to the anomaly (if any), and t_s

[3] A displacement is characterized as significant when the movement is not just a minor involuntary change of position, like for example a medicine box moved within a drawer while grasping a different box

and t_e are respectively the starting and ending time of the anomaly occurrence. A manipulation occurrence is represented by the logic fact

$$manipulation(o, m, t_s, t_e)$$

where o is the object being manipulated, m is the manipulation label provided by the machine learning algorithm, and t_s and t_e are the starting and ending time of the manipulation, respectively. A sensor event as, for example, the opening of a cabinet is represented with the logic fact

$$action(\textit{Cooking-Id, open, Kitchen-Drawer, 2016-11-12 12:01:34})$$

where the first argument is an activity instance id (used only when the system has classified the event as part of an activity), the second argument is the event type, the third is the object/area involved, and the last is the timestamp. An activity, as for example *Cooking*, once recognized by the system is represented by the logic fact:

$$activity(\textit{Cooking-Id}, \textit{Cooking}, \text{2016-11-12 11:58:00, 2016-11 12-12:05:12})$$

where the first argument is the activity instance identifier, the second the activity type, and the last two are starting and ending timestamps, respectively. Subject-dependent knowledge, like prescribed medication, is also added to the knowledge base in terms of logic facts. For example, if $medicineA$ has to be taken every day between 8 and 9 am, the fact $prescribedMedicineTime(medicineA, 8am, 9am)$ is added to the knowledge base.

Table 6.2 illustrates some examples of first-order logic rules used to infer abnormal behaviors. The role of object manipulations in the process of recognizing fine-grained anomalies is highlighted by the first two rules. Both rules detect an *omission*: the subject did not take a medicine which was prescribed in a specific time interval (e.g. in the morning). In the first case, the subject completely forgets to take the medicine, while in the second case the subject actually retrieves the medicine from the repository but then forgets to take it. Even if the practical implication of the two anomalies is the same, they represent two different patterns which may be important to distinguish for devising appropriate intervention mechanisms and possibly also for the clinical evaluation.

Table 6.2: Examples of rules modeling abnormal behaviors

No.	Rule	Anomaly type
1	$anomaly(medicine\_not\_even\_retrieved, Aid, Medicine, T_s, T_e) \leftarrow isMedicine(Medicine) \wedge prescribedMedicineTime(Medicine, T_s, T_e) \wedge activity(Aid, takingmedicine, T_{sa}, T_{ea}) \wedge (T_{sa} > T_s) \wedge (T_{sa} < T_e) \wedge (T_{ea} > T_s) \wedge (T_{ea} < T_e) \wedge not(isManipulated(Medicine, T_s, T_e))$	Omission: the subject completely forgot to take a prescribed medicine
2	$anomaly(medicine\_retrieved\_but\_not\_taken, -, Medicine, T_s, T_e) \leftarrow isMedicine(Medicine) \wedge prescribed(Medicine, T_s, T_e) \wedge isRetrieved(Medicine, T_s, T_e) \wedge isReturned(Medicine, T_s, T_e) \wedge not(isOpened(Medicine, T_s, T_e))$	Omission: the subject retrieved a prescribed medicine from the repository but then forgot to take it
3	$anomaly(refrigeratedfood\_in\_wrong\_repository, Aid, Item, T_s, T_e) \leftarrow isRefrigeratedFood(Item) \wedge manipulation(Aid, return, Item, Repository, T_s, T_e) \wedge not(isRefrigeratedRepository(Repository))$	Replacement: the subject places an object which needs to be refrigerated in a non-refrigerated repository
4	$anomaly(searching, -, -, T_{o_1}, T_{c_k}) \leftarrow action(-, open, Repository_1, T_{o_1}) \wedge action(-, close, Repository_1, T_{c_1}) \wedge (T_{o_1} < T_{c_1}) \wedge not(returnedOrRetrievedObjectsBetween(T_{o_1}, T_{c_1})) \wedge \ldots \wedge action(-, open, Repository1, T_{o_1}) \wedge action(-, close, Repository_k, T_{c_k}) \wedge (T_{o_k} < T_{c_k}) \wedge not(returnedOrRetrievedObjectsBetween(T_{o_k}, T_{c_k})) \wedge ((T_{o_k} - T_{o_1}) < (\Delta t \times k))$	Searching: the subject opens and closes several repositories more than k times to find some item without returning and retrieving any object (k is subject-dependent) within a short time interval.

The third example illustrates a rule for the recognition of an activity-independent anomaly. When the manipulation *return* of a refrigerated object is close to the interaction with a non-refrigerated repository, the anomaly is fired. Note that this particular behavior is anomalous regardless of the performed ADL.

The last rule is an example of subject-dependent and activity-independent abnormal behavior. The anomaly is fired when the subject consecutively opens and closes k repositories without retrieving or returning any item, which indicates confusion about where an item is placed. The number of repositories k which are consecutively accessed to fire the anomaly is subject-dependent, and it is mined by analyzing the past normal behavior of the subject. If the subject consecutively accesses more repositories than the usual, then the behavior is considered as abnormal. Hence, the rule is automatically generated based on the value of k. The value Δt indicates the maximum amount of time between the opening of two repositories.

6.3.3 Experimental evaluation

Experimental setup

We implemented the system's prototype within our smart lab, which is instrumented with several environmental sensors like magnetic, power, presence and plug sensors. Those sensing devices are used to capture the interaction of the inhabitant with the home environment (repositories, chairs, electrical stove, . . .) and continuously communicate their readings to a smart-home gateway using Z-Wave protocol. We also attached tiny BLE accelerometers to several objects which are interesting to monitor different ADLs. In particular, we considered medicine boxes, a liquid bottle, a knife, food/beverage packages and a watering can. Those devices continuously transmit their accelerometer data to an Android mobile application which runs the MANIPULATIONS DETECTION module. Manipulations are classified in real-time and then transmitted to the gateway.

We configured a Raspberry Pi to act as the smart-home sensor gateway to collect environmental sensors observations and object manipulations. A NodeJS REST server is in charge of receiving sensor data and storing it in a SQLite database. Periodically (e.g. at the end of each day) the gateway executes the SEMANTIC INTEGRATION OF SENSOR DATA module on environmental sensors, and transmits the derived high-level events, along with objects manipulations, to the MANIPULATIONS REFINEMENT module. Both modules are written in Java language. The MANIPULATIONS REFINEMENT module produces several log files containing the refined and aggregated sensor data. The sensor logs are used by the recognition algorithms of the ANOMALIES RECOGNITION PLATFORM, which are executed off-line.

Dataset collection

In order to validate our system, we accurately designed the acquisition of a dataset of several ADLs and anomalies. Our target activities are related to the kitchen environment. In the specific we considered the following ADLs: *taking the prescribed medicines*, *preparing breakfast*, *preparing meal* (i.e. lunch or dinner), *laying the table*, *eating*, *cleaning up* (i.e. clear the table and washing dishes) and *watering plants*. We designed several realistic scenarios, where each scenario represents a whole day of ADLs and abnormal behaviors performed by a different subject in its kitchen. Activities execution is designed to be as realistic as possible, with complex and interleaved patterns. In order to obtain a dataset which is the more general and robust possible, we introduced in all the scenarios several levels of variability in performing the ADLs/anomalies:

- **Variability in how a task is performed**: the same ADL can be performed in several different ways. For instance, a medicine can be taken with or without drinking water. Another example is the preparation of the meal, which can significantly vary depending on the recipe.
- **Variability in the order of actions**: Even two different ADLs execution which consist on the same task can significantly vary, since the order of actions can be different. Suppose, for instance, the pasta preparation. The inhabitant can significantly vary the order at which he/she accesses to repositories to retrieve food items and cooking instruments.
- **Variability in how ADLs are interleaved**: ADLs are often performed in an interleaved fashion. Hence, we introduced in the scenarios different ways of interleaving the activities. For instance, while sitting at the table during lunch, the inhabitant stops eating for a while to take its medications.

