Table of contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

In this thesis, we present a research contribution which takes the form of a data-driven learning model for unsupervised reasoning in the Smart Home. In this chapter, we first present motivation and context concerning the issue addressed in this thesis. Then we summarize existing works and point out their limitations, which leads to thesis achievement. At the end of the chapter, we offer a brief summary of our contributions and introduce research objectives and methodology.

1.1 Research context: aging population and the need for new technology

Population aging represents an important and growing preoccupation for most governments around the world, which fear its fast progression and catastrophic economical and social effects [14]. Since age is the main risk factor in the manifestation of dementia, it is also fair to say that the number of cases of cognitive impairment will increase with aging population. At this time, the number of people with dementia in the world is estimated at 35.6 million, and this number is predicted to double by 2030 [1]. The main cause of dementia is Alzheimer's disease (AD), which can be defined as a neurodegenerative illness characterized by a progressive decline of cognitive functions. It results in memory impairment, aphasia, agnosia, apraxia and executive dysfunctions [15]. These physical deficits lead to a need for assistance in performing the activities of daily living (ADL) [16]. ADLs can be instrumental (e.g. cooking, driving and handling personal finances) or basic (e.g. eating, dressing and bathing) and the capacity to complete instrumental ADLs decreases in the early stages of AD. In the middle phase of AD progression, some deterioration in basic ADLs occurs, as well as a bigger loss of instrumental ADLs. As the disease gets worse, patients lose more and more skills and become more dependent on caregivers. It has an impact on patients because the loss of autonomy can lead to frustration, apathy or depression, but it also has an impact on caregivers for which the physical, financial and social burdens increase [17]. These factors often force patients to be institutionalized, even if most of them wish to stay home as long as possible [1]. Since the cost associated with AD is very high [18] and resources are lacking (e.g. medical staff), governments want to prolong the time people with AD can remain in their homes.

1.2 Smart Homes as a possible solution

One of the promising solutions to help facing the current generation's challenge on this matter concerns the development of technological tools for automatically assisting elders in their residence, in what we call a Smart Home [19]. Smart Homes have recently become a very active trend in research, bringing hope to the effort to postpone the institutionalization of the elderly [20]. A Smart Home can be defined as an environment enhanced by various types of miniaturized sensors (electromagnetic, infrared, radio-identification tags, etc.) and prompting devices (lights, speakers, IPad, etc.) embedded in everyday life objects (refrigerator, cabinets' doors, etc.). It can provide assistance services by making decisions and effectively prompting the resident (by giving hints, suggestions and reminders) while remaining as less intrusive as possible [21]. The Smart Home is based on the emerging concept of ambient intelligence [22], which refers to a multidisciplinary approach that consists in enriching a common environment (room, building, car, etc…) with technology - mainly composed of sensors - in order to build a system that makes useful decisions which benefit the users of this environment based on real-time information and historical data gathered from a large set of ubiquitous sensors.

One of the main challenges related to this Smart Home concept concerns the huge amount of data resulting from the observation of the ongoing activities through smart sensors [23]. In this sense, a Smart Home can be seen as a challenging big data warehouse in need of an efficient automated computational way to interpret the sensors' data in order to provide high level information about the home's state of normality and needed assistance. More specifically, temporal data requires analysis in order to extract behavioral patterns which can be then used by the Smart Home system for recognizing ongoing activities, predicting subsequent actions, determining a potential situation where assistance is needed and identifying the right moment to provide proper guidance or reminders [12]. This issue is key in developing Smart Home technologies and refers to a very prolific field of artificial intelligence which is called data mining [24] or, more specifically, temporal data mining [25].

1.3 Problem: data mining in the Smart Home

Data mining (DM), also commonly known as knowledge discovery in databases, is a process that uses a variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions [26]. More specifically, it refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases [27]. While data mining and knowledge discovery in databases (or KDD) are frequently treated as synonyms, data mining is actually only a part of the knowledge discovery process. The following figure (Figure 1.1) shows data mining as a step in an iterative knowledge discovery process.

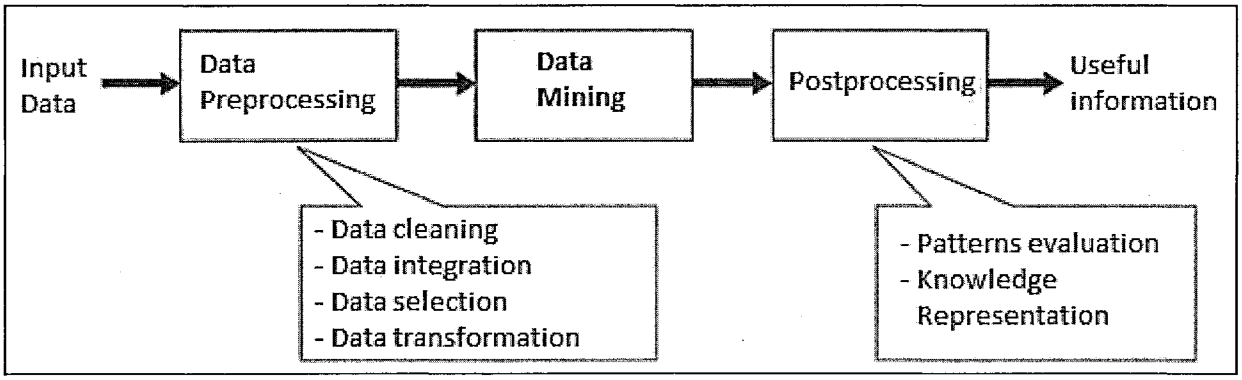


Figure 1.1.: The process of knowledge discovery in a database

As one can see in Figure 1.1, a typical KDD process is composed of several important steps [28] allowing a huge quantity of raw input data to be analyzed and interpreted in order to extract useful high-level information (patterns). The first step of this process is *data cleaning*, which is also known as data cleansing. It is the first phase in which noise and irrelevant data are removed from the collection. The second step is called *data integration*. This maneuver is required to create a single common data source if input comes from multiple heterogeneous data sources. The third step of the process is *data selection*, which consists in retrieving data relevant to the analysis from the collection. Once the right data is retrieved, the next step is *data transformation*, also known as *data consolidation*. This is a phase in which selected data is transformed into forms appropriate for the mining procedure. Only after that can one proceed to the data-mining phase, which is a crucial step in which appropriate techniques are applied to extract patterns that may be potentially useful. Finally, after the data-mining process, two post-processing phases remain. The first one is *pattern evaluation*. It consists in identifying the truly interesting patterns representing knowledge based on some interestingness measures. The second one is *knowledge presentation*, where visualization and knowledge representation techniques are used to present the mined knowledge to the user. This phase is not required if the data resulting from the mining process is directly passed to a program.

In our specific context, the Smart Home can be seen as a warehouse storing a variety of data about the resident, coming from different sensors [12]. On one hand, a huge amount of data which includes information about activities is produced, and, on the other hand, we aim to decrease the role of human expertise in knowledge provision in order to achieve more automation in the Smart Home; therefore, data mining (DM) techniques are extended for making smarter homes [23, 29-31]. The complexity of daily activity recognition in a Smart Home is due to the large number of activities that an occupant can perform. This complexity initially causes a problem in creating models of activities, an essential step in the process of activity recognition where we find not only all the activities that an occupant usually performs, but also the various actions that compose them [32]. The discovery of all frequent activities performed by a Smart Home occupant is the first task that must be tackled to constitute an activity database. It is by exploiting knowledge that the Smart Home is able to recognize the activity which an occupant is trying to accomplish and can try to make predictions about the future actions, which is a key component from an assistive point of view. Data mining allows the creation of an activity database by extracting frequent patterns from the Smart Home sensor history log. Using DM techniques is important because it creates an activity database tailor-made for the specific occupant. However, knowing how the occupant realizes his activities is not enough. Temporal relationships must also be explored in order to know which activity is being performed and when the occupant needs assistance. For instance, if it is detected that the occupant is boiling water, knowing the time and duration of this action gives precious information about what he or she is trying to do by comparing this data with an activity knowledge base. If water is heated for two minutes, then he or she might be making tea, but if it is boiled for 15 minutes, making pasta would be a better guess. The use of temporal information for pattern mining in Smart Homes constitutes a very important issue that only few teams of scientists have tried to address [23].

1.4 Summary of related works on temporal data mining in the Smart Home

Data-mining methods may be categorized as either supervised or unsupervised [25]. In unsupervised methods, no target variable is identified as such. Instead, the data-mining algorithm searches for patterns and structure among all variables[30].

Most data-mining approaches developed for activity recognition in Smart Homes are, however, supervised methods [23, 25, 33]. This means that (i) there is a particular pre-specified target variable, and (ii) the algorithm is given many examples where the value of the target variable is provided, so that the algorithm may learn which values of the target variable are associated with which values of the predictor variables [30]. In this kind of supervised method, it is assumed that we know the form of activities in terms of sensor events and, therefore, we can use training examples to train the model [34]. This requires that we have access to annotated data where each sensor event appearing in the data is labeled with its appropriate activity name. The supervised data-mining approaches for activity recognition range from simple methods such as Naive Bayes [35], based on sensor events independence assumption, to more recent and sophisticated methods such as conditional random fields [36] that model sensor events as probabilistic sequences. Other no suitable supervised methods include decision trees [37], Markov models [38], and Dynamic Bayes Networks [33]. There are a number of problems with supervised approaches. Firstly, the assumption of consistent predefined activities does not hold in reality. Due to physical, mental, cultural and lifestyle differences [39], not all individuals perform the same set of tasks. Even for the same predefined activity, different residents might perform it in various ways, making the reliance on a list of predefined activities impractical due to inter-subject variability. Therefore, data needs to be annotated for each individual and task. However, annotating and hand labeling data is a very time-consuming and laborious task. Consequently, unsupervised approaches seem to be more suitable for activity recognition in a normal day-to-day setting [30].

In contrast to supervised methods, unsupervised methods require no labeled data; rather, they look for interesting patterns in data. There are some works that have explored unsupervised data mining for activity recognition in Smart Homes such as [40] looking for frequent sensor sequences; [41] in mining discontinuous activity patterns and in the work of [42], mixed frequent and periodic activity patterns have been detected. Most of these approaches either do not discover discontinuous patterns, as well as patterns whose order varies from occurrence to occurrence; however, the erratic nature of human activities requires a method that is able to identify discontinuous patterns and, also, their variations. For example, [43] found that variation in overall activity level at home was correlated with mild cognitive impairment. This highlights the fact that it is important for an assistive Smart Home system to be able to recognize and monitor all of the daily activities and their variations which are performed regularly by an individual in his or her environment.

Finally, the information log provided by sensors in a Smart Home can be of different natures (Boolean, logical, temporal, spatial, etc.). All of these kinds of information can be used for pattern mining in activity recognition, but it is clear that the temporal information is highly valuable for characterizing human behavior [44]. In our literature review, we found only few teams, such as [23, 32, 45], who tried to exploit this temporal information (duration, delay, action ordering, etc.) for that purpose. Most of them, such as the well-known, and probably one of the most reputed in the field, work of Cook's [23] based is mining temporal approach on the Allen's temporal framework [5], which is not robust enough for our context, which is ambiguous in the sense that it allows a large variability in the interval of time between boundary, and which does not take into account the fundamental aspects of human behavior related to the normal/abnormal duration of an action and the normal/abnormal delay between two actions.

1.5 Contributions of this thesis

The contribution of this thesis follows in the footsteps of temporal data mining and activity recognition approaches that have been cited in an earlier section. This thesis takes a step forward by providing answers to the questions which have arisen that are related to temporal data mining in the Smart Home for cognitive assistance. The contribution is thus threefold: theoretical, practical and experimental.