- **Variability in how an anomaly occurs**: Abnormal behaviors can occur with different patterns just like ADLs. Suppose for example the anomaly "*forgetting to take a prescribed medicine*". This can be done by totally forgetting to take it (no interaction with the medicine box) or by retrieving it at its repository but then forgetting to take it. Thanks to the manipulation recognition, we can monitor and distinguish these two cases.

Moreover, we included additional realistic scenarios, asking the actors to simulate ADLs without following pre-defined scripts. Of course, those scenarios do not contain abnormal behaviors.

In total we acquired 752 instances of ADLs and 150 different patterns of abnormal behaviors. Those ADLs and anomalies have been collected in 40 scripted and 20 unscripted scenarios executed by 19 different adult volunteers. Unfortunately, in this phase we couldn't involve in the experiments senior adults or individuals with cognitive decline.

Results

In this work we focus on the accuracy in recognizing fine-grained abnormal behaviors mostly based on the execution of ADLs. This accuracy is affected by different factors: a) the propagation of errors from noisy sensing devices, b) mistakes in manipulations classification, c) mistakes in ADLs recognition and d) inaccuracy in the modeling of abnormal behaviors rules. Since the accuracy of activity recognition has been extensively evaluated in several previous works, we assume to have an ADL recognition system which is accurate at 100%. This allows us to have a better understanding of which errors are introduced by the anomaly recognition rules engine in combination with a possibly noisy sensing infrastructure. This assumption does not imply completely unrealistic results, since running the ADLs recognition algorithm presented in Chapter 3 on this dataset resulted in an overall F1 score greater than 0.9. A detailed evaluation of the manipulations classification accuracy can be found in Chapter 5. Our preliminary results are summarized in Table 6.3.

True positives, false positives and false negatives are computed by comparing the abnormal behaviors inferred by our rules with the ground truth, since the dataset has been annotated with the occurred anomalies. In this phase of the project we

Anomaly	Precision	Recall	F_1 score
Medicine X not even retrieved	0.79	1.0	0.88
Retrieve Medicine X But Not Opened	0.67	1.0	0.80
Open Medicine X Twice	0.88	0.70	0.78
Wrong Medicine Opened	1.0	0.8	0.89
Object Retake Multiple Times	1.0	1.0	1.0
Wrong Repository	0.9	1.0	0.95
Repository Search	0.89	0.89	0.89
Overall	0.88	0.91	0.89

Table 6.3: Preliminary results of our fine-grained anomaly recognition method considering abnormal behaviors based on objects' manipulations.

focused our attention on anomalies related to medicine packages manipulations and more general anomalies which can occur by manipulating different types of objects.

The first two anomalies in the table describe the scenario where the subject does not take a certain medicine which was prescribed in a particular time interval by the clinicians. In particular, the occurrence of the anomaly "*Medicine X not even retrieved*" implies that the subject did not even retrieve that medicine from its drawer, while "*Retrieve medicine X but not opened*" occurs when the subject retrieves the medicine from the drawer but then he/she forgets to take it. The precision in recognizing these anomalies is negatively affected by mistakes in manipulations classification. Indeed, the manipulation "*accessing the content of medicine box*" is sometimes confused with the manipulation "*significant movement*". In other words, in some cases the system detected that a medicine box was moved from a place to another, while it was actually manipulated to extract the medicine. These mistakes generated a small number of false positives of anomalies regarding skipped intakes of medicines.

The anomaly "*Open Medicine X Twice*" represents the scenario where the subject wrongly takes the same medicine twice within the same prescription time, while the anomaly "*Wrong Medicine Opened*" describes the situation where the subject takes a medicine which is not prescribed at the time of intake. As expected, the previously mentioned mistakes in manipulations detection negatively impact the sensitivity (i.e., the recall value) in recognizing these anomalies. Indeed, since the manipulation "*accessing the content of medicine box*" is sometimes misclassified, the capability of our method of identifying abnormal medicine intakes is affected.

The remaining anomalies are based on more generic manipulations of objects, and they are easier to detect. The anomaly "*Object Retake Multiple Times*" occurs when the subject repeatedly interacts with an object without performing any useful action with it. This anomaly is perfectly captured by our method.

The "*Wrong Repository*" anomaly captures the situation where an object is placed in a not appropriate repository (e.g., the salt is placed in the fridge). The few mistakes in recognizing this anomaly are due to mis-classifications of "*significant movement*" manipulations (i.e., our system was sometimes unable to detect that an object was actually moved to a repository).

Finally, the "*Repository Search*" anomaly describes the scenario where the subject continuously opens and closes the home's repositories several times, without retrieving and returning any item. This anomaly should capture a confusion state of the monitored subject. However, few times the subject normally executed ADLs by opening and closing several times the home's repositories and interacting with objects which were not equipped with wireless accelerometers. Our system wrongly identified these occurrences as anomalies.

The overall results are promising, showing an average F_1 score of almost 0.9. Moreover, the method produced a very low number of false positives. Consider that the dataset consists of over 700 activity instances which included in total over 6.000 sensor events, while the total number of actual instances of abnormal behaviors is just 150. The total number of false positives considering the whole dataset is low, and this means that rarely an abnormal behavior is fired during the normal execution of ADLs (i.e. the true negative rate is very high).

6.3.4 Summary

In this section we showed how, by detecting specific object manipulations, we could significantly refine and improve our previous work on the recognition of fine-grained abnormal behaviors. Our approach is based on the translation of high-level descriptions of abnormal behaviors (provided by domain experts) into first-order logic rules. Detected ADLs, sensor events, recognized manipulations and subject-specific information are translated in logic facts and added to the knowledge base in order to infer fine-grained anomalies performed by the subject. We experimentally evaluated the effectiveness of our method on a large dataset, considering a set of anomalies specifically based on objects manipulations and analyzing how mistakes produced by manipulation detection algorithm proposed in Chapter 5 impacts on the recognition of abnormal behaviors. This method addresses the research question **Q4)** presented in Section 2.5, thus recognizing abnormal behaviors at a fine-grained level.

It is important to note that the knowledge-based approach used to detect anomalies requires the effort of knowledge engineers and domain experts (e.g., clinicians) to formally define the abnormal behaviors. The quality of their work is critical to obtain significant results. Moreover, the anomalies should be defined independently from the specific technological setup, which may vary in different environments.

While the considered anomalies are indicators of possible abnormal behaviors, they are not intended to provide an automatic diagnosis of the patient's cognitive status, especially when they occur in isolation. For instance, the fact that the subject has taken a medicine that was not prescribed is critical if he does it unintentionally (e.g., for a memory disorder). In other cases it may be a normal behavior; e.g., if the patient intentionally takes an over-the-counter drug that does not interfere with his medical prescriptions. Therefore, our system is not intended to provide a diagnosis hypothesis, but simply as a powerful data analysis tool at the service of practitioners reporting the type, frequency, correlation and temporal trend of detected anomalies. By joining this information with other methods and with the subject profile and therapy, it is also possible to set personalized trigger alarms.

Future work includes a) the investigation of probabilistic methods to model and detect certain types of anomalies that are not suited to be modeled by logic, b) the real-time operation of our framework, and c) the acquisition of a dataset from seniors using monitored objects while performing the activities.

6.4 Long-term analysis of abnormal behaviors

The method proposed in the previous section detects abnormal behaviors on a short-term basis. What we propose in this section, instead, is a long-term analysis to detect significant changes in the trend of performing activities and to avoid raising alerts for isolated abnormal activities. We propose a framework called LOTAR, which exploits unobtrusive ADLs and fine-grained anomalies recognition algorithms to automatically detect long-term abnormal behaviors (e.g., changes in habits regarding timing of meal consumption).

6.4.1 Architecture

In Figure 6.2 we show the general architecture of LOTAR. Its core is composed of two main software modules: (i) the first one takes as input the timestamped sensor data and the models of abnormal behaviors, and it returns recognized actions, manipulations, activities and fine-grained anomalies; (ii) the historical behavior analysis module performs historical data analysis to identify long-term abnormal behaviors. The outputs of both fine-grained and long-term abnormal behavior recognition are transmitted to the e-HealthCare service, and can be inspected by clinicians through a Web dashboard.

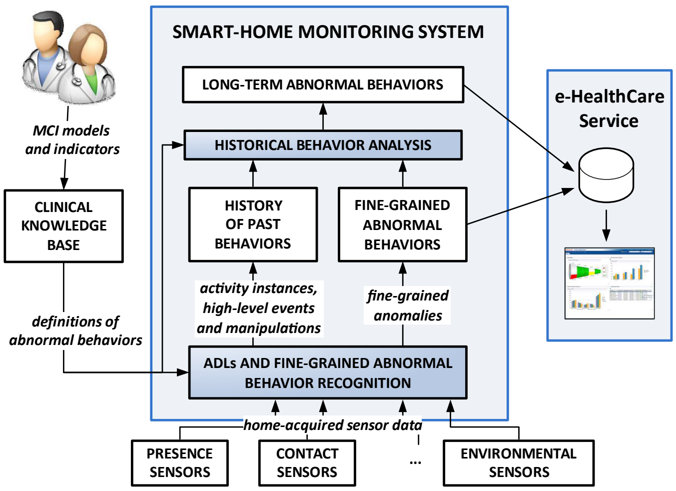


Figure 6.2: The architecture of our long-term analysis framework

6.4.2 Historical behavior analysis

While fine-grained anomalies identify situations that can be precisely specified and effectively detected through symbolic methods, long-term abnormal behaviors are characterized by wide inter- and intra-individual variability; hence, we rely on a personalized statistical approach to detect them. In our approach, we map the daily activities of the patient in *activity feature vectors*, which succinctly describe some characteristics of interest of the activities performed during a given period (e.g., one day). The goal is to statistically monitor the temporal evolution of those vectors to detect significant changes from the patient's usual behavioral pattern.

Building activity feature vectors

Of course, the technique to build activity feature vectors depends on the considered activities, on their characteristics of interest and on the patient's profile. In the following, we illustrate an application of the technique considering meal preparation activities, where the characteristics of interest is the temporal distribution of their occurrences during the day.

Example 6.4.1. In order to represent the distribution of meal preparation activities during a day, we partition the day in k time slots, not necessarily of equal length, and map each occurrence of meal preparation to the time slot in which that activity has ended. Hence, for each day we build an activity feature vector v_i of length k that stores the number of meals prepared during each time slot during day i. For example, if we consider the partition:

0: breakfast	5am - 11am
1: morning	11am - 12noon
2: lunch	12noon - 3pm
3: afternoon	3pm - 6pm
4: dinner	6pm - 10pm
5: night	10pm - 5am

vector $v_i = \langle 102010 \rangle$ means that "during day i, the patient prepared one meal within the breakfast slot, two meals within the lunch slot, and one meal within the dinner slot; he/she did not prepare any other meal during that day".

Mining for long-term abnormal behaviors

In order to detect whether there has been any recent change in the patient's habits, we compare the activity feature vectors of the last n days (called *current period*) with the ones observed in a preceding period of m days (called *baseline period*), with $m \gg n$. Note that there is no intersection between the days in the baseline period and the ones in the current period. We assume that the baseline period represents the usual behavior of the patient in a recent past. A frequent pattern mining [131] algorithm can be applied to the activity feature vectors of the baseline period B to obtain the set V of typical activity routines; i.e., those vectors whose frequency in B is equal to or larger than the support value s. Then, for each day i in the current period C, we check whether the associated vector v_i appears in V or not. If not, we consider day i as *anomalous*. If the rate of anomalous days during C exceeds the threshold t, we detect a long-term anomaly during C and the algorithm returns the set of anomalous days in C.

The algorithm pseudo-code for checking if a long-term abnormal behavior occurred in the current period is shown in Algorithm 4. Note that F is the set of frequent patterns, while N is the set of anomalous days. The function set takes as input a sequence and outputs the set of its elements (without repetitions). The algorithm is executed using a sliding window approach: for instance, each day it is executed considering the last two weeks as the current period, and the previous three months (last two weeks excluded) as the baseline period.

Extensions to consider periodic routines

Based on the individual's profile, the mining algorithm can be refined to take into account periodic habits and routines. For instance, it is possible to divide the days used for the analysis into classes (e.g., working days vs holidays), and apply the algorithm to each class separately to discover changes in periodic routines or abnormal behaviors correlated with them.

Profile-based calibration of parameters

Parameters s and t need to be carefully calibrated based on the patient's habits. In general, increasing the value of s reduces the number of activity feature vectors that are considered normal, and therefore increases the number of days in the current period detected as anomalous. A higher value of t, instead, will make the

ALGORITHM 4: Long-term abnormal behavior detection

Input:

C: set of days of the current period; B: set of days of the baseline period; s: minimum support value for frequent pattern mining; t: threshold for anomalous days in C; S_C, S_B sequences of activities feature vectors associated to the days in C and B, respectively.

$F \leftarrow \emptyset$; $N \leftarrow \emptyset$; $S'_B \leftarrow set(S_B)$
forall $w \in S'_B$ **do**
  if w *appears in* S_B *at least* s *times* **then** $F \leftarrow F \bigcup \{w\}$;
end
forall $v_i \in S_C$ **do**
  if $v_i \notin F$ **then** $N \leftarrow N \bigcup \{i\}$;
end
if $|N| \geq t \cdot |S_C|$ **then**
  return N
else **return** $\emptyset$;

algorithm require a higher portion of abnormal days to output a long-term anomaly. To effectively run the analysis, we need to carefully balance those values, so that we can properly recognize whether the current days are deviating from the baseline activity pattern.

In the following we explain our approach to calibrate s and t values. We fix the value s based on the profile of the patient. If the patient has very regular habits, he/she would tend to execute very frequently a limited set of routines. In this case, a relatively high value of s should be chosen, to include only his/her normal routines in the set of frequent activity feature vectors. On the contrary, a relatively low value of s should be chosen when the patient has not very regular routines, to account for the wide variability of his/her typical activity patterns. The patient's profiling can be done manually by practitioners during the clinical assessment, or by automatically mining a dataset of the typical activity routines of the patient. The value of s should be periodically re-calibrated to account for changes in the patient's habits.

After fixing s, we initially set the value of t to a default value, which is currently manually chosen according to the current cognitive status of the patient. The value of t is periodically re-calibrated considering the clinical assessment of the patient.

6.4.3 Evaluation

We have applied the technique to recognize long-term abnormal behaviors (Algorithm 4) using the real-home dataset presented in Chapter 3 and the meal preparation routines previously discussed. We have used the time slots shown in Example 6.4.1, which were calibrated according to the patient's habits. We have considered a baseline period (B in Algorithm 4) of 30 days from 30 October 2014 to 22 December 2014. We had to skip some days due to temporary failures of the sensor platform used for the data acquisition. For the sake of simplicity, we have considered the days in the test period as consecutive, disregarding skipped days. We have applied our algorithm, with a temporal sliding window of 7 days (C in Algorithm 4), over a test period ranging from 10 January 2015 to 15 February 2015, for a total of 32 days. We have used our profile-based technique for parameter calibration. According to the patient's clinical profile, we have set $s = 2$ and $t = 0.5$. As explained in Section 6.4.2, s must be carefully calibrated according to the personal profile and health status of the subject. In our case, we have chosen a small value for the support s, since the subject exhibited large variability in the execution pattern of activities, probably due to MCI symptoms. The value t ($0 \leq t \leq 1$) determines the sensibility of the long-term recognition algorithm: in our experimentation we have chosen an intermediate value. For the sake of this work and driven by the indications of clinicians, we also divided fine-grained abnormal behaviors in three levels of seriousness: green (e.g.; if a meal is consumed out of the prescribed time), yellow (e.g.; if a meal is skipped), and red (e.g.; if a prescribed medicine is not taken). This classification is orthogonal to the one presented in Section 6.2.