On a theoretical level, the originality of this thesis relies on the proposition of a new unsupervised temporal data-mining model [12] for activity recognition addressing the problem of current temporal approaches based on Allen's framework. This new model incorporates some elements of fuzzy logic [6] in order to take into account the imprecision in the realization of an activity of daily living by a resident. More specifically, we propose an extension of the fuzzy-clustering technique in order to group together observations based on similarity degrees between observations, so that the activities are modeled and recognized. Hence, two viewpoints about activities are proposed. In the first, activities are regarded as a series of temporal *fuzzy events* that occur in the Smart Home environment. A fuzzy event is inferred when an action is accomplished in the environment. By this event-driven viewpoint, it is expected that realization of similar activities leads to occurrence of similar-event series. Therefore, in order to perform activity recognition, we propose an activities mining model which is useful for finding a series of events which is similar to current observations. The second viewpoint concerning activities refers to a sort of complementary extension of the first viewpoint. Through this viewpoint, we regard activities as fuzzy concepts formalized as multiple regression function that their forming entities are the fuzzy events and each fuzzy function of the activity is dependent to a set of role-playing variables which traverse of the fuzzy events in realization of the activities. In others words, we propose a formal framework based on the multivariable function $f_a(\alpha, s)$ where α_i indicates the weight factor of the role played by the i^{th} sensor s_i in activity a. The goal of this mined function is to recognize activities and, to do that, it projects the observations of each sensor s_i into the fuzzy activity space. For instance, if we consider the positions of the *glass* and *sugar* objects during the realization of the *coffee-making* activity, by transferring observations to the activity space, the activity-recognition reasoning system makes hypotheses and concepts around these. These hypotheses are summarized in a smoothing curve or line that runs across the fuzzy states of the *coffee-making* activity. At recognition time, it checks how close (similar) the observations are to their smoothing curve of the *coffee-making* function. Finally, activities are ranked based on the inferred similarities to the observations. By this viewpoint, we can reason the occurrence of two or more simultaneous concepts (activities) [46]. In order to perform activity recognition, the mining model is applied to find possible combinations of the known concepts that may better explain the observations. Hence, around a single reality several hypotheses are made and the hypothesis which matches reality best is selected as the most possible ongoing activity concept. As a main element of the originality of our contribution, we propose to transform the recognition problem into a multivariable mining activity function on role playing sensors. From a practical and experimental standpoint, we have implemented this new model to evaluate how it would perform in a realistic context. To achieve that, we used MATLAB software [47] as a simulation platform to test the proposed model. Moreover, a Visual Studio .Net environment and Microsoft SQL server have also been used. The reason for selecting MATLAB as the main simulation environment is that by transforming the format of the initial primary data (sensed data) into matrix format, we could benefit from the matrix operators of MATLAB and extract knowledge in matrix format, which lets us apply and represent knowledge in various forms. We then performed a series of tests, taking the form of several case studies related to common activities of daily living, in order to show the functionality and efficiency of the proposed temporal data-mining approach for real world cases and, especially, the activity recognition application. The proposed thesis also has some advantages which enable more practical recognition of activities. For example, it needs relatively fewer training samples rather than other samples; it welcomes application of more sensors and processes many features of the activities; it addresses sensor imprecision and uncertainty in the human behavior; it considers and analyzes multiple types of data sourced from different types of sensors; it takes the temporal features of activities into account, and it makes inference and reasoning independent from the physical layer (hardware structure and sensors' positions). The aforementioned contributions and advantages have directed us toward more reliable ways of activity recognition.

1.6 Research methodology

The research project presented in this thesis was carried out by means of research methodology divided into four key steps.

During the first step of this research project, the problems of temporal data mining and activity recognition were investigated in depth. At this stage, we first aimed to gain general knowledge of the field of research, which is ambient intelligence and data mining related to the Smart Home context, by reviewing the key books [22, 25, 48] and works [21, 49, 50] on the subject. Then we conducted a much more targeted survey by reviewing the fundamental and applied works [51-53] related to the specific problem of temporal data mining in the Smart Home for activity recognition. Finally, we investigated why a temporal data-driven approach for activity recognition should essentially be proposed. This first step served as a basis for elaborating our conceptual contribution.

During the second phase of this research, we developed a new formal model inspired by the advantages and weak points discovered in the literature review. As pointed out in the review of previous works related to the Smart Home, temporal information constitutes a key element for correctly identifying activities of daily living, but only few works [23, 32, 45] made use of that kind of temporal data in the mining process. Moreover, most of them only relied on Allen's temporal framework [5], which is a very limited approach. Therefore, the second phase of our methodology consisted in developing a formal framework, adapted for our specific context, for temporal data mining in order to recognize activities. As a novelty, we also introduced fuzzy logic and, more specifically, *fuzzy time* in reasoning, which allows the taking into account of imprecision in activity patterns and recognition of activities. We modeled the activities, formalized our conceptual ideas and articulated them into a complete model. We then provided a framework for automatic reasoning with simultaneous activities.

The third step of our methodology consisted in validating the formal model developed during the previous step, in order to be able to test its efficacy. For this step, we used Microsoft

Visual Studio .Net, SQL server and MATLAB software to simulate the equations of the proposed model. The developed application includes a fuzzy-inference system, subtractive clustering algorithm, regression and some classification processes to interpret and make inferences about the observations, as well as a few software tools.

Finally, the fourth and final stage of the methodology consisted in experimenting our model inside the laboratory infrastructure of LIARA [2]. This infrastructure consists of a standard apartment (including a kitchen, living room, dining room, bedroom and bathroom) with a set of infrared sensors, pressure mats, video and audio actuators, smart tags (radio frequency identification or RFID) to obtain the position and real-time identification of individuals and main objects in the environment. More types of sensors relating to light and temperature are also embedded in this laboratory. We conducted experiments on our new temporal data mining model by using several case studies based on real-case scenarios (historical data from sensors). We then completed a statistical analysis of obtained results which we compared with the ones presented in former works. The results obtained were very promising.

Throughout all of the project's different steps, contributions to the field were made which took the forms of scientific publications in proceedings of reputed international conferences (Springer LNCS proceedings [7], AAAI symposium [9], AAAI expertise workshop of STAMI [8], MAICS [10], IEEE Workshop on Data Integration and Mining [11], etc.) and two long papers in a recognized journal (Springer Journal of Ambient Intelligence and Humanized Computing [12, 13]). These contributions were peer-reviewed and well received in the scientific community. These constitute a clear sign of recognition and confirm the potential of the contribution proposed in this thesis.

1.7 Organization of the thesis

This thesis is organized into five chapters that fall into direct chronology with the research methodology. The first chapter, now coming to an ending, was intended as an introduction to the thesis, our context of study and issues raised in this research. It provides a summary of the problems related to data mining for activity recognition in the Smart Home. It has also situated our contribution in the field, and has provided a comprehensive plan of what was achieved in the course of this research project.

The second chapter provides an introduction to ambient intelligence and the Smart Home, and introduces the reader to the main works in the field exploiting data mining, temporal data mining and activity recognition using learning techniques. The formal tools necessary to understand the related works are also presented. In other words, the chapter is a state-of-the-art look on the main data-mining approaches used in the Smart Home for activity recognition. At the end of the chapter, we conclude with a comparison between advantages and disadvantages of the different models presented and justify our proposition in regard to these previous works.

The third chapter examines the fundamental contribution resulting from this research. It describes in detail the formal elements of our temporal data-mining model and the incorporation of fuzzy temporal elements. The first contribution refers to the modeling of activities and proposes an innovative method used to perceive activities. As a second contribution, a fuzzy-logic based mining model is proposed to reason world state normality and events that may occur in future.

The fourth chapter presents the practical and experimental contribution of this research. It shows that validation of this new model of recognition within an intelligent home was achieved.

This chapter also discusses the process employed to test the proposed model with a series of case studies and presents a comparative analysis of results obtained from the main approaches in the field.

Finally, the fifth and last chapter concludes the thesis by presenting a detailed account of the research project highlighting the contribution of this work above and beyond previous works. This chapter will also address the limitations of the proposed model and future work which may arise from this research.

2 Literature review: data mining in the Smart Home

In this chapter, we aim to review the problem which motivated us to write this thesis; go over the preceding works related to the subject and associated artificial intelligence techniques which are concerned with activity recognition in the context of the Smart Home, particularly the ones which apply temporal data mining in the recognition process. They are analyzed; their proposed advantages, deficiencies and weak points are highlighted. Consequently, summary of their limitations in reviewed works is put forward in order to justify the proposal of the presented thesis. At the end of this section, we will briefly describe a schema about the proposed solution of this thesis.

2.1 Introduction to research in ambient intelligence and Smart Homes

A major development in recent years is the importance given to research on ambient intelligence (AmI) in the context of activity recognition. AmI, in opposition to traditional computing where the desktop computer is the archetype, consists of a new approach based on the capacities of mobility and integration of digital systems in the physical environment, in accordance with ubiquitous computing [51]. In this new vision, processors, sensors and actuators are integrated to the objects that are applied every day in order to carry out the activities of daily living (ADLs). This issue allows the ADL objects to be embedded in the environment as unobtrusive, interconnected, adaptable, dynamic and intelligent. This mobility and fusion are made possible by the miniaturization and reduced power consumption of electronic components and the omnipresence of wireless networks. It allows foreseeing the opportune composition of devices and services of all kinds on an infrastructure characterized by a granular and variable geometry, endowed with faculties of capture, action, treatment, communication and interaction [22]. Therefore, AmI is the vision of a future in which environments support the people inhabiting them. The envisioned AmI environment is sensitive to the needs of its inhabitants, and capable of anticipating their needs and behavior [19]. It is aware of their personal requirements and preferences, and interacts with people in a user-friendly way. A majority of works in literature [54] on the subject identify AmI as an intelligent, embedded, digital environment that is sensitive and responsive to the presence of people. For most authors, the key features here are intelligence and embedding [55]. By the term embedding, we mean miniaturized devices that merge into the environment while an inhabitant performs his or her ADLs. By intelligence, we mean the system is sensitive to context, is adaptive, learns from the behavior of users and gives punctual required services at the right moment. This important feature constitutes the central focus of this thesis as we address the issue of intelligence, learning and activity recognition. Similarly, research in [56] identifies five related key technology features: embedded, context aware, personalized, adaptive and anticipatory. Once again, we note that most of the identified key features (personalized, adaptive and anticipatory) are related to the issues that we address in this thesis, which are learning and recognition.

The AmI concept comes with great expectations. Some of them are near to realization, and others look more like science fiction for now. But the scope of potential AmI applications is rich and vast. For instance, families may now never lose track of their Alzheimer's disease-ridden relatives because of location sensors and miniature communication devices sewn into the fabric of clothes [55]. Blind people may be guided in unfamiliar environments by intelligent signposts and public transport timetables that may communicate via wireless headsets [57]. Traditional memory aids can remind the user about activities on their daily schedule, but more sophisticated memory aids can be context-sensitive. These can "observe" the user in his or her activities, infer desired tasks and, on that basis, issue reminders and guidance [21]. Today, we could even monitor walking patterns of cognitively-impaired people with smart shoes [58].

2.1.1 Application of AmI in the indoor context: Smart Homes

The potential of AmI at home has been the subject of research for at least a decade in both academic laboratories and major industries. For instance, the Microsoft Corporation Domestic Home [55] shows a case of AmI-based smart appliance technologies for residences. There are no desktop or laptop computers, but the wallpaper is interactive, and can be controlled by tablet PCs. The mailbox outside tracks the mailman's location by using a GPS and users can get a real-time estimate of when mail will arrive, on the mailbox display or by cell phone. RFID (radio frequency identification) tags embedded into envelopes even provide information about what mail is on the way. RFID tags are active barcodes which attach digital information to objects. This technology is being increasingly used by industry for tracking inventory. These tags can be quite small and do not require a battery. In the Microsoft Home, there are RFID tags on clothes as well. In a bedroom, by holding clothes up to a mirror which doubles as a screen, one can get information about these, including whether matching items like a skirt or jacket are in the wardrobe or laundry. The kitchen has an intelligent note board. If you pin a pizza coupon on it, the restaurant's menu and phone number are displayed. You can call the restaurant with a tap on the board.

In the UK, British Telecom (BT) and Liverpool City Council have run trails on telecare technology within a project devised by BT's Pervasive ICT Research Centre [59]. The trial concerned a system that responded to crises in the home. Each home contained 12 wireless sensors, including passive infrared sensors, entry switch sensors, occupancy sensors, a toilet flush sensor and a temperature sensor, all connected to a central gateway and a BT server via a secure broadband IP channel. If a cause for concern is flagged, a voice call is made to the home's occupant. If he or she confirms that they are okay, voice recognition technology is used to cancel the concern, otherwise a voice alert is sent to selected personnel who can then use a standard Web browser to access information about the inhabitant and circumstances of the alert.

Another example refers to the work of [60], where the bathroom mirror can remind a person of the medication he or she has to take, and the car stereo can tune into the same station that was on during breakfast. Some ideas and functionalities are distant visions, such as self-monitoring and self-painting walls, and lighting and furniture recognizing emotions and moods; some have already produced prototypes such as dormitories that learn the simple preferences of their single occupants regarding open or shut windows and level of lighting or heating. Some simple AmI-based devices such as temperature sensitive heating systems, movement-sensitive lighting and light-sensitive blinds are even routinely available in stores.