The algorithm detected two long term anomalies, one from 12 January to 23 January, and one from 29 January to 6 February. Those intervals are shown in Figure 6.3 as horizontal bars. The days that were classified as anomalous by the algorithm are colored in violet. According to the choice of parameters, each abnormal interval bar includes at least 4 anomalous days. In order to understand whether those intervals actually correspond to a period characterized by anomalous behaviors, we have identified the days in the overall test period in which the highest number of red anomalies occurred. We found 5 days in which the patient did 7 or more such anomalies (identified by a red square in the figure), while in the other days no more than 5 red anomalies occurred. We can notice that 4 out of 5 among those days are contained in the two intervals. We believe that the correlation is significant, especially considering that our long-term abnormal behavior recognition algorithm considered meal preparation activities, while red anomalies regard medicine intake, which are not related to meal consumption according to the patient's clinical prescriptions. However, we point out that more extensive experiments, carried out with more patients and for longer time periods, are needed to thoroughly assess the effectiveness of the algorithm.

We also computed the long-term trend of the occurrences of detected fine-grained abnormal behaviors, using a simple sliding window approach: for each day in the dataset, we count the number of anomalies detected in the 15 previous days. We used the previously described rule-based fine-grained abnormal behaviors recognition method, considering the anomalies presented in [3]. Figure 6.4 shows a comparison between the results obtained using our technique and the actual ones (i.e., the ground truth). We can notice that, in general, the amount of anomalies detected with our technique is close to the ground truth. Moreover, we can notice that, despite the differences in value, the general trend is preserved; hence, our method provides the clinicians with a reliable tool to recognize significant changes in the rate of anomalies.

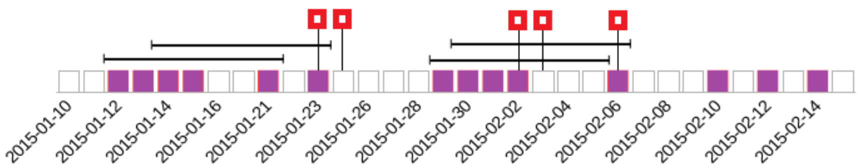
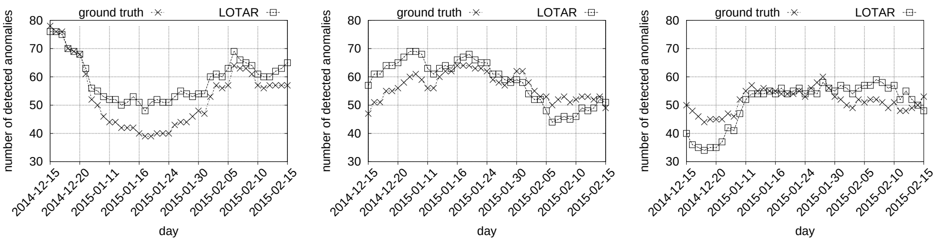


Figure 6.3: Detection of two long-term abnormal behavior intervals with our technique

6.4.4 Summary

We addressed the challenging issue of unobtrusively recognizing long-term abnormal behaviors exhibited by elderly persons at home. The recognition of behavioral anomalies is guided by medical models provided by cognitive neuroscience experts. We have implemented the system and conducted an extensive experimentation, considering a three months deployment in a patient's home.

Even though the achieved results are promising, we consider to improve this



(a) Red anomalies (b) Yellow anomalies (c) Green anomalies

Figure 6.4: Trend of fine-grained abnormal behaviors. For each day, the value represents the number of anomalies detected in the previous 15 days.

work in several directions. We plan to evaluate the effectiveness of long-term abnormal behavior recognition by executing additional experiments with multiple patients and for longer time periods. We will also closely collaborate with clinicians for both identifying anomalies of interest to be monitored, and evaluating the clinical utility of our framework.

It is important to note that after few months that we published our method [32], Dawadi et al. published an independently developed framework to recognize long-term abnormal behaviors [132]. This method is significantly better than our approach, and it has been evaluated on 18 subjects in their homes for an extended period (almost 2 years). In that work, the authors propose to model the behavior of a subject during a time period with an *activity curve*: a model that describes the generalized routine of the inhabitant. In particular, considering a period of d days, each day is divided in m equal-size consecutive windows, and a probability distribution over the activities is defined for each window. An *activity curve* thus represents a compilation of these distributions for the considered d days. Long-term abnormal behaviors occur when an *activity curve* diverges from another one considered as the "*regular behavior*" of the subject.

Chapter 7

Conclusions

7.1 Summary

In this thesis we proposed novel methods to continuously and unobtrusively monitor the ADLs performed by elderly subjects in their homes, with the objective of detecting abnormal behaviors which can be indicators of cognitive decline. We focused on unobtrusive sensing infrastructures, avoiding the privacy issues of video/audio based solutions (unacceptable for many subjects in home environments) and the obtrusiveness of wearable solutions.

We proposed two hybrid ADLs recognition algorithms which addressed some of the known problems of *data-driven* and *knowledge-based* approaches. We also introduced a novel framework to detect the manipulations that the inhabitant performs on household objects, combining cheap sensors and machine learning techniques. Finally we discussed how unobtrusive sensing, objects manipulations detection and ADLs recognition can be combined together to detect fine-grained abnormal behaviors. In particular, we considered abnormal behaviors that may indicate the onset of Mild Cognitive Decline, and more generally of a cognitive decline. The trend and the frequency of those anomalies can be inspected by clinicians in order to support the early diagnosis of cognitive disorders. For this purpose, we also introduced a preliminary investigation on the analysis of long-term abnormal behaviors. Extensive evaluation on several datasets showed the effectiveness of our methods. In the following we summarize the specific contributions introduced in this thesis.

Hybrid ADLs recognition

Our major contributions introduced in this thesis are two novel hybrid ADLs recognition algorithms. The first, proposed in Chapter 3, combines both *data-driven* and *knowledge-based* approaches in order to take advantage of their strengths to improve the recognition rate. In particular, we combined supervised learning with a knowledge-based algorithm to correct statistical mis-predictions and to identify the activities boundaries. Experimental results showed that our method outperforms classic solutions purely based on supervised learning. The main drawback of that approach is that it requires the acquisition of a comprehensive annotated dataset. For this reason in Chapter 4 we proposed an unsupervised approach which combines probabilistic and ontological reasoning. This method overcomes the main limitations of *data-driven* and *knowledge-based* approaches. First of all, it does not require the acquisition of an expensive dataset. Further, the activity model is based on general semantic relations among activities and smart-home infrastructure and the model can be seamlessly reused with different individuals/environments. We exploited ontological reasoning to derive (in an offline phase) semantic correlations between sensor events and activities. A probabilistic reasoning module based on Markov Logic Network is in charge of combining those correlations with the sensor events collected in the home in order to derive the most likely performed activities. We evaluated this method on two different datasets, showing that the recognition rate is comparable to the one obtained by supervised solutions. One of the biggest limitations of this approach is the relevant knowledge engineering which is needed to build a comprehensive ontology. We thus need to investigate with larger scale implementations whether the same ontology can be adequate to cover every possible home environment and individuals' mode of activity execution.

Objects manipulations recognition

Monitoring the interaction of the inhabitant with household items is an important step to accurately detect the ADLs performed by the subject in his/her home. Moreover, clinicians are interested in monitoring *how* objects are manipulated for cognitive assessment. Wearables solutions have been proposed to identify the objects manipulations. However, there is no guarantee that wristbands or pendants are constantly worn and there are also indications of a general adversity or disaffection of users to wearables targeted to healthcare related applications. Computer vision techniques have been proposed as well to track the objects' usage, but cameras are too privacy intrusive in a smart-home environment. To address the problems of existing solutions, in Chapter 5 we introduced a novel framework to detect manipulations performed on everyday objects. To do so, we attached tiny Bluetooth Low Energy accelerometers to the objects of interest. A real-time machine learning algorithm is in charge of continuously segmenting the stream of accelerometer data produced by the objects and to classify each segment with the most likely performed manipulation. We evaluated our system with a dataset consisting of thousands of manipulations performed by several volunteers on three different types of objects. Our results show that a simple segmentation strategy and standard machine learning algorithms already provide acceptable results. However, the usage of Bluetooth Low Energy sensors has several limitations. First of all, the life-time of the battery is not acceptable for a real-home deployment. Moreover, problems arise when these sensors are attached to short-life objects (e.g. food packages) or objects that need to be dipped in water or exposed to high temperatures. We are however confident that these limitations will be solved by technological evolution.

Fine-grained and long-term abnormal behaviors detection

Most of the state-of-the-art approaches to detect abnormal behaviors build a model of the "regular" behavior in order to identify those activity patterns which diverge from the expected ones. However, those methods do not provide a detailed description of the anomalous behavior which can be useful to clinicians to support their diagnosis. To overcome this issue, in Chapter 6 we proposed a novel framework to recognize abnormal behaviors at a fine-grained level. In particular, we considered anomalies related to objects manipulations, taking advantage of the framework presented in Chapter 5. Through the collaboration with neuroscience experts, we considered anomalies which can be indicators of early symptoms of cognitive disorders. We translated the natural language descriptions of the anomalies provided by clinicians to first-order logic rules. Periodically (e.g., daily) sensor events, detected manipulations, recognized ADLs and subject-dependent information (e.g., medical prescriptions) are converted in logic facts and anomaly recognition is performed with a logic programming engine. We evaluated the proposed system on a dataset acquired in a smart lab consisting of hundreds of ADLs and anomalies performed by several volunteers. Preliminary results show that our method generates a low number of false positives, while reaching a promising accuracy. It is important to note that the occurrence of a single fine-grained anomaly is not a direct indicator of cognitive disease, while their frequencies and temporal trend can be used to derive behavioral changes. Hence, we also proposed tools to detect long-term abnormal behaviors and we also studied how the variations of the trend of fine-grained abnormal behaviors could be predictive of anomalous situations. Extensive evaluation with data acquired in real deployments from senior subjects is needed to further validate our approaches.

7.2 Future work

The results presented in this thesis are encouraging, and we plan to further improve our methods by investigating several interesting research directions. In the following we outline the ones we believe are more promising.

Online recognition and active learning

Except for the manipulation recognition algorithm presented in Chapter 5, the proposed methods do not support real-time recognition. However, many important application scenarios require detecting on-the-fly the subject's behavior. For instance, a system to detect dangerous behaviors of the elderly should report the potential danger as it happens, since a delay could put the elderly's safety at risk. In contrast to offline recognition, online recognition is typically harder: it has to deal with a continuous stream of sensor events to be processed in nearly real-time. We thus will extend our ADLs and fine-grained anomalies recognition algorithms in order to realize a behavioral analysis system capable to work in real-time scenarios. In combination with online recognition, we will also investigate semi-supervised and active learning techniques to improve the recognition rate of our methods. In particular, we want to investigate how active learning can be used to personalize and fine-tune the recognition model for a specific subject. Moreover, we want to investigate how similar subjects (e.g., according to physical characteristics or home environments) can be grouped together in order to share the same recognition model and to fine-tune it collaboratively.

Privacy aspects

Smart-homes produce a huge amount of data which is not feasible to store and manage in local gateways. For this reason, sensor events along with detected ADLs and anomalies can be periodically outsourced to cloud servers in order to achieve storage scalability and data availability at reduced costs. Moreover, cloud servers provide efficient and scalable tools to query and process big data. Those tools could be used by clinicians in order to inspect the subject's behavior and to support their diagnosis of cognitive disorders. Unfortunately, many privacy issues arise when data collected from smart-homes is outsourced to untrusted third parties, since the control on such data is lost. Indeed, outsourced data can reveal several sensitive information about the inhabitants: what they are doing and when. Cloud servers are operated by commercial providers which are very likely to be outside of the trusted domain, offering data protection only against outsiders. We plan to investigate a privacy model which allows outsourcing sensitive smart-home data while maintaining its confidentiality and integrity. The model should allow trusted clinicians and caregivers to efficiently perform fine-grained queries on encrypted data in order to inspect the subject's behavior.

Outdoor behavioral monitoring

While in this thesis we only focused on behavioral monitoring in smart-home environments, it is very important to monitor elderly subjects also when they are outside home. The rate of adults older than 65 who own mobile devices (e.g., smart-phones) increased by 24% since 2013. This trend allows designing systems that take advantage of sensors integrated on those devices to monitor the behavior of elderly subjects when they are not at home. We aim to use inertial sensors (i.e., accelerometer, magnetometer and gyroscope) to track the motion patterns and thus the physical activities performed by the subject. Moreover, we want to integrate contextual information (e.g., location, light, weather, ...) to refine activity recognition and at the same time to detect abnormal situations.

Experiments with seniors

The majority of experiments that we presented in this work were performed on datasets acquired in controlled environments (e.g., smart labs). The only dataset acquired in a real-home scenario involved only one subject and few ADLs were observed. In order to further validate our methods, extensive and long-term experiments with real elderly subjects are needed.

Acknowledgments

The first person I would like to thank is my supervisor Claudio Bettini: he patiently guided, supported and believed in me during these three years. His wise guidance has been essential for this work. I've learned a lot from him.

Special thanks go to my co-supervisor Daniele Riboni which carefully guided my first steps in the amazing world of research.

Thanks to Jadwiga Indulska, Juan Ye and Gergely Zaruba for their precious comments which contributed improving this thesis.

I would like to thank Heiner Stuckenschmidt which kindly hosted me at the University of Mannheim for a successful research visit. Thanks to Timo Sztyler: collaborating with you on challenging problems was great!

Thanks to Sergio Mascetti for his support and for the motivational conversations about my next steps in my research career.

Special thanks to all the students that contributed with care to this work (in chronological order): Stefano Belfiore, Claudio Tappia, Riccardo Presotto, Chiara Mariani, Alessandra Plati, Sara Cifola, Alessio Anghileri, Matteo Zampatti, Federica Dondi, Alessandro Sperotti and Luca Tarallo.

I would like to thank Andrea, Christian, Dragan, Letizia and Matteo for their advice while I was deciding whether to do or not the Ph.D. and for their help and suggestions during these years. Thanks also to all the other guys with whom I shared lunches and funny moments: Alessandro, Lorenzo, Marco, Michela, Nadia, Silvia, Stefano.

Thanks to my family which never stopped believing in me. Thanks to my best friend Stefano which has always been there for me to share laughs and tears. Thanks to all the friends who have been at my side despite everything.

And of course, thanks to you, Valeria. With you by my side life is easier and beautiful.

Bibliography

[1] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579–590, 2013.

[2] Centers for Disease Control and Prevention. The state of aging and health in america 2007. *Aging Society*, 2007.

[3] Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Rim Helaoui. Smartfaber: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment. *Artificial Intelligence in Medicine*, 67:57–74, 2016.

[4] Bernd Rechel, Emily Grundy, Jean-Marie Robine, Jonathan Cylus, Johan P Mackenbach, Cecile Knai, and Martin McKee. Ageing in the european union. *The Lancet*, 381(9874):1312–1322, 2013.

[5] George Demiris, Brian K Hensel, Marjorie Skubic, and Marilyn Rantz. Senior residents perceived need of and preferences for smart home sensor technologies. *International journal of technology assessment in health care*, 24(01):120–124, 2008.

[6] James Clawson, Jessica A Pater, Andrew D Miller, Elizabeth D Mynatt, and Lena Mamykina. No longer wearing: investigating the abandonment of personal health-tracking technologies on craigslist. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 647–658. ACM, 2015.

[7] Prafulla N Dawadi, Diane J Cook, and Maureen Schmitter-Edgecombe. Automated cognitive health assessment using smart home monitoring of com-

plex tasks. *IEEE transactions on systems, man, and cybernetics: systems*, 43(6):1302–1313, 2013.