2.2 Smart Homes

Various names have been used to describe homes equipped with pervasive technology to provide AmI services to the residents who inhabit these. Smart Homes may be the most popular term [48], and other terms include aware houses, intelligent homes, integrated environments or alive, interactive, responsive homes/environments. Innovation in domestic technology has long been driven and marketed by the desire to reduce labor and improve the quality of time spent at home. This continues to be one of the motivations for development of AmI at home. Other factors include technological advances and expectations, and an increasing trend in a way of life that blurs the boundaries between home, work and places of rest and entertainment. In this thesis, Smart Homes are foreseen as a potential way of assisting cognitively-impaired people in performing their activities of the daily living [12] at home, thus increasing their quality of life and optimizing spent energy, which are the desired goals of Smart Home design. Technically, such a Smart Home can be described as an ambient home-similar environment which, through its embedded sensors, captures accomplishment of actions concerning to the realization of ADLs. The data resulting from observation of activity realization in the Smart Home forms what we consider a big data warehouse. It must be analyzed by artificial intelligence techniques embedded within the Smart Home in order to provide information about home state normality and then infer required assistance for the Smart Home resident or achieve other Smart Home design objectives. Thus activity recognition applies artificial intelligence techniques in order to provide key knowledge for the Smart Home by interpretation of Smart Home observations, in order to make decisions for assistance of the Smart Home resident or to achieve any objective of Smart Home design.

2.3 Knowledge provision for reasoning

Knowledge is the set of rules that indicate relationships between data points. Knowledge is a basic and essential requirement in reasoning. Generally, the required knowledge for a typical reasoning system can be sourced in two main ways. The first way is to transfer the human expert idea and apply his idea as knowledge; but considering that there may be a lot of machine states or narrow and delicate rules in activity recognition, it would be a very difficult task for an expert to define the rules for recognition of activities. Moreover, the expert may not be available every moment that we need him or her. Sometimes, he or she may not be fast enough at this task for every Smart Home in the world. There would be many exceptions for this imaginary rule-base and the expert would have to consider many parameters and variables. For example, assuming that many activity features would be observed in a typical Smart Home (600 activity features are observed in LIARA), and most sensors indicate multiple states of their observations (rather than binary-state sensors), thus activity recognition in LIARA would deal with *at least* 2^{600}=4.149516e+180 machine states. The reality is that this problem could cause billions of times more possible states, rather than the latter mentioned digit. Even the activity recognition problem is demonstrated as a sort of NP-Hard problem. It could become even more complicated if we desire personalized knowledge for each resident. The expert's high-level rules should be translated into low-level rules or instructions in order to conduct reasoning and world actuation through machine facilities (in this case, the Smart Home). Furthermore, at the machine level, there are some delicate specific low-level rules (instructions) that an expert could not normally recognize or even express with his high-level knowledge. Although there are other limitations to an expert-driven approach, the mentioned argument is sufficient to not select a *pure* knowledge-driven approach as a practical solution for activity recognition in the Smart Home. The alternative solution, which solves knowledge-driven problems, is intelligent data understanding by inferring the hidden knowledge from this data.

Hence, by this second generic way, the data resulting from frequent observation of activity realization is mined. This operation is called data mining [26] and it refers to a group of artificial intelligence techniques that deal with knowledge discovery taken from data. Generally, discovered knowledge is kept in a knowledge base or rule base. The challenge of a pure data-driven approach is that even if the best data-understanding method is applied, many high-level rules or concepts (easily understandable for an expert) may not be discovered. The solution proposed by this thesis is not to completely eliminate the role of the expert's knowledge from the knowledge provision process; however, this knowledge can be merged to the perceived rules from mined data through the fuzzy-logic theory [6, 61]. This theory provides facilities in order to correlate humanistic lingual knowledge to low-level instructions. Benefiting from this property on one hand, the expert could place the discovered knowledge resulting from data mining in a *fuzzy* rule base (which holds fuzzy rules) and, on the other hand, the expert can modify, remove or add the data-driven inferred fuzzy rules. Therefore, a fuzzy rule plays the role of a sort of mediator between complicated high-level concepts (derived from expert's idea) and the lower-level concepts that are understandable from data analysis.

2.4 Data mining

As it was mentioned earlier, knowledge is the set of rules that indicate relationships between data points. In order to automatically discover knowledge, data is mined. Generally, four operations may be conducted to discover relationships among data points in large databases. These are classification or prediction, clustering or segmentation, association rules discovery and frequent sequential pattern discovery [25]. The resulting knowledge, which is the set of rules, patterns or relationships, is stored in a knowledge base:

- **Classification** refers to an algorithmic procedure for assigning a given piece of input data into one of a given number of categories. In the classification process, stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order.
- **Clustering** means the assignment of a set of observations into subsets (called *clusters*) so that observations in the same cluster are similar in some sense. In the clustering process,

data items are grouped according to logical relationships or preferences. For example, data can be mined to identify market segments or consumer affinities.

- **Association rule learning** concerns the methods applied for discovering relationships between variables in large databases. From a technical viewpoint, association rule mining searches for relationships between items in a data set. For example, in a sports store, T-shirts and trousers are mostly bought together (or in the same quantity). The sequence of transactions is not notified in this example. Another example is that 98% of people who purchase tires and auto accessories also buy automotive services. One other specification of association rule mining is that there are no restrictions on the number of items in the head or body of the rule.
- **Sequential pattern discovery** refers to the discovery of the frequent patterns that exist in a sequence of events. In this process, data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes. In this process, the sequence of transactions is annotated.

Basic data-mining operations (classification and clustering) can be conducted based on various parameters or logics; for example, regardless of existing data values in a data set, values of data points can be compared to an absolute value like a definitive integer and any number greater than it may be classified in a specific category, as well as classification of the smaller numbers. Another way is to find the average of the data values and to classify these based on the mean value. In conclusion, there are various ways to cluster or classify data points and so there are various data-mining algorithms which are pertinent to find the rules that explain data sets succinctly; however, the efficiency and functionality of each algorithm may differ.

Choosing the best algorithm to use in a specific analytical task can be a challenge. While we may use different algorithms to perform the same task, each algorithm produces a different result, and some algorithms can produce more than one type of result. For example, we can use the decision tree [52] algorithm not only for prediction, but also as a way to reduce the number of columns in a data set because the decision tree can identify columns that do not affect the final mining model [25]. Algorithm complexity (time and memory), data-mining purpose, error rate, ability to consider data noise and uncertainty are some of the criteria that a data miner can consider in accomplishing his or her task [62]. Depending on the data type, various data-mining techniques may be applied in order to discover underlying knowledge. For example, if the data type is "text", then text data-mining (text mining) techniques can be applied in order to find relationships between texts [63]. Another example is "temporal data mining", which refers to knowledge discovery from temporal data sets [25]. We dealt with temporal data in activity recognition so that we could limit our literature review to the group of temporal data-mining methods. Before we review the methods pertaining to knowledge discovery from temporal data, we will survey how a typical data-mining algorithm enables one to discover knowledge. Therefore, in continuing this chapter, we will review some of the most important temporal-data comprehension methods and discuss the theoretical foundation and philosophy of each one.

2.4.1 Philosophy behind data-mining algorithms

The summarization of primary data into higher-level expressions is the principal task of data mining. A simple example of data mining is the equation that describes a line such as $y = x$. By a mathematical expression, all the points that exist by that equation are indicated *in a summarized manner* (see Figure 2.1).

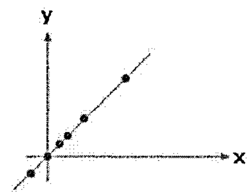


Figure2.1.Diagram of a $y = x$ equation

This equation can be discovered by a simple regression technique [34, 64]. Knowledge resulting from the data-mining process may include *exceptions (errors)*; for example, some data points (objects or individuals) may not be described by the discovered expression or the rules may validate objects that do not really exist. It is not impossible for a random-based function to explain reality with an error rate that is smaller than many algorithms. Therefore, the "error" may exist always with any data-mining method; however, its rate does matter. Algorithm complexity traditionally is another way to compare the functionality of data-mining algorithms.

A typical rule discovered by a typical data-mining algorithm is a formality that succinctly indicates a set of data points with their existing interrelationships. Rules can be expressed from different viewpoints with different logics. For example, when using set theory, the $y = x$ equation is formalized as $\{(x, y) \in \mathcal{R}^2 : y = x\}$. Out of the possible ways of rule representation, first order logic (FOL) is a way to indicate the discovered rules in consequence of causes and effects. In this way, rules like $a \rightarrow b \rightarrow c$ express a consequence of causes while demonstrating the priority of asserts and prepositions. For example in this case, a causes b and, by observation of c, the validation of a and b can be inferred; however, a has a higher priority than b and c [65]. Therefore, one of the more important tasks of data mining is to discover causality in the observations' data.

In order to discover rules and causality in data sets, typical data-mining algorithms' work is based on one of two possible general strategies, which are quantitative and qualitative mining.

2.4.1.1 Quantitative strategy mining

Quantitative strategy mining can be summarized by the following law: if there is a rule, then we should observe a quantity of individuals, samples, objects, or concerning data that are caused by it; if not, it does not exist. The rule that produces more individuals is of higher priority. For example, if there is a car factory that produces cars, so we should observe its production on the street. If these cars are observed on the street, then there is a factory that produced them (rule). Through this viewpoint, we are able to count similar objects and, according to the quantity of objects, representative rules are indicated and ranked. Then, based on calculated statistics, probabilistic rules – which estimate the future states – are calculated. In this way, these probabilistic rules are called discovered knowledge.

Let us indicate aforementioned approaches that depend on the count of objects as being instances of quantitative or probabilistic strategy. This strategy has some limitations that are indicated well in [66-68]. One important limitation that probabilistic data-mining approaches typically face is that observations must be stochastic, accidental or random. Furthermore, these observations must not be interfered intelligently. This is a contrast to Smart Home observations, which are directed intelligently by the Smart Home resident. The second limitation is that objects must be completely similar to be classified in a common category. The third limitation is that the quantity of observations must be sufficient and the observation set should include all possible events in order to obtain enough quantity. The fourth constraint is that object properties should be stable and objects cannot be of a dynamic nature or the observer should be non-dynamic (steady, fixed or stable). Another consequence of the latter limitation is that all objects should be measured and parameterized with common parameters and even common measuring units.

By probabilistic strategy, objects are not compared together; however, they are compared to unique and universal criteria (absolute values). Uncertainty is an indispensable feature of probabilistic approaches, since this classification process would naturally categorize two or more dissimilar objects in a common class just because they appear similar regarding absolute, universal criteria. This kind of classification (clustering) would make the inferred knowledge completely dependent on the expert-selected "absolute measures" and a partial modification (change) in the mentioned values would cause invalidity of knowledge. For instance, in activity recognition, we put some sensors which estimate the positions of objects in the environment. These sensors measure the distance to these objects; therefore, the quantitative inferred knowledge extracted from the analysis of this data would be highly dependent on the position (absolute or universal criterion) of the sensors in the environment and, if they were relocated, the previously generated knowledge would be no longer valid.

By a quantitative approach, if we have more information about an object (or many objects) than other objects, we should ignore this knowledge. In other words, we must consider only the common attributes of all objects in order to discover knowledge. Thus, "Knowledge ignorance" is another property of quantitative (probabilistic) machine-learning strategies. The most essential presumption of quantitative machine-learning approaches is that there is no intelligence behind the observed events and any event is caused accidentally.

There is another major disadvantage with quantitative strategies. Perhaps in reality there is a rule, but we observe no individuals concerning it, or the quantity of its individuals is much fewer than other individuals; therefore, as a result, this rule would be ignored or not be taken into account. For instance, if Factory *A* produces 1000 cars, while another one (Factory *B*) produces only five cars, then Factory *B* would be probably ignored, although it does indeed exists.

2.4.1.2 Qualitative strategy mining

There is a second machine learning strategy which includes *qualitative* methods of data understanding. These approaches work based on discovered similarities between objects. In these approaches, the possible number of discoverable relationships (rules) may be even greater than the number of objects. Qualitative approaches essentially presume that there is a plan behind events. By such strategies, possible plans that cause events in the world are modeled, and if observations match a part of a previously learned model, then it can be inferred that the plan is repeated. The law of this strategy says *"if there is a rule, then there is a perfect instance of such a rule and other objects of this rule are similar to this perfect object. The object that is the most similar to all of the other objects is selected as the best representative and indicates the generic rule's specifications. Other objects that represent specifications of some other groups of individuals are labeled as instances of more particular rules."*

Qualitative approaches welcome consideration of more objects' attributes in order to better distinguish objects' differences. This is in contrast to the quantitative machine-learning approaches that try to ignore the differences between objects. In this way, different machine states would be better distinguished and the rules which describe delicate relationships would be discovered.

Because a comparison is made between objects, inferred knowledge would be indicated based on the interactive states of the objects (relative knowledge) and it could be free from the measuring universal units. For example, in activity recognition, no matter where location-estimator sensors are placed in the environment, the distance and position of objects regarding themselves would be taken into account. This would provide some independence for the reasoning approach from steady-positioned hardware and other absolute parameters [7, 12].

Most of the proposed data-driven works in activity recognition up to now are probabilistic. Uncertainty in such approaches is the deviation from the learned plan and it would be handled if the plan structure were respected. In such approaches, no event occurs unless there is a reason behind it. In the next part of this thesis, we will discuss a series of machine-learning approaches that are applied in order to discover knowledge in Smart Homes.

As a brief conclusion, it can be said that basically, activity recognition in ambient environments is not a proper application of quantitative approaches because it may lead to high-process complexity, unreliability of knowledge and lack of confidence in reasoning and results.

2.4.2 Temporal data mining

Temporal data results when different states of a parameter (variable) are observed frequently, and per each definitive time point the value of the variable is indicated (see Figure 2.2).

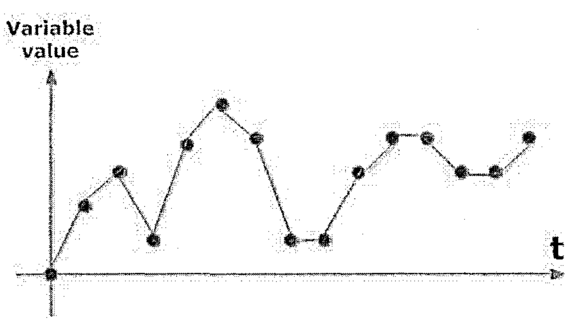


Figure 2.2. Temporal data sample

For example, sensors that frequently observe an activity feature in a Smart Home generate temporal data and this data is stored in temporal data sets (see Table 2.1). As it is shown in Figure 2.2, a temporal data set represents the statuses of variables at *common* time points. For instance, we can see the different statuses of different variables at time point t_2.

Table2.1 Temporal data set sample

Time	Var_1	Var_2	Var_3
t_1	Var_{11}	Var_{12}	Var_{13}
t_2	Var_{21}	Var_{22}	Var_{23}
t_3	Var_{31}	Var_{32}	Var_{33}
...	...	...	...

Temporal data sets indicate primary data concerning *behavior* of an observed system or agent. A total temporal data set tells how a plan (activity or operation) is realized in pure detail. These concrete details are highly low-level and need to be interpreted in order to be mined for modeling or understood by humans. Analysis of temporal data is a more difficult task than analysis of non-temporal data, such as an IRIS data set [69] in a botanic domain, because in non-temporal data each data record explains an object's specifications. However, it is the total temporal data set which indicates the behavior of a system. In other words, each temporal data set is one or more object that is intended to be learned.

An important task in temporal data mining is the computation of similarity between data points [25]. A prerequisite of performing such mentioned calculations is that one needs to know the temporal data type from which one would like to mine it.

2.4.2.1 Identification of temporal data type

Temporal data types can be of three different categories, which are time series, temporal sequences and semantic temporal data.

- Time series represent ordered real-value measurements at regular temporal intervals. A time series $X = \{x_1, x_2, \ldots, x_n\}$ for $t = t_1, t_2, \ldots, t_n$ is a discrete function with value x_1 for time point t_1, value x_2 for time point t_2, and so on. Time series data consists of varying frequencies in which the presence of noise is also a common phenomenon. A time series can be divided into multivariate and univariate[1] groups or into stationary and non-stationary groups[2];
- Temporal sequences are time-stamped at regular or irregular time intervals[3]. An example of a temporal sequence is the time-stamped sequence of purchases of a customer on a website. Another example refers to activity recognition if the sequence of actions of the Smart Home resident is time-stamped;
- Semantic temporal data refers to definition of temporal data within the context of ontology. For example, "senior" and "middle-aged" are defined within the context of human-lifetime ontology.

Higher level temporal data types can be defined based on the mentioned basic temporal data types. For example, an *event* can be considered as a special case of temporal sequence with one

---

[1] A multivariate time series is created by more than one variable, while in a univariate time series there is one underlying variable.

[2] A stationary time series has a mean and variance that do not change over time, while a non-stationary one has no salient mean and can decrease or increase over time.

[3] Time intervals represent the beginning and ending times of temporal entities such as activities.

time-stamped element. Similarly, a series of events is another way to denote that a temporal sequence is of the same types semantically, such as actions, operations, earthquakes, alarms, etc. After defining the temporal data type, it is time to clean it.

2.4.2.2 Temporal data cleaning

In order to get meaningful results from the data-mining process, the data must be appropriately preprocessed to become *clean*. Data cleaning aspects are the *missing data handling* and *noise removal*, which can be performed depending on the application.

Data can be missed for several reasons such as malfunctioning equipment, human error or environmental conditions. In order to handle the missing data there are two ways to proceed. The first way is to ignore the missing data and eliminate it from the data-mining process; however, the second way consists in estimating the missing data using interpolation techniques or other mathematical solutions.

Noise is defined as a random error that occurs in the data-mining process. By the expression random error, we refer to unplanned or unintended error. Such errors can be caused by several factors such as faulty measuring equipment and environmental factors. Noise can be removed in two general ways, the first of which is data binning. Through the binning process, data is divided into buckets or bins of equal size and then it is smoothed by using the mean, median or boundaries of the bin. In other words, binning methods smooth [70] a sorted data value by consulting its neighborhood. The sorted values are distributed into a number of buckets or bins. Because binning methods consult the neighborhood of values, they perform local smoothing. For instance, assuming that we had the following time series data: {4, 8, 15, 21, 21, 24, 25, 28, 34}, we would partition it into (equal-depth) bins, for example $Bin_1 = \{4, 8, 15\}$; $Bin_2 = \{21, 21,$

24}; $Bin_3 = \{25, 28, 34\}$. Using median smoothing, which is the middle value in the bin, we would get the following: $Bin_1 = \{9, 9, 9\}$; $Bin_2 = \{22, 22, 22\}$; $Bin_3 = \{29, 29, 29\}$; smoothing by bin boundaries would result in: $Bin_1 = \{4, 4, 15\}$; $Bin_2 = \{21, 21, 24\}$; $Bin_3 = \{25, 25, 34\}$.

The second way to remove noise is called moving-average smoothing. A moving average is a form of average which has been adjusted to allow for seasonal or cyclical components in a time series. Moving-average smoothing is a smoothing technique used to make the long term trends of a time series clearer. This technique relies on the principle that averages of data can be used to represent the original data. When applied to a time series, a number of data points are averaged, then another group of data points are averaged in a systematic fashion, and so on and so forth. It is generally quite simple. Table 2.2 represents a moving-average calculation:

Table2.2. Calculation of moving average in a time series

Time (t)	Data (y)	Moving average
1	12	
2	10	(12+10+15)/3=12.3
3	15	(10+15+13)/3=12.7
4	13	(15+13+16)/3=14.7
5	16	(13+16+13)/3=14.0
6	13	(16+13+18)/3=15.7
7	18	(13+18+21)/3=17.3
8	21	(18+21+19)/3=19.3
9	19	

The third column of Table 2.2 is computed from the first two in this way:

1- Take the first three t points (1, 2, 3) and find their average (2); take the first three y points in the table (12, 10, 15) and find their average (12.3).

2- Take the next three t points (2, 3, 4) and find their average (3); take the next three y points in the table (10, 15, 13) and find their average (12.7).

3- Repeat until you reach the last three t points.

4- Take the last three t points (7, 8, 9) and find their average (8); take the last three y points in the table (18, 21, 19) and find their average (19.3).

After data cleaning, it is time to survey the measuring of similarity degrees between data points.

2.4.2.3 Time-series similarity measures

If we have a time series that includes N data points, then the similarity measures of this time series can be surveyed in two ways. The first way is to define an absolute value like an epsilon (ϵ) and find all data-point pairs that have a distance that is less than ϵ. The second way is index by the content, which is done either by finding all-time series that have distance which is less than ϵ, or by finding the m closest neighbors for a specific time series. Out of several existing methods of similarity-degree estimations, we will deal with Euclidean distance and KL divergence in order to discuss the concept of similarity estimation adequately.

2.4.2.3.1 Euclidean distance

Euclidean metric is the ordinary distance between two points that one would measure with a ruler. The Euclidean distance between points p and q is the length of the line segment $\overline{pq}$ connecting them. In Cartesian coordinates, if $p = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$ are two points in Euclidean n-space, then the distance from p to q, or from q to p is given by $d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots . + (q_n - p_n)^2}$. The utility of the Euclidean distance metric is limited to time series that have the same baseline, scale and length. Furthermore, the two time series should not have any gaps. In addition, the Euclidean distance is sensitive to the presence of noise and variations in the time axis.

2.4.2.3.2 KL divergence

In both probability and information theory, Kullback–Leibler divergence [71], which is also called information divergence or KL divergence, is a non-symmetric measure of the difference between two probability distributions P and Q. If x is a discrete variable, then the KL divergence on two time series of P and Q would be $KLD(P,Q) = \sum_x P(x)\log(P(x)/Q(x))$, and, if x is a continuous variable, then the KL divergence on two time series of P and Q would be; $KLD(P,Q) = \int \log(P(x)/Q(x))\, d_{p(x)}$. The KL divergence method is different from simple Euclidean-distance measurement. The solutions in which Euclidean distance measurement is applied and comparison between pairs of data points matters; however, KL divergence measures the distance between two distributions of P and Q or, in other words, two time series. $KLD(P,Q)$ measures the similarity ratio of a P sample to a P time series and a Q sample to the Q time series. Furthermore, it identifies which random sample is typically similar to which time series.

2.4.3 Temporal classification and clustering

Classification is the task of assigning a new sample to a set of previously known classes, while clustering is the task of grouping samples into clusters of similar samples. This is the reason that classification is known as supervised learning and clustering is known as unsupervised learning. Hence this definition, we infer that the clustering operation on temporal data is generally of a higher priority than the classification process. But, it is possible to apply an expert's knowledge in data-mining processes and thus, the clustering phase would be eliminated or completed after classification.

2.4.3.1 Temporal clustering

A *cluster* is a set of similar objects, where similarity is defined by some measure of distance. Any of the distance measures discussed in the previous chapter can be used for the computation of similarity in clustering. Clustering is a type of *unsupervised* machine-learning technique because clusters are not predetermined by human beings. Unsupervised algorithms seek out similarity between pieces of data in order to determine whether they can be characterized as forming a group called a cluster.

The clustering problem is a kind of challenge because attributes[4] and their values, which differentiate one cluster from another, are not known. We do not have data to tell us which features differentiate objects that belong to different clusters and, as the quantity of data increases, the number of clusters as well as the number and type of differentiating factors might change. Moreover, there is no guide to indicate what constitutes a cluster and the success of clustering algorithms is influenced by the presence of noise in the data, missing data and outliers.

Generally, clustering can be done in three strategies. The first way is to cluster data is based on predetermined or absolute values. The second way is to cluster data based on existing differences and similarities within data values and the third strategy is to cluster data points based on a combination of absolute and relative parameters. The first strategy of clustering, which deals with segmentation of data points based on some absolute values which are irrelevant

---

[4]These are also called features, aspects or properties.

and independent from data *values*, is called partitioning. The k-means algorithm is an instance of this strategy. The fuzzy c-means algorithm is an instance of the second strategy which considers relative values for clustering and, finally, subtractive clustering is an instance of the third type of clustering which considers both the relative and absolute criteria in clustering.

2.4.3.1.1 K-means clustering

The main idea in a k-means [72] clustering algorithm is to create k clusters of data, where the number k is entered by the *expert*. This algorithm is suitable mainly for numerical data. The original clustering, also known as *partitioning*, is performed randomly and then objects are moved in and out of clusters, using as a guide the criterion of "closeness". Partitioning algorithms are very popular because of their ease of implementation and low computational cost. However, they do have the following disadvantages: (1) they are sensitive to the presence of noise and outliers, (2) they can discover only clusters with convex shapes and (3) the number of clusters needs to be specified.

As it was mentioned earlier, the main idea in the k-means algorithm is to use the *means* to represent clusters and use these as a guide to assign objects to clusters. K-means is one of the simplest unsupervised learning algorithms. The procedure follows a way to classify a given data set through a certain number of clusters (assume k clusters) fixed beforehand. The main idea is to define k centers with a ratio of one for each cluster. These centers should be placed in an intelligent way because their placement in different locations will cause different results. As a result, the better choice would be to place them as far away as possible from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate new k centers resulting from re-performance of the previous step. After we have found these k new centers, a new binding has to be conducted between the same data set points and the nearest new centers. A loop has thus been generated. As a result of this loop, we may notice that k centers change location step by step until no more changes are required. In other words, these centers stop moving.

Finally, this algorithm aims to minimize an *objective function*, in this case a squared error function. The objective function $J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)} - c_j\right\|^2$, where $\left\|x_i^{(j)} - c_j\right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the center, which is an indicator of the distance of the n data points from their respective cluster centers. Algorithm 2.1 is composed of the following steps:

Input:
- learning data set, k is the number of clusters

Output:
- cluster center matrix

1: Place randomly k points into the space represented by the objects that are being clustered.
2: Assign each object to the group that has the closest centre.
3: When all objects have been assigned, recalculate the positions of the k centroids.
4: Repeat steps 2 and 3 until the centroids no longer move.