[8] Berndt Winblad, Katie Palmer, Miia Kivipelto, V Jelic, Laura Fratiglioni, L-O Wahlund, A Nordberg, L Bäckman, Michael Albert, O Almkvist, et al. Mild cognitive impairment–beyond controversies, towards a consensus: report of the international working group on mild cognitive impairment. *Journal of internal medicine*, 256(3):240–246, 2004.

[9] Debraj Basu, Giovanni Moretti, Gourab Sen Gupta, and Stephen Marsland. Wireless sensor network based smart home: Sensor selection, deployment and monitoring. In *Sensors Applications Symposium (SAS), 2013 IEEE*, pages 49–54. IEEE, 2013.

[10] Anuroop Gaddam, Subhas Chandra Mukhopadhyay, and Gourab Sen Gupta. Elder care based on cognitive sensor network. *IEEE Sensors Journal*, 11(3):574–581, 2011.

[11] Manfred Huber, Gergely Záruba, Nicholas Brent Burns, and Kathryn Daniel. Smartcare: An introduction. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*, pages 394–400. IEEE, 2017.

[12] Daqing Zhang, Tao Gu, and Xiaohang Wang. Enabling context-aware smart home with semantic web technologies. *International Journal of Human-friendly Welfare Robotic Systems*, 6(4):12–20, 2005.

[13] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[14] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. In *1st International Conference on Pervasive Computing (Pervasive)*, volume 2414 of *Lecture Notes in Computer Science*, pages 167–180. Springer, August 2002.

[15] Alessandra Agostini, Claudio Bettini, and Daniele Riboni. Hybrid reasoning in the CARE middleware for context awareness. *International Journal of Web Engineering and Technology*, 5(1):3–23, 2009.

[16] Karen Henricksen, Steven Livingstone, and Jadwiga Indulska. Towards a hybrid approach to context modelling, reasoning and interoperation. In *Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning And Management, in conjunction with UbiComp*, volume 2004, 2004.

[17] Gabriele Civitarese and Claudio Bettini. Monitoring objects manipulations to detect abnormal behaviors. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*, pages 388–393. IEEE, 2017.

[18] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM, 2008.

[19] Daniele Riboni, Timo Sztyler, Gabriele Civitarese, and Heiner Stuckenschmidt. Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1–12. ACM, 2016.

[20] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and information systems*, 36(3):537–556, 2013.

[21] Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. Collection and curation of a large reference dataset for activity recognition. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 30–35. IEEE, 2011.

[22] Liming Chen and Chris Nugent. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, 5(4):410–430, 2009.

[23] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 345–354. ACM, 2013.

[24] Alistair Burns, Brian Lawlor, and Sarah Craig. Rating scales in old age psychiatry. *The British Journal of Psychiatry*, 180(2):161–167, 2002.

[25] Diane J Cook and Maureen Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of information in medicine*, 48(5):480, 2009.

[26] Enamul Hoque, Robert F Dickerson, Sarah M Preum, Mark Hanson, Adam Barth, and John A Stankovic. Holmes: A comprehensive anomaly detection system for daily in-home activities. In *Distributed Computing in Sensor Systems (DCOSS), 2015 International Conference on*, pages 40–51. IEEE, 2015.

[27] Barnan Das, Diane J Cook, Narayanan C Krishnan, and Maureen Schmitter-Edgecombe. One-class classification-based real-time activity error detection in smart homes. *IEEE journal of selected topics in signal processing*, 10(5):914–923, 2016.

[28] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1):107–136, 2006.

[29] Gabriele Civitarese, Stefano Belfiore, and Claudio Bettini. Let the objects tell what you are doing. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 773–782. ACM, 2016.

[30] Liming Chen, Chris D Nugent, and Hui Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):961–974, 2012.

[31] Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Rim Helaoui. Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment. In *Proc. of PerCom'15*, pages 149–154. IEEE Comp. Soc., 2015.

[32] Daniele Riboni, Gabriele Civitarese, and Claudio Bettini. Analysis of long-term abnormal behaviors for early detection of cognitive decline. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[33] Michel C. A. Klein, Nataliya M. Mogles, and Arlette van Wissen. Intelligent mobile support for therapy adherence and behavior change. *J Biomed Inform*, 51:137–151, 2014.

[34] Carolyn Parsey and Maureen Schmitter-Edgecombe. Applications of technology in neuropsychological assessment. *The Clinical Neuropsychologist*, 27(8):1328–1361, 2013.

[35] Juan Ye, Simon Dobson, and Susan McKeever. Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, 8(1):36–66, 2012.

[36] Aharon Satt, Alexander Sorin, Orith Toledo-Ronen, Oren Barkan, Ioannis Kompatsiaris, Athina Kokonozi, and Magda Tsolaki. Evaluation of speech-based protocol for detection of early-stage dementia. In *Proc. of INTERSPEECH*, pages 1692–1696. ISCA, 2013.

[37] Somboon Hongeng, Ram Nevatia, and Francois Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, 2004.

[38] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive Computing: Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings*, pages 1–17, Berlin, Heidelberg, 2004. Springer.

[39] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33:1–33:33, 2014.

[40] Timo Sztyler and Heiner Stuckenschmidt. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9, Washington, D.C., 2016. IEEE Computer Society.

[41] James Clawson, Jessica A. Pater, Andrew D. Miller, Elizabeth D. Mynatt, and Lena Mamykina. No longer wearing: Investigating the abandonment of personal health-tracking technologies on craigslist. In *Proceedings of*

the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '15, pages 647–658, 2015.

[42] Tim van Kasteren, Athanasios K. Noulas, Gwenn Englebienne, and Ben J. A. Kröse. Accurate activity recognition in a home setting. In *Proc. of UbiComp*, volume 344 of *ACM ICPS*, pages 1–9. ACM, 2008.

[43] Christopher Richard Wren and Emmanuel Munguia Tapia. Toward scalable activity recognition for sensor networks. In *LoCA*, volume 3987, pages 168–185. Springer, 2006.

[44] Kyoung Nam Ha, Kyung Chang Lee, and Suk Lee. Development of pir sensor based indoor location detection system for smart home. In *SICE-ICASE, 2006. International Joint Conference*, pages 2162–2167. IEEE, 2006.

[45] Tamara L. Hayesa, Francena Abendroth, Andre Adami, Misha Pavel, Tracy A. Zitzelberger, and Jeffrey A. Kaye. Unobtrusive assessment of activity patterns associated with mild cognitive impairment. *Alzheimers Dement*, 4(6):395–405, 2008.

[46] Shwetak N Patel, Matthew S Reynolds, and Gregory D Abowd. Detecting human movement by differential air pressure sensing in hvac system ductwork: An exploration in infrastructure mediated sensing. In *International Conference on Pervasive Computing*, pages 1–18. Springer, 2008.

[47] Henrique Serra, João Correia, António J Gano, António M de Campos, and Isabel Teixeira. Domestic power consumption measurement and automatic home appliance detection. In *Intelligent Signal Processing, 2005 IEEE International Workshop on*, pages 128–132. IEEE, 2005.

[48] Antimo Barbato, Antonio Capone, Marta Rodolfi, and Davide Tagliaferri. Forecasting the usage of household appliances through power meter sensors for demand management in the smart grid. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 404–409. IEEE, 2011.

[49] Jon Froehlich, Eric Larson, Elliot Saba, Tim Campbell, Les Atlas, James Fogarty, and Shwetak Patel. A longitudinal study of pressure sensing to infer real-world water usage events in the home. In *International Conference on Pervasive Computing*, pages 50–69. Springer, 2011.

[50] James Fogarty, Carolyn Au, and Scott E Hudson. Sensing from the basement: a feasibility study of unobtrusive and low-cost home activity recognition. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 91–100. ACM, 2006.