Algorithm 2.1.: k-means

Although it can be proven that the procedure will always end, the k-means algorithm does not necessarily identify the most optimal configuration corresponding to the global objective function minimum. The algorithm is also significantly sensitive to initial randomly selected cluster centers. Consider to this effect the following k-means example where we have six time-series data sets, which are represented using a 2-D vector consisting of the standard deviation and fractal dimension of the time-series data as seen here: $s_1(0.8,\ 0.1)$, $s_2(0.1,\ 0.6)$, $s_3(0.2,\ 0.7)$, $s_4(0.6, 0.2)$, $s_5(0.7, 0.3)$, $s_6(0.3, 0.9)$. As Figure 2.3 illustrates, there are two distinct clusters. A new data point at (0.3, 0.6) will be classified into cluster 1.

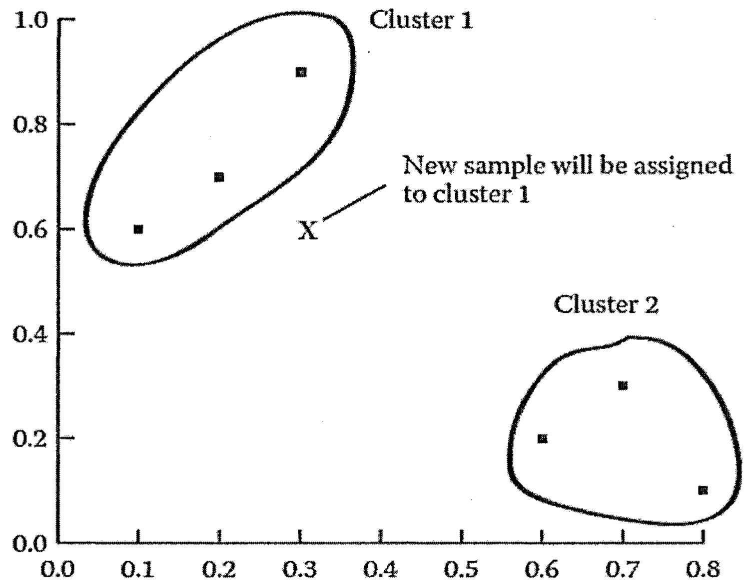


Figure2.3. k-means clustering example

2.4.3.1.2 C-means clustering

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method [73] is frequently used in pattern recognition. It is based on minimization of the following objective function: $J_m = \sum_{i=1}^{N}\sum_{j=1}^{C}(u_{ij})^m\left\|x_i - c_j\right\|^2$ where $m \in [1, \infty]$ is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the $i^{th}$of d-dimensional measured data, c_j the d-dimension center of the cluster, and $\|*\|$ is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by

$u_{ij} = 1/\sum_{k=1}^{C}\left(\|x_i - c_j\|/\|x_i - c_k\|\right)^{\frac{2}{m-1}}$ and $c_j = \sum_{i=1}^{N}(u_{ij})^m . x_i/\sum_{i=1}^{N}(u_{ij})^m$. This iteration will stop when $\underbrace{max}_{ij}\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right| \le \varepsilon$, where ε is a termination criterion between 0 and 1, whereas k is comprised of the iteration steps. This procedure converges to a local minimum or saddle point of J_m. The FCM steps are shown in algorithm 2.2:

Input:

- Learning data set

Output:

- Cluster center matrix

1: Initialize $U = [u_{ij}], U^{(0)}$
2: At k step: calculate the centers vectors $C^{(k)} = [c_j]$ with $U^{(k)}$, $c_j = \sum_{i=1}^{N}(u_{ij})^m . x_i/\sum_{i=1}^{N}(u_{ij})^m$
3: Update $U^{(k)}, U^{(k+1)}$, $u_{ij} = 1/\sum_{k=1}^{C}\left(\|x_i - c_j\|/\|x_i - c_k\|\right)^{\frac{2}{m-1}}$
4: If $\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right| \le \varepsilon$ then STOP; otherwise return to step 2.

Algorithm 2.2.: c-means

In the following example, we have time-series data sets which are presented in table 2.3 applying a fuzzy c-means algorithm, and the cluster centers of these temporal data sets are desired. By performing c-means clustering, the positions of the cluster centers are highlighted in Figure 2.4. The cluster centers of the temporal data represented in table 2.3 are at the (0.2654, 0.7230) and (0.6969, 0.3206) positions. In this way, any other new data points would be classified in one of the mentioned clusters based on the membership value that it could have to the clusters. In other words, the more similarity to one of the calculated cluster centers is found, the more membership value is assigned to the new point.

Table2.3.: Source of Temporal Data for the FCM example

time	parameter 1	parameter 2	time	parameter 1	parameter 2	time	parameter 1	parameter 2
1	0.21895919	0.71971108	50	0.24784176	0.91691429	99	0.75294041	0.33203456
2	0.67929641	0.31389837	51	0.4764318	0.57842135	100	0.66952064	0.50487466
3	0.83096535	0.47903993	52	0.38931417	0.52762005	101	0.63342991	0.42841739
4	0.053461635	0.77931485	53	0.20325033	0.71023354	102	0.22700772	0.84878311
5	0.68677271	0.52219568	54	0.94748678	0.33117965	103	0.69983444	0.26863618
6	0.091964891	0.71405761	55	0.13118853	0.75624968	104	0.52612328	0.47793654
7	0.65391896	0.16631787	56	0.88564837	0.28843555	105	0.32966639	0.75485767
8	0.70119059	0.422224	57	0.09217363	0.73633707	106	0.48532518	0.40283149
9	0.91032083	0.31883553	58	0.36533903	0.54023723	107	0.8602257	0.38888944
10	0.73608188	0.45760685	59	0.25305736	0.76713302	108	0.55683578	0.22796971
11	0.63263857	0.14380738	60	0.78315317	0.50865692	109	0.73899661	0.48685151
12	0.7226604	0.3206678	61	0.34952414	0.73388084	110	0.52854827	0.30300425
13	0.75335583	0.46373323	62	0.21524838	0.86596086	111	0.31073887	0.59246226
14	0.072685883	0.87749311	63	0.67959237	0.20414434	112	0.58811912	0.51315049
15	0.27270997	0.64868126	64	0.25012559	0.82884477	113	0.51808359	0.52021714
16	0.76649478	0.39236853	65	0.86085984	0.3939987	114	0.37022628	0.72335304
17	0.35926498	0.64307502	66	0.81756148	0.49034177	115	0.47589622	0.29663054
18	0.90465309	0.18695199	67	0.75584353	0.17416348	116	0.078263208	0.76459728
19	0.49397668	0.54814877	68	0.82469739	0.36823673	117	0.36974201	0.58650577
20	0.26614451	0.73637975	69	0.10343393	0.79543708	118	0.67178415	0.21699766
21	0.073749075	0.79043769	70	0.57671664	0.30030901	119	0.67623692	0.079747635
22	0.52974739	0.40064069	71	0.87656572	0.29353873	120	0.51393637	0.31850888
23	0.46444582	0.56809172	72	0.44003866	0.50536909	121	0.72860836	0.17882267
24	0.77020455	0.16379344	73	0.86926374	0.28842873	122	0.72076782	0.472587
25	0.62954342	0.23056262	74	0.88603112	0.13881256	123	0.32156009	0.74857788
26	0.73622451	0.065969527	75	0.46332273	0.5554808	124	0.46043417	0.34848546
27	0.88857221	0.23088594	76	0.71342232	0.47622826	125	0.6613555	0.064064059
28	0.5132737	0.31833422	77	0.66767907	0.24682623	126	0.6056397	0.14123407
29	0.59111358	0.24322982	78	0.68204912	0.4084118	127	0.6700984	0.49318005
30	0.53730398	0.38338957	79	0.31573241	0.68839481	128	0.52280771	0.29752862
31	0.46791737	0.62853425	80	0.46753178	0.53192527	129	0.26661332	0.70293582
32	0.28721237	0.7751988	81	0.31917759	0.88998057	130	0.24673315	0.66149041
33	0.1763277	0.7663085	82	0.68249423	0.48122673	131	0.81710133	0.51471396
34	0.80240573	0.25589577	83	0.83641988	0.14312388	132	0.16074891	0.88552542
35	0.49848012	0.33800649	84	0.70892061	0.48309189	133	0.7078263	0.1353905
36	0.55458385	0.51581954	85	0.82870795	0.32547527	134	0.43663845	0.50815926
37	0.89073748	0.37895436	86	0.2135468	0.65412626	135	0.75171009	0.19508763
38	0.62484929	0.085975014	87	0.38985359	0.6198854	136	0.90330118	0.3487054
39	0.71470997	0.503418	88	0.77686582	0.30887354	137	0.5847872	0.080508049
40	0.2399108	0.6709439	89	0.7638652	0.23765611	138	0.62686137	0.47778535
41	0.68134621	0.060274597	90	0.28215589	0.92267568	139	0.65905333	0.13837822
42	0.147533	0.87615123	91	0.81972609	0.16874402	140	0.53866129	0.13050593
43	0.56718662	0.47366216	92	0.60101011	0.27616354	141	0.43663845	0.50815926
44	0.59010861	0.48229278	93	0.82835472	0.50918299	142	0.75171009	0.19508763
45	0.55614614	0.15842396	94	0.15773118	0.83858058	143	0.90330118	0.3487054
46	0.40876669	0.67277244	95	0.23359919	0.72674889	144	0.5847872	0.080508049
47	0.56489868	0.42322414	96	0.63471744	0.33295071	145	0.62686137	0.47778535
48	0.48851455	0.24175423	97	0.79476981	0.45126372	146	0.65905333	0.13837822
49	0.65125374	0.54492761	98	0.69624281	0.38926233	147	0.53866129	0.13050593

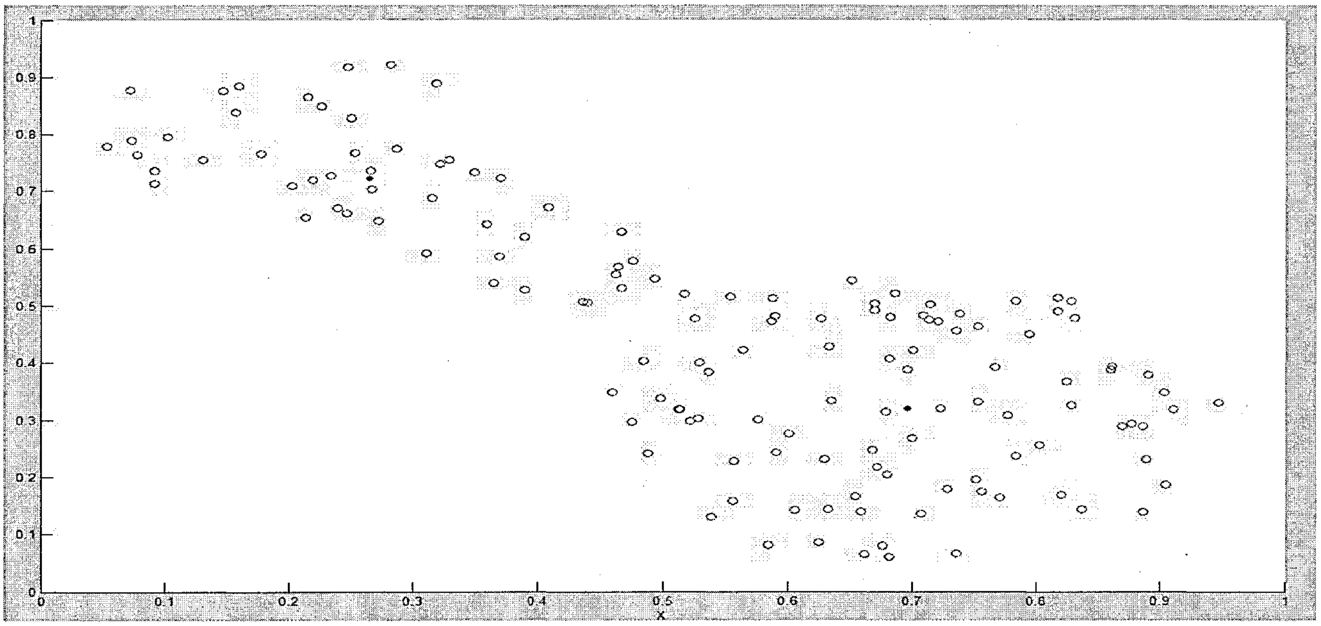


Figure2.4.: fuzzy c-means clustering example

In a c-means algorithm, the minimum absolute parameter (that is the quantity of desired clusters) is applied and the cluster centers are calculated based on the existing relationships and similarities in data points. This method of clustering is functional in cases where only the relationship between the surveyed is desired. In other words, the relationship between applied data and any other sources of data (like context data) is not desired. For example, in table 2.3, regardless of the objection of the problem or its functionality and despite the significance of the parameters, the cluster centers are calculated. However, it is possible that some absolute conditions in clustering are effective. For instance, it is not necessary here that parameter 2 be lower than 0.3 (regardless of the significance of the second parameter) in order to be allowed to be classified in the first cluster. While it is not intended that this data be compared to any other sources of data, so this clustering could be acceptable.

There are some cases in which we intend to extract rules that indicate both the absolute and relative relationships between data points and, in this way, not only can we explain the system's internal states (relative rules), but also the system's external rules (absolute rules). Subtractive clustering is an instance of such a data mining strategy, which will be discussed in the next section.

2.4.3.1.3 Subtractive clustering

Supposing that we do not even have a clear idea how many clusters there should be for a given set of data; *Subtractive clustering* is a fast, one- pass algorithm for estimating the number of clusters and the cluster centers in a set of data. To describe subtractive clustering technique, we start with the concept of *cluster centers* that mark the mean location of each cluster. Initially these cluster centers are very inaccurately placed. Additionally, every data point has a membership grade for each cluster. By iteratively updating the cluster centers and the membership grades for each data point, we can watch the cluster centers move to the right location. This iteration is based on minimizing an objective function that represents the distance from any given data point to a cluster center weighted by that data point's membership grade.

The objective function in the algorithm tries to find the best point has the best potential to be the cluster center. Considering a collection of n data points $\{x_1, x_2, \dots, x_n\}$ in an N dimensional space; without loss of generality, and assuming that the data points have been normalized in each dimension so that they are bounded by a unit hypercube, the potential of a point x_i in order to be the cluster center is calculated by $P_i = \sum_{j=1}^{n} e^{-\alpha\|x_i - x_j\|^2}$, $\alpha = 4/(r_a)^2$, where $\|.\|$ denotes the Euclidean distance, r_a is the cluster radius which is a positive constant. Thus, the measure of the potential for a data point is a function of its distances to all other data points. In other words, the possibility that a point becomes a cluster center depends on its distance to other points and a data point with many neighboring data points will have a high potential value. The constant r_a is effectively the radius defining a neighborhood; data points outside this radius have little influence on the potential. After the potential of every data point has been computed, we select the data point with the highest potential as the first cluster center.

Let x_1^* be the location of the first cluster center and P_1^* be its potential value. The potential of each data point x_i is revised by the formula $P_i \leftarrow P_i - P_1^* e^{-\beta\|x_i - x_1^*\|^2}$ and $\beta = 4/{r_b}^2$, where r_b is a positive constant and we subtract an amount of potential from each data point as a function of its distance from the first cluster center. The data points near the first cluster center will have greatly reduced potential, and therefore will be unlikely to be selected as the next cluster center. The constant r_b is effectively the radius defining the neighborhood which will have measurable reductions in potential. To avoid obtaining closely spaced cluster centers, r_b is set to be somewhat greater than r_a; a good choice is $r_b = 1.25\, r_a$. When the potentials of all data points have been revised, the data point with the highest remaining potential is selected as the second cluster center. The process is then continued further. In general, after the k^{th}cluster center has been obtained, the potential of each data point is revised by the formula $P_i \leftarrow P_i - P_k^* e^{-\beta\|x_i - x_k^*\|^2}$. As a conclusion on the subject of subtractive clustering, let us mention that four parameters generally affect the decision of cluster-center selection. They are named the cluster radius rate, squash factor, reject ratio and accept ratio.

- Cluster radius rate or influence range (IR): defines a neighborhood, the data points outside of this radius have little influence on potential.
- Squash factor: represents the penalty rate in data space. It defines the neighborhood, which will have measurable reductions in potential.

- Reject ratio: specifies a threshold for the potential above which the data point is definitively accepted as a cluster center;
- Accept ratio: specifies a threshold below which the data point is definitively rejected.

Cluster centers are found based on the four mentioned parameters. As it was stated earlier, the subtractive-clustering method assumes each data point is a potential cluster center and calculates the measure of likelihood that each data point would define the cluster center, based on the density of surrounding data points. In short, the algorithm does the following:

1. It selects the data point with the highest potential to be the first cluster center.

2. It removes all data points in the vicinity of the first cluster center in order to determine the next data cluster and its center location.

3. It iterates this process until all of the data is within the radius of a cluster center.

In a MATLAB [47] workbench-development environment, the cluster estimates obtained from the *subclust* function can be used to initialize iterative optimization-based clustering methods (like fuzzy c-means) and model identification methods (like ANFIS). The *subclust* function finds clusters by using the subtractive-clustering method. The *genfis2* function builds upon the *subclust* function to provide a fast, one-pass method to take input-output training data and generate a Sugeno-style fuzzy inference system [74] that models data behavior.

Considering the following example, we have time series data sets which are presented in table 2.3 applying a subtractive-clustering algorithm. The cluster centers of this temporal data set are desired. By performing subtractive clustering, the positions of the cluster centers are highlighted in Figure 2.5:

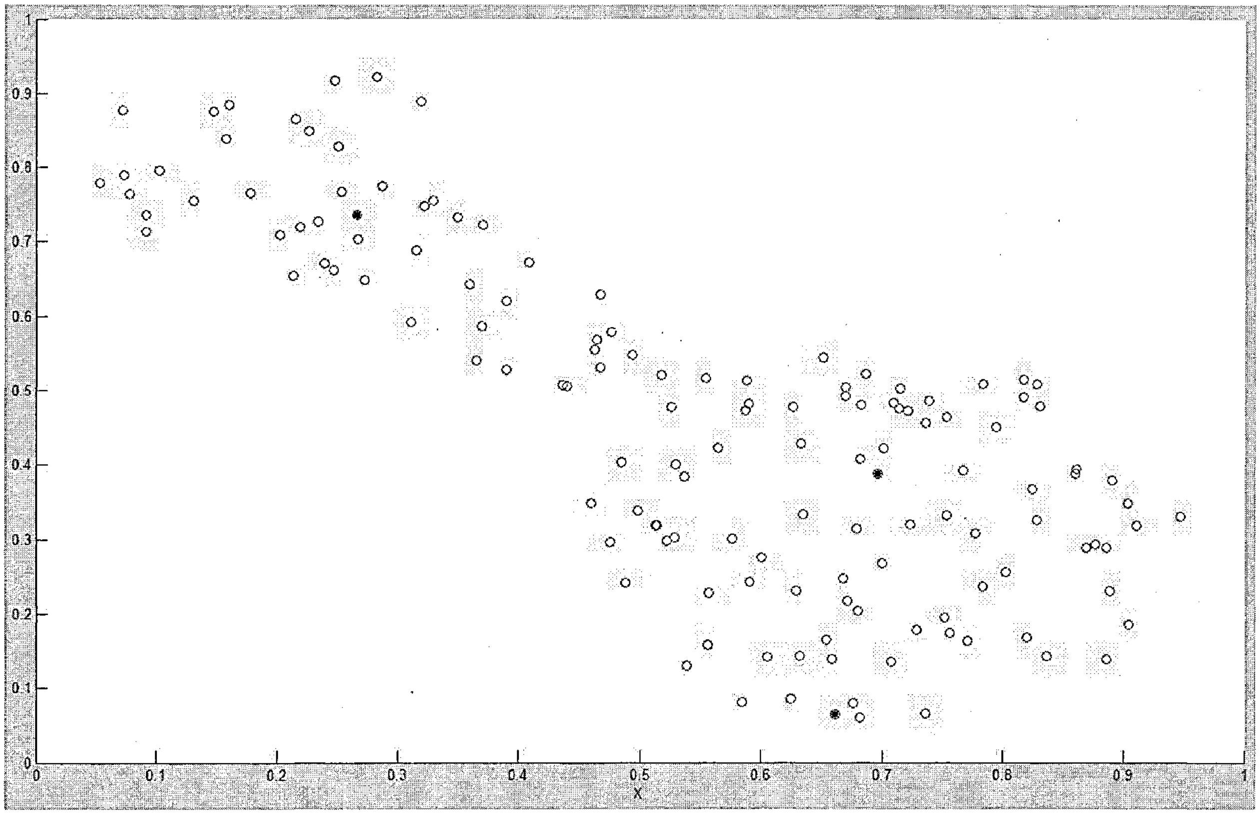


Figure2.5.: subtractive clustering example

As it is highlighted in Figure 2.5, cluster centers of the temporal data represented in table 2.3 are at the (0.6962, 0.3893), (0.2661, 0.7364) and (0.6614, 0.0641) positions. These points are calculated with an influence range of 0.5. Each discovered cluster center in the figure is mapped as a fuzzy rule in the fuzzy-rule base [75]. Based on the estimated fuzzy-membership values of a new data point's (object's) attributes regarding the discovered fuzzy clusters, the object would then be assigned to the appropriate clusters.

In subtractive clustering, each data point is compared to other data points in order to find the points that are closed regarding each other (relative rules). The second comparison is made based on their mapped values on the axes (absolute rules). Therefore, as is the case with the second and third discovered cluster centers in Figure 2.5, the cluster centers that are relatively close (or not too far from each other), but which justify different delicate rules, are distinguishable.

This method of clustering is functional for cases in which the relationships inside data points are desired, as well as the relationships of data points regarding external measurers (axes) or absolute parameters. In some cases like [75], it is mentioned that this method is an appropriate way to indicate rules between the inputs and outputs of systems.

2.4.3.1.4 Application of temporal clustering in activity recognition

Temporal clustering technique has been applied in some recent works pertaining to activity recognition. For example, in [32] the c-means clustering technique is applied. In [29], the k-means clustering technique is utilized. In the current section, we will discuss work done by [30] as a sample of the state of the art.

In [29], it is discussed that in order to model and recognize activities, temporal dependencies of these same activities should be taken into account. A temporal model for discovering temporal features and relations of activity patterns from sensor data is proposed. Temporal features of activities are information such as order of activities, their usual starting times and durations. Such information is discovered by k-means clustering and it can be used in order to recognize anomalies or for reminder systems. Discovering a confidence level for relations of order, as well as finding typical starting times and durations of mixture models allows one to assign probability to a particular activity occurring at a given interval. Using the assigned probabilities and anomaly detection methods, one can exploit such information to detect any anomalous or suspicious activities.

Discovered temporal information can be used to construct a schedule of activities for an upcoming period. Such a schedule is constructed based on the predicted starting time intervals, as well as relative order of activities. For example, it can be used as part of an activity reminder system for an Alzheimer's patient; in order to automatically generate prompts to remind the patient to initiate activities if s/he does not follow the usual schedule within expected starting times. The data derived from 50 days of activities is processed in this work. A sample of input data for the temporal data-mining process of this work is presented in table 2.4.

Table2.4. Sample of collected data in [29]. Here, M11 is a motion sensor and D34 is a door sensor.

Sensor ID	State	Timestamp
M11	ON	05/13/2009, 08:33
D34	Open	05/13/2009, 08:35

After that the k-means clustering algorithm is performed on the temporal data, clusters representing the beginning time of activities and their durations are calculated. In table 2.5, we present the calculated clusters relating to the beginning time of the medication-taking activity.

Table2.5. Beginning time of the medication-taking activity [29]

Cluster number	Mean (hh:mm)	Number of members	Standard deviation (hh:mm)
1	9:33	62	0:56
2	7:22	12	1:50
3	20:37	7	1:41

As is the case with the calculation of clusters representing the beginning time of activities, the activity durations are calculated. In consequence, the probabilities indicating which activity would be likely to begin after the current activity are calculated and a probabilistic prediction is provided (see table 2.6). The probability that the eating activity begins after the bathing activity is 0.14.

Table 2.6. Temporal relations for five ADLs [29]

	Activity	Cluster	Start Time (hh:mm) $[\mu_s - \sigma_s$ to $\mu_s + \sigma_s]$	Duration (hh:mm) $[\mu_d - \sigma_d$ to $\mu_d + \sigma_d]$	Next Activity	Conf
1	Bathing	1	[0:46 – 01:40]	[0:25 – 0:31]	Sleeping in Bed	0.42
					Bed-Toilet Transition	0.38
					Personal Hygiene	0.20
		2	[08:22 – 11:02]	[0:15 – 0:19]	Personal Hygiene	0.71
					Eating	0.14
		3	[21:55 – 23:03]	[0:04 – 0:10]	Sleeping Not in Bed	0.50
					Personal Hygiene	0.25
					Housekeeping	0.25
2	Bed-Toilet Transition	1	[20:00 – 23:46]	[0:14 – 0:16]	Bed Toilet Transition	0.33
					Sleeping Not in Bed	0.22
					Housekeeping	0.22
		2	[05:09 – 07:49]	[0:08 – 0:10]	Sleeping in Bed	0.95
		3	[00:54 – 03:10]	[0:02 – 0:04]	Sleeping in Bed	0.58
					Bed-Toilet Transition	0.36
3	Eating	1	[06:25 – 11:01]	[0:08 – 0:24]	Housekeeping	0.61
					Personal Hygiene	0.39
		2	[12:19 – 14:47]	[2:18 – 3:20]	Housekeeping	0.60
					Personal Hygiene	0.26
		3	[18:38 – 22:28]	[0:34 – 1:02]	Housekeeping	0.59
					Personal Hygiene	0.27
4	Enter Home	1	[18: 03 – 20:57]	N/A	Personal Hygiene	0.57
					Meal Preparation	0.29
		2	[09:35 – 14:07]	N/A	Personal Hygiene	0.46
					Meal Preparation	0.42
		3	[15:10 – 17:14]	N/A	Personal Hygiene	0.67
					Leave Home	0.14
5	Housekeeping	1	[12:55 – 16:17]	[1:29 – 1:33]	Personal Hygiene	0.59
					Leave Home	0.17
					Sleeping Not in Bed	0.17
		2	[19:40 – 23:08]	[0:01 – 0:03]	Sleeping Not in Bed	0.48
					Personal Hygiene	0.39
		3	[04:33 – 11:53]	[0:23 – 0:39]	Personal Hygiene	0.91

An advantage of this work is that the expert's role in calculating the probabilities of the sequence of probable activities is decreased, but a disadvantage is that it is still dependent on the expert's role to recognize the beginning and ending time of activities. Because this work applies a probabilistic method for knowledge discovery, it includes probabilistic strategy constraints. For example, a high quantity of training tests is needed to be effective.