[51] Jianxin Wu, Adebola Osuntogun, Tanzeem Choudhury, Matthai Philipose, and James M Rehg. A scalable approach to activity recognition based on object use. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[52] Matthai Philipose, Kenneth P Fishkin, Mike Perkowitz, Donald J Patterson, Dieter Fox, Henry Kautz, and Dirk Hahnel. Inferring activities from interactions with objects. *IEEE pervasive computing*, 3(4):50–57, 2004.

[53] Donald J Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, pages 44–51. IEEE, 2005.

[54] Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Viola Bulgari. From lab to life: Fine-grained behavior monitoring in the elderly's home. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, pages 342–347, Washington, D.C., 2015. IEEE Computer Society.

[55] M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele. ADL recognition based on the combination of RFID and accelerometer sensing. In *Proceedings of Pervasive Health 2008*, pages 2237–2242, Washington, D.C., 2008. IEEE Computer Society.

[56] Xin Hong, Chris D. Nugent, Maurice D. Mulvenna, Suzanne Martin, Steven Devlin, and Jonathan G. Wallace. Dynamic similarity-based activity detection and recognition within smart homes. *Int. J. Pervasive Computing and Communications*, 8(3):264–278, 2012.

[57] Derek Hao Hu, Sinno Jialin Pan, Vincent Wenchen Zheng, Nathan Nan Liu, and Qiang Yang. Real world activity recognition with multiple goals. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 30–39. ACM, 2008.

[58] Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–9. IEEE, 2009.

[59] Yongmian Zhang, Yifan Zhang, Eran Swears, Natalia Larios, Ziheng Wang, and Qiang Ji. Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2468–2483, 2013.

[60] Jiahui Wen, Seng Loke, Jadwiga Indulska, and Mingyang Zhong. Sensor-based activity recognition with dynamically added context. In *proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015.

[61] Diane J. Cook, Kyle D. Feuz, and Narayanan Chatapuram Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and Information Systems*, 36(3):537–556, 2013.

[62] Seng W. Loke. Representing and reasoning with situations for context-aware pervasive computing: A logic programming perspective. *The Knowledge Engineering Review*, 19(3):213–233, 2004.

[63] Daniele Riboni and Claudio Bettini. Owl 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing*, 7(3):379–395, 2011.

[64] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung. Ontology Based Context Modeling and Reasoning using OWL. In *Proceedings of Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 18–22, Washington, D.C., 2004. IEEE Computer Society.

[65] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory,*

Implementation and Applications. Cambridge University Press, New York, NY, USA, 2nd edition, 2010.

[66] Anand Ranganathan, Jalal Al-Muhtadi, and Roy H. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2):62–70, 2004.

[67] Xin Hong, Chris D. Nugent, Maurice D. Mulvenna, Suzanne Martin, Steven Devlin, and Jonathan G. Wallace. Dynamic similarity-based activity detection and recognition within smart homes. *International Journal of Pervasive Computing and Communications*, 8(3):264–278, 2012.

[68] Georgios Meditskos, Efstratios Kontopoulos, and Ioannis Kompatsiaris. Knowledge-driven activity recognition and segmentation using context connections. In *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II*, pages 260–275, Cham, 2014. Springer.

[69] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, 10, Part B:155–172, 2014.

[70] Juan Ye and Graeme Stevenson. *Semantics-Driven Multi-user Concurrent Activity Recognition*, pages 204–219. Springer International Publishing, Cham, 2013.

[71] Daniele Riboni and Claudio Bettini. COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[72] Georgios Meditskos, Efstratios Kontopoulos, and Ioannis Kompatsiaris. Redef: Context-aware recognition of interleaved activities using owl 2 and defeasible reasoning. In *Joint Proceedings of SSN-TC and OrdRing 2015*, volume 1488 of *CEUR Workshop Proceedings*, pages 31–42, Bethlehem, Pennsylvania, United States, 2015. CEUR-WS.org.

[73] Georgios Meditskos, Efstratios Kontopoulos, and Ioannis Kompatsiaris. Knowledge-driven activity recognition and segmentation using context connections. In *Proc. of ISWC, Part II*, volume 8797 of *LNCS*, pages 260–275. Springer, 2014.

[74] Matthew Richardson and Pedro Domingos. Markov logic networks. *Mach Learn*, 62:107–136, 2006.

[75] Juan Ye, Graeme Stevenson, and Simon Dobson. Usmart: An unsupervised semantic mining activity recognition technique. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(4):16:1–16:27, 2014.

[76] Femke De Backere, Femke Ongenae, Floris Van Den Abeele, Jelle Nelis, Pieter Bonte, Eli Clement, Matthew Philpott, Jeroen Hoebeke, Stijn Verstichel, Ann Ackaert, and Filip De Turck. Towards a social and context-aware multi-sensor fall detection and risk assessment platform. *Comput Biol Med*, 64:307–320, 2015.

[77] B. Winblad et Al. Mild cognitive impairment - beyond controversies, towards a consensus. *J Intern Med*, 256(3):240–246, 2004.

[78] Ahn Inn Sook, Kim Ji-Hae, Kim Seonwoo, Chung Won Jae, Kim Hyeran, Shin Hyo Shin Kang, and Kim Doh Kwan. Impairment of instrumental activities of daily living in patients with mild cognitive impairment. *Psychiatry Investig*, 6(3):180–184, 2009.

[79] G. Sacco, V. Joumier, N. Darmon, A. Dechamps, A. Derreumaux, J. H. Lee, J. Piano, N. Bordone, A. Konig, B. Teboul, R. David, O. Guerin, F. Bremond, and P. Robert. Detection of activities of daily living impairment in alzheimer's disease and mci using ict. *Clin Interv Aging*, 7:539–549, 2012.

[80] Carlos Fernando Crispim-Junior, Veronique Joumier, Yu-Liang Hsu, Pau-Choo Chung, Arnaud Dechamps, Ming-Chyi Pai, Philippe Robert, and Franois Bremond. Alzheimer's patient activity assessment using different sensors. *Gerontechnology*, 11(2):266–267, 2012.

[81] Anastasios Karakostas, Georgios Meditskos, Thanos G. Stavropoulos, Ioannis Kompatsiaris, and Magda Tsolaki. A sensor-based framework to support clinicians in dementia assessment: The results of a pilot study. In *Proc. of ISAmI*, volume 376 of *Advances in Intelligent Systems and Computing*, pages 213–221. Springer, 2015.

[82] ROSETTA-project. Robot control for skilled execution of tasks in natural interaction with humans; based on autonomy, cumulative knowledge

and learning. http://www.fp7rosetta.org/, Project reference: 230902, Funded under: FP7-ICT, 2009–2013.

[83] MARIO-project. Managing active and healthy aging with use of caring service robots. http://www.mario-project.eu/, Project reference: 643808, Funded under: H2020-EU.3.1., 2015–2018.

[84] Heike Felzmann, Kathy Murphy, Dympna Casey, and Oya Beyan. Robot-assisted care for elderly with dementia: Is there a potential for genuine end-user empowerment? In *Proc. of ACM/IEEE International Conference on Human-Robot Interaction Workshops*. IEEE, 2015.

[85] Dem@Care-project. Dementia ambient care - multi-sensing monitoring for intelligent remote management and decision support. http://www.demcare.eu/, Project reference: 288199, Funded under: FP7-ICT, 2011–2015.

[86] Adriana M. Seelye, Maureen Schmitter-Edgecombe, Diane J. Cook, and Aaron Crandall. Naturalistic assessment of everyday activities and prompting technologies in mild cognitive impairment. *J Int Neuropsychol Soc*, 19(4):442–452, 2013.

[87] Prafulla N. Dawadi, Diane J. Cook, Maureen Schmitter-Edgecombe, and Carolyn Parsey. Automated assessment of cognitive health using smart home technologies. *Technol Health Care*, 21(4):323–343, 2013.

[88] Prafulla Dawadi, Diane J. Cook, and Maureen Schmitter-Edgecombe. Automated cognitive health assessment using smart home monitoring of complex tasks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(6):1302–1313, 2013.