2.4.3.2 Temporal classification

Classification is the task of assignment of new objects to previously known classes. Classes can be formed or defined in two ways. The first way is to calculate classes by application of the clustering process (a data-driven approach), and the second way refers to the application of the expert's knowledge (supervision or assistance) in class definition, such as the class definition by a programmer in object-oriented programming (OOP). By the knowledge-driven approach it is assumed that expert has some domain knowledge about the problem and so the classes are defined by his interruption. Therefore, the clustering process may be bypassed.

There is a supplementary task that a classifier may do; however, it is not taken into account as a main task of classification. This task refers to an operation, which may be done on the continuous or continuous-like variables such as integers in some classifiers like C4.5 [76]. Similarly, in support vector machines [77], for each data point vectors supporting intervals around it are calculated. By this operation, numeric variables are broken into some intervals and, instead of consideration of the data points in classification, intervals are substituted. This supplementary operation would help to decrease the number of machine states in classification.

Assignment of new observations into classes is the challenge of the classification process. Usually, there is more than one way for classifying new objects and so there is more than one way to represent learned knowledge. Therefore, there are several ways of reasoning in order to classify objects into categories. For example, the C4.5 and ID3 classifiers accomplish their task objecting to minimize information entropy, or the support vector machine (SVM) algorithm aims to find the best classification function to distinguish between members of two classes in training data.

Prediction is the goal of classification. By the classification process, the conditions in which a target can be achieved are extracted and then put into order so as to predict the future value of a target variable. This is useful because of the need to know which classes cause what results. There are several methods which are pertinent to use in demonstrating extracted knowledge from the classification process. One way is to describe the knowledge discovered from training data by the "if-then" propositions of first-order logic. In this way, the predicates indicate the target variable. The decision tree [52, 78] is another way of knowledge representation. In a decision tree, the leaves indicate the target variable.

2.4.3.2.1 Decision tree

A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision. Decision trees are commonly used for gaining information for the purpose of decision making. A decision tree starts with a root node on which users take action. From this node, users split each node recursively according to decision-tree learning algorithm. The final result is a decision tree in which each branch represents a possible scenario of decision and its outcome. The main modus operandi of decision trees is using inductive methods to the given values of attributes of an unknown object to determine appropriate classification according to decision-tree rules. For example, in table 2.7, attributes of the weather conditions necessary in order to make a decision to play a game outside, including the decision taken (in the last column), are indicated.

Table 2.7. Primary data for classification in a decision tree

ID Code	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cold	Normal	False	Yes
6	Rainy	Cold	Normal	True	No
7	Overcast	Cold	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cold	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Hot	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Now, a decision tree which indicates the primary data of table 2.7 in a summarized and classified manner is presented in Figure 2.6.

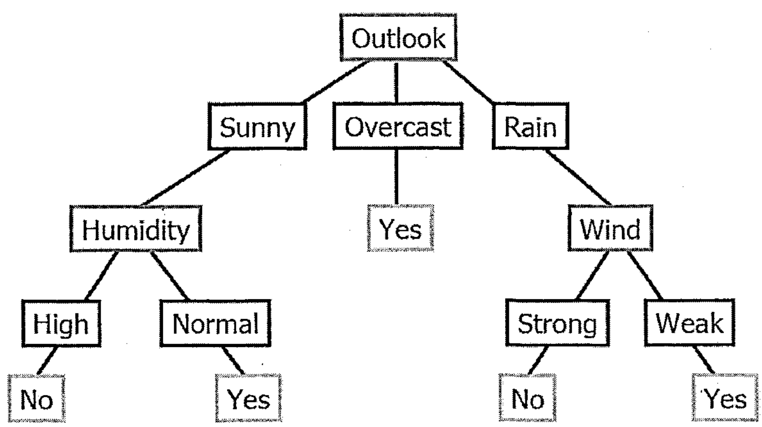


Figure2.6. Decision tree representing the inferred knowledge from table 2.7

In brief, decision-tree learning is a method for approximating discrete-valued target functions in which the learned function is represented by a decision tree. Decision trees classify instances by traversing from root node to leaf node. We start from the root node of a decision tree, testing the attribute specified by this node, and then moving down the tree branch according to the attribute value in the given set. This process is then repeated at the sub-tree level. Decision trees represent rules, which can be understood by humans and used in a knowledge system such as a database.

Generally, decision-tree learning algorithm suites for cases that input instances are represented as attribute-value pairs[5]; the target function has discrete output values, and the training data may contain errors[6].

It is generally considered that decision trees provide an effective method of decision making because they firstly provide a framework to quantify the values of outcomes and probabilities of achieving them. Secondly, they clearly describe the problem so that all possible outcomes can be investigated. Thirdly, they allow humans to analyze the possible consequences of a decision [52]. Decision trees are considered helpful when trying to make the best decisions on the basis of existing information and best guesses. Decision trees could be also helpful to create an environment which is capable of reacting 'intelligently' by anticipating, predicting and making decisions with signs of autonomy.

Visual presentation makes the decision-tree model easy to understand. Thus, it is suitable for human inspection as well as for making automated decisions. Consequently, the decision-tree approach is frequently used in the data-mining domain. Decision trees may also be used for

---

[5]For example, attribute of 'temperature' and its value 'hot', 'mild', 'cool'.
[6]This can be dealt through use of pruning techniques that we will not cover here.

regression (predicting a specific real value). The outcome of a decision-tree algorithm is a decision-tree model, or shortly decision tree.

As it is presented, a simple decision-tree learning algorithm completes the classification task in a non-optimized manner and its main goal is classification, but the ID3 and C4.5 algorithms apply the concepts behind decision trees in order to classify training test samples and represent knowledge; each one optimizes the final result based on a parameter. The final discovered knowledge (by each method) is represented using a decision tree. The training process that creates a decision tree is called induction. Induction requires a certain number of passes through the training data set. Generally, the algorithms which pass lesser through the training data set are more efficient than the ones that pass more through the training data set [52, 79]. In the following section, we will discuss the ID3 algorithm.

2.4.3.2.1.1 ID3 algorithm

In decision-tree learning, the ID3 algorithm [76] is used for generating a decision tree. ID3 is the precursor to the C4.5 algorithm and it uses information theory introduced by Shannon in 1948 [80] and builds the decision tree from the top down without backtracking the tree. In ID3, an information-gain criterion is used to select the most useful attribute for classification. The information gain of a given attribute X with respect to the class attribute Y is the reduction in uncertainty about the value of Y when we know the value of X. Evidently, in order to calculate the information gain, we need to know the information entropy. Information entropy is calculated according to the following formula: if $E(S)$ is the information entropy of the set S and n is the number of different values of the attribute in S, and $f_s(j)$ is the frequency of the value j in the set S, then $E(S) = \sum_j f_s(j) log_2(f_s(j))$.

The "best attribute"[7] is selected based on the information gain factor given by the following formula: $G(S,A) = E(S) - \sum_j^m f_s(A_i)E(S_{A_i})$ is the gain of the set S after a split over the A attribute, m refers to the number of different values of the attribute A in S, $f_s(A_i)$ is the frequency of the items processing A_i as i^{th} value of A in S and $S_{A_i}$is a subset of S.

There are three requirements for the training data of the ID3 algorithm. The first one is that all of the training data objects must have common attributes and these attributes should be previously defined. The second requirement is that the attributes' values should be clearly indicated and a value indicating a special attribute should indicate no more than one state. The third requirement is that there must be enough test cases to distinguish valid patterns. After that, the training data is provided; the ID3 would complete its task according to following algorithm:

Input:

- S-learning data set; R a set of non-categorical attributes, C the categorical attribute

Output:

- Decision tree

1: If S is empty, return a single node with the value "Failure".
2: If S consists of records all with the same value for the categorical attribute, return a single node with that value.
3: If R is empty, then return a single node with that value, the most frequent of the values of the categorical attribute.
4: Let D be the attribute with largest gain $G(D,S)$ among attributes in R.
5: Let $\{d_{j=1,m}\}$ be the values of attribute D.
6: Let $\{S_{j=1,m}\}$ be the subsets of S consisting respectively of records with value d_j.
7: Return a tree with root labeled D and arcs labeled $\{d_{j=1,m}\}$ going recursively for each $\{S_{j=1,m}\}$ to build the tree : $ID3(R - \{D\},\ S_j)$.

Algorithm 2.3. ID3

[7] Best attribute refers to the highest node in a decision tree, or the attribute who divides the observations in the fairest way, in order to meet the target.

For example, considering the primary data of table 2.7, the information entropy of S is $E(S) = -9/14log_2(9/14) - 5/14log_2(5/14) = 0.94$ bits. Supposing S is a set of 14 training test samples in which one of the attributes is "*wind speed*"; the values of the wind speed attribute can be "*Weak* or *Strong*". The classification of these 14 training test samples is that 9 samples are categorized YES and 5 are found in the NO category; for attribute wind speed, suppose there are 8 occurrences of $Wind = Weak$ and 6 occurrences of $Wind = Strong$; for $Wind = Weak$, 6 of the examples are YES and 2 are NO; For $Wind = Strong$; 3 are YES and 3 are NO. Therefore, $Gain(S, Wind) = E(S) - (8/14 * E(S_{weak}) - 6/14 * E(S_{strong})) =$ 0.940 - (8/14)*0.811 - (6/14)*1.00 = 0.048. To form a decision tree like the one in Figure 2.6 for each attribute, the gain is calculated and the highest gain is used in the decision node.

The general advantage of ID3 is that it builds the fastest and short tree; however, data may be over-fitted or over-classified. If a small sample is tested, only one attribute at a time is tested for making a decision, and classifying continuous data may be computationally expensive. ID3 is a non-incremental algorithm, meaning that it derives its classes from a fixed set of training instances[8]. The classes created by ID3 are inductive, meaning that, given a small set of training instances, the specific classes created by ID3 are expected to work for all future instances. A limitation of ID3 is that the distribution of unknown conditions must be the same as the test cases and the induced classes cannot be proven to work in every case since they may classify an infinite number of instances[9].

[8]An incremental algorithm revises the current concept definition, if necessary, with a new sample.
[9]Note that ID3 (or any inductive algorithm) may misclassify data.

2.4.3.2.1.2 C4.5 algorithm

The C4.5 is a technically improved version of the ID3 algorithm, which was discussed in the previous section. Compared to ID3, C4.5 handles both continuous and discrete attributes; to do this, it creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it. C4.5 handles training data with missing attributes. Missing attribute values are simply not used in gain and entropy calculations. Moreover, it handles attributes with differing costs. Finally, C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes (pruning). In order to optimize the final result, a data point is taken and compared several times, which leads to relatively high temporal complexity in this algorithm. Some experimental work in [81] demonstrates that although C4.5 provides more efficient and precise results as compared to other famous classifiers such as SVM [77] and naive Bayesian [82] methods, it does its task in a longer time. Experimentally, we infer that although C4.5 elapses the time for building quite complex models with decision trees that include several levels, it works comparatively better rather than its competitors when not more than 5000 attributes for a problem are considered. Such a high quantity of attributes exists in text mining problems; however, in a normal activity recognition problem, no more than 800 world features are observed.