[89] Prafulla Dawadi, Diane J. Cook, Carolyn Parsey, Maureen Schmitter-Edgecombe, and Miya Schneider. An approach to cognitive assessment in smart home. In *Proc. of the 2011 DMMH Workshop*. ACM, 2011.

[90] Diance J. Cook and Maureen Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods Inf Med*, 48(5):480–485, 2009.

[91] Gilles Virone, Norbert Noury, and Jacques Demongeot. A system for automatic measurement of circadian activity deviations in telemedicine. *IEEE Trans Biomed Engineering*, 49(12):1463–1469, 2002.

[92] Marjorie Skubic, Rainer Dane Guevara, and Marilyn Rantz. Testing classifiers for embedded health assessment. In *Proceedings of ICOST*, volume 7251 of *Lecture Notes in Computer Science*, pages 198–205. Springer, 2012.

[93] Florence Duchêne, Catherine Garbay, and Vincent Rialle. Learning recurrent behaviors from heterogeneous multivariate time-series. *Artif Intell Med*, 39(1):25–47, 2007.

[94] Vikramaditya Jakkula and Diane J. Cook. Anomaly detection using temporal data mining in a smart home environment. *Methods Inf Med*, 47(1):70–75, 2008.

[95] Parisa Rashidi and Diane J. Cook. COM: A method for mining and monitoring human activity patterns in home-based health monitoring systems. *ACM Trans Intell Syst Technol*, 4(4):64, 2013.

[96] Jose M. Juarez, Jose M. Ochotorena, Manuel Campos, and Carlo Combi. Spatiotemporal data visualisation for homecare monitoringof elderly people. *Artif Intell Med*, to appear, available online 14 June 2015.

[97] Jianxin Wu, Adebola Osuntogun, Tanzeem Choudhury, Matthai Philipose, and James M. Rehg. A scalable approach to activity recognition based on object use. In *Proc. IEEE International Conference on Computer Vision (ICCV 07)*, pages 1–8. IEEE Press, 2007.

[98] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 345–354, New York, NY, USA, 2013. ACM.

[99] Narayanan Chatapuram Krishnan and Diane J. Cook. Activity recognition on streaming sensor data. *Pervasive Mob Comput*, 10:138–154, 2014.

[100] M. Richardson and P. Domingos. Markov logic networks. *Mach Learn*, 62:107–136, 2006.

[101] Leo Breiman. Random forests. *Mach Learn*, 45(1):5–32, 2001.

[102] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSp Magazine*, 3(1):4–16, 1986.

[103] Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. Collection and curation of a large reference dataset for activity recognition. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 30–35, Anchorage, Alaska, USA, 2011. IEEE.

[104] Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*, pages 40:1–40:8, New York, NY, USA, 2012. ACM.

[105] Liming Chen and Chris Nugent. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, 5(4):410–430, 2009.

[106] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, and Ulrike Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics*, 6(4):309–322, 2008.

[107] Christopher W. Geib and Robert P. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, 2009.

[108] Paulito Palmes, Hung Keng Pung, Tao Gu, Wenwei Xue, and Shaxun Chen. Object relevance weight pattern mining for activity recognition and segmentation. *Pervasive and Mobile Computing*, 6(1):43 – 57, 2010.

[109] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th National Conference on Artificial Intelligence*, volume 1, pages 21–27, California, USA, 2005. AAAI Press.

[110] Jakob Huber, Christian Meilicke, and Heiner Stuckenschmidt. Applying markov logic for debugging probabilistic temporal knowledge bases. In *AKBC 2014: 4th Workshop on Automated Knowledge Base Construction*

AKBC 2014 at NIPS 2014 in Montreal, Canada, December 13, 2014, pages 1–6, New York, NY, USA, 2014. ACM.

[111] Melisachew Chekol, Jakob Huber, Christian Meilicke, and Heiner Stuckenschmidt. Markov logic networks with numerical constraints. In *22st European Conference on Artificial Intelligence (ECAI2016)*, pages 1–9, Amsterdam, The Netherlands, 2016. IOS Press.

[112] Geetika Singla, Diane J Cook, and Maureen Schmitter-Edgecombe. Tracking activities in complex settings using smart environment technologies. *International journal of biosciences, psychiatry, and technology (IJBSPT)*, 1(1):25–35, 2009.

[113] Diane J. Cook, Aaron S. Crandall, Brian L. Thomas, and Narayanan C. Krishnan. Casas: A smart home in a box. *Computer*, 46(7):62–69, 2013.

[114] Michael Beigl, Hans-w Gellersen, and Albrecht Schmidt. MediaCups : Experience with Design and Use of Computer- Augmented Everyday Artefacts. *Computer Networks*, 35:401–409, 2001.

[115] Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC '05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, Washington, DC, USA, 2005. IEEE Computer Society.

[116] Oliver Amft and Gerhard Tröster. Recognition of dietary activity events using on-body sensors. *Artificial intelligence in medicine*, 42(2):121–36, mar 2008.

[117] Ifat Afrin Emi and John A Stankovic. SARRIMA : Smart ADL Recognizer and Resident Identifier in Multi-resident Accommodations. In *Wireless Health '15*, 2015.

[118] Aftab Khan, Sebastian Mellor, Eugen Berlin, Robin Thompson, Roisin McNaney, Patrick Olivier, and Thomas Plötz. Beyond activity recognition: Skill assessment from accelerometer data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 1155–1166. ACM, 2015.

[119] Liming Chen, Chris D. Nugent, and Hui Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Trans. Knowl. Data Eng.*, 24(6):961–974, 2012.

[120] E. Hoque, R. F. Dickerson, S. M. Preum, M. Hanson, A. Barth, and J. A. Stankovic. Holmes: A comprehensive anomaly detection system for daily in-home activities. In *2015 International Conference on Distributed Computing in Sensor Systems*, pages 40–51, June 2015.

[121] B. Das, D. J. Cook, N. C. Krishnan, and M. Schmitter-Edgecombe. One-class classification-based real-time activity error detection in smart homes. *IEEE Journal of Selected Topics in Signal Processing*, 10(5):914–923, Aug 2016.

[122] Jhun-Ying Yang, Jeen-Shing Wang, and Yen-Ping Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern recognition letters*, 29(16):2213–2220, 2008.

[123] Timo Sztyler and Heiner Stuckenschmidt. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*, pages 1–9. IEEE, 2016.

[124] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.

[125] T. Giovannetti, B. M. Bettcher, L. Brennan, D. J. Libon, M. Burke, K. Duey, C. Nieves, and D. Wambach. Characterization of everyday functioning in mild cognitive impairment: A direct assessment approach. *Dement Geriatr Cogn Disord*, 25:359–365, 2008.

[126] Sarah C. Seligman, Tania Giovannetti, John Sestito, and David J. Libon. A new approach to the characterization of subtle errors in everyday action: Implications for mild cognitive impairment. *Clin Neuropsychol*, 28:97–115, 2014.

[127] Myrna F Schwartz, Mary Segal, Tracy Veramonti, Mary Ferraro, and Laurel J Buxbaum. The naturalistic action test: A standardised assessment for everyday action impairment. *Neuropsychological rehabilitation*, 12(4):311–339, 2002.

[128] Zaffar Haider Janjua. *Sensor-based Behavioral Monitoring for Early Detection of Cognitive Decline.* PhD dissertation, Università degli Studi di Milano, 2016.

[129] Alistair Burns, Brian Lawlor, and Sarah Craig. Rating scales in old age psychiatry. *Br J Psychiatry*, 180:161–7, 2002.

[130] A.M. Troyer and K.J. Murphy. Memory for intentions in amnestic mild cognitive impairment: Time- and event-based prospective memory. *J Int Neuropsychol Soc*, 13(2):365–369, 2007.

[131] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Min. Knowl. Discov.*, 15(1):55–86, 2007.

[132] Prafulla N Dawadi, Diane J Cook, and Maureen Schmitter-Edgecombe. Modeling patterns of activities using activity curves. *Pervasive and mobile computing*, 28:51–68, 2016.