2.4.3.2.1.3 Application of a decision tree in activity recognition

Because a decision tree is a sort of popular modeling technique, and the corresponding models are predictive and descriptive, they are applied in activity recognition [7, 12, 52] projects. In [52], the decision tree is applied in order to detect if the undergoing activity or event in the Smart Home is usual (normal). In order to make a decision automatically in the Smart Home based on several activity features, a decision tree could help to identify which factors should be taken into account. Moreover, they indicate how each feature has historically been associated with different outcomes of a decision. Decision trees could also be helpful to create an environment which is capable of reacting intelligently by anticipating, predicting and making decisions with signs of autonomy. In [52], the data describing the whereabouts of a person, her/his interaction with appliances and the duration of specific events is collected. A decision-tree algorithm is then used to create a model as either a graphical tree or a set of text rules that can define the normal setting leading to a particular event in the Smart Home. Any event occurring outside the normal setting should lead to various degrees of awareness/alarm/alert in the system or some sort of automatic behavior (see Figure 2.7).

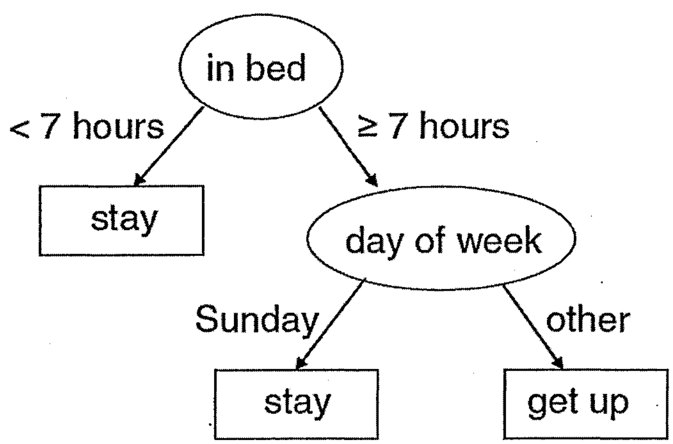


Figure2.7.: Tree structure of a decision tree of activities

Its visual presentation makes the decision tree model easy to understand, so it is suitable for human inspection as well as for making automated decisions. A typical decision-tree based approach requires that the attributes describing the state of the Smart Home have finite value sets. For example, the attribute “cooker” may have a finite number of values (e.g.: *on for* 30 *minutes, on for* 40 *minutes, and on for more than* 50 *minutes*). For example, in [52] the following example is considered: "*…when the person entered the kitchen, he turned the light on; he stayed there for 30 minutes and then left the kitchen forgetting to turn the light off. So, the light remains on for more than 2 hours.*" This could be mapped in the following sequence of events presented in the following table where the current state of the Smart Home would correspond to the attribute-value pairs.

Table2.8. Some observations in the work of [52].

Attribute	Value
at living on	false
at reception on	false
at kitchen on	false
kitchen light on	true for 2 hours
cooker on	true for 2 hours
at reception on	false
at bedroom on	true for 1.5 hours
inbed on	true for 1.5 hours

After the observations are gathered, the expert opens a new column and assigns each record of observation to an activity and he also marks which records are normal (usual). In order to go about induction and make the decision tree, the ID3 algorithm is applied and the decision tree which describes the learned knowledge of the Smart Home is drawn. Figure 2.8 illustrates a part of a decision tree in [52]. As it is illustrated by gathering statistical information about occurring events in the Smart Home, the probabilities required in order to recognize the world's normality status are generated.

One could briefly summarize the reviewed work by saying that it regards the activity recognition problem as a sort of multi-attribute decision-making problem. As the number of knowledge levels is relatively high, the best way to demonstrate this knowledge is to apply decision trees. Moreover, decision trees are sorts of predictive and descriptive tools which may also be used for reasoning. For example, we could reason about the normality of the world state. This work reveals an effort to bring automatic reasoning services to the Smart Home.

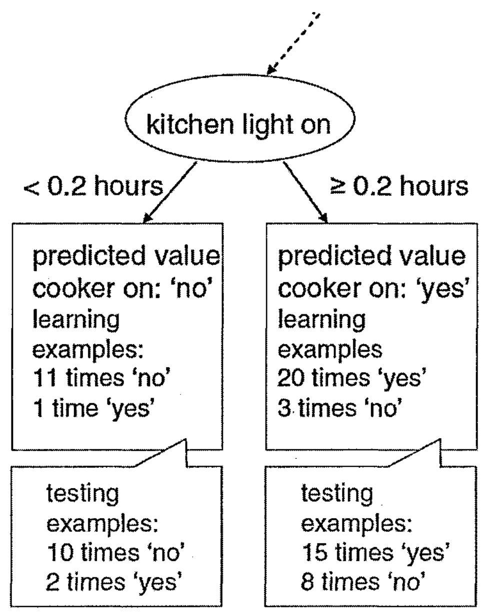


Figure 2.8. A part of a decision tree induced from 35 examples in [52].

Except for the error rate of this work, a critical problem refers to its training phase. In this step, the expert should assign observations to the target class of activities. Moreover, he or she should supervise training. The effect of this problem would also remain in the recognition phase. During this step, the system would once again require the expert's idea to preprocess the data by cleaning and removing noise and should indicate which data record corresponds to which activity; then, the system would say if the activity was completed correctly or not.

Moreover, activity duration is not recognizable in a data-driven manner. Except for inflexible and rigid definitions of classes, the second critical problem which makes this solution impractical for the Smart Home refers to the quantity of training tests. As is illustrated in Figure 2.8, in order to calculate a probability for a machine-state transition, at least 35 realization of that part of activity is needed. Considering that in a LIARA Smart Home, except for multi-state attributes such as temperature, there are at least 100 observed bi-status activity attributes, and the combination of these attributes would result in many machine states (2^{100}), and also considering that the time of transition between nodes should be taken into account (for example, 2 hours or more according to Figure 2.8), thus training time would necessitate over $2 * 35 * 2^{100} hours$. Furthermore, a small change in Smart Home observatory facilities would cause invalidity of knowledge. Considering the reasons we have mentioned, we recommend performing even more improved work; for example, in [7], it is proposed that instead of considering classification as the core of data mining, it be accomplished during the last step of data mining.

2.4.3.2.2 Support vector machines

A support vector machine or SVM [77] is a sort of classification algorithm that typically assigns input data to a target variable. To do this task, it makes continuous spaces for groups of discrete data points so that it can reason in order to proceed with the classification of unobserved but approximately familiar data points. For example, an SVM classifier is able to consider two variables drawn in a virtual space.

Considering a schema presented in Figure 2.9, we can see the input data of a typical classification problem in which the x marks are known to be assigned to a specific class (*class 1 reds or x marks*) and the circle points are known to be another specific class (*class 2 or blues).*

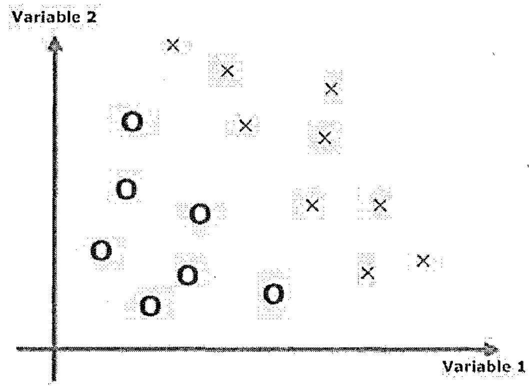


Figure2.9. A typical classification problem

Now, an SVM algorithm would try to find linear or non-linear borders that separate the elements of the classes (see Figure 2.10).

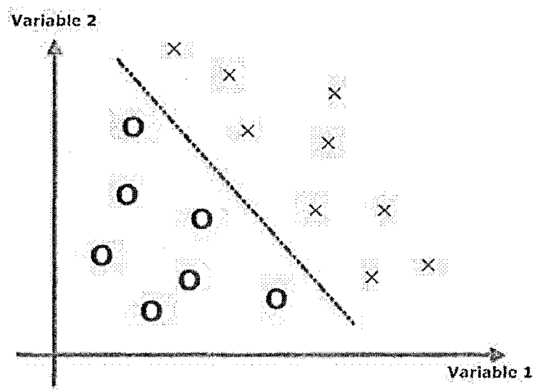


Figure 2.10. SVM bordering

In Figure 2.10, it is shown that *a typical SVM would put virtual frontiers between the elements of different classes*. In order to calculate the best frontiers, applying an optimization process inside the SVM which acts depending on the input and output data specifications, the data points are classified into classes and the best separators are outputted by the SVM. Now, if a new input data point at the running time is inputted, the SVM would apply its drawn borders in order to classify the new inputs (see Figure 2.11).

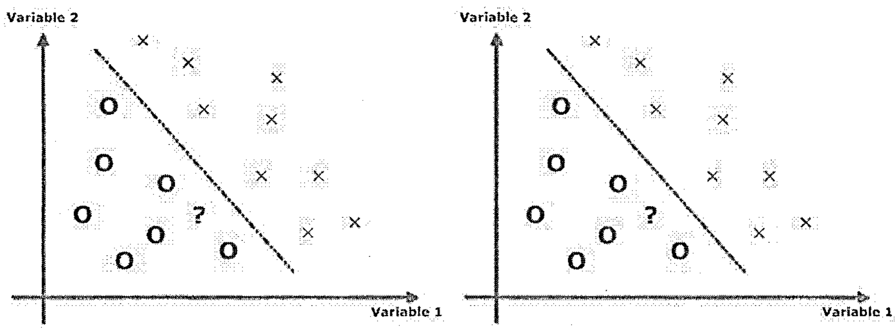


Figure2.11. Classification of new data points (here, new data point is shown by"?")

In Figure 2.11, a new data point is inferred to be in the "blues'" class, while it is on the border of that class. Depending on the optimization technique inside the SVM, which are generally linear or non-linear methods, we are able to classify SVM types. This is besides the grouping of SVMs in two groups of bi-class and multiclass SVMs.

Bi-class SVMs can reason in assignment to only two target classes; however, a multiclass SVM reasons in assignment of input data to more than two classes. The important point here is that by combination of two or more bi-class SVMs, we are able to make a multiclass SVM; therefore, in order to present the SVM functionality, we would present a linear bi-class SVM. For more information about more sophisticated SVMs, we recommend that you refer to [77, 83].

2.4.3.2.2.1 Linear SVMs

Linear SVMs are ideal for classification problems where there are a few training samples[84]. Considering two classes of points, labeled with their output $\{-1, 1\}$, and the fact that we have a set of N vectors $x_i \in X \subset \mathbb{R}^d$, $i \in [1; N]$, where d is the dimension of our input space with their associated class $y_i \in \{-1, 1\}$: thus, supervised learning is the problem of inferring a function f so that:

$$f: X \subset \mathbb{R}^d \to \{-1; 1\}$$

From a set of observations which will correctly classify the maximum number of vectors x_i, this will correctly describe the phenomenon responsible for separation between the two classes so that a new and unknown point will be classified into the right category (capacity of generalization of the classifier). This problem can be solved with multiple existing algorithms. A simple method based on perceptron [85] builds a linear separation starting point with random initialization and followed by testing of the different points in the training database, so as to adjust the separation until it classifies a maximum number of points from that database correctly. Vapnik [86] designed another algorithm, based this time on linear separation, but which tries to maximize the margin between the separation and the nearest points of the training database in each class [87]. This margin will ensure maximum "safety" for generalization of the algorithm and its application to new points. Support vector machines are equivalent to the construction of a hyperplane of equation $w^T x + w_0 = 0$, where w and w_0 are the equation parameters of the hyperplane to be computed. From this hyperplane, we build function f given by:

$$\begin{cases} w^T x + w_0 > 0 \Longrightarrow f = 1 \\ w^T x + \omega_0 < 0 \Rightarrow f = -1 \end{cases}$$

where f represents the output of the algorithm for a new point x_i, output that allows us to classify x_i as belonging to one of the two classes. To build the hyperplane, we have to solve the following equation that maximizes the distance between the closest points of each class and its separation:

$$\underset{w, w_0}{argmax} \quad \min_{i=1..N}\{\|x - x_i\| : x \in \mathbb{R}^d,\ w^T x + x_0 = 0\}$$

This is done by solving the following linear problem:

$$min\frac{1}{2}\|w\|^2$$

$$s.t.\ f(w^T x_i + x_0) \geq 1, \qquad i = 1..N$$

The solution to this problem is the saddle point of the Lagrangian function:$L_p = \frac{1}{2}\|w\|^2 - \sum_{k=1}^{N} \alpha_k (f(w^T x_i + w_0) - 1)$

In this equation, coefficients α_k are the Lagrange multipliers.

$$\begin{cases} \sum_{\alpha_k > 0} f(\alpha_k < x, x_i > +w_0) > 0 \Rightarrow f = 1 \\ \sum_{\alpha_k > 0} f(\alpha_k < x, x_i > +w_0) < 0 \Rightarrow f = -1 \end{cases}$$

x_k are the support vectors, the ones chosen in each class to define the separation, and $\langle \quad \rangle$ is the inner product of the two vectors. This last equation allows for the classification of a new vector x unknown in the training database. This case is limited to the description of the classification of a binary problem that can be linearly separated. Non-linear SVM kernels and more advanced concepts concerning SVMs are problems which surpass the scope of this thesis; therefore, we will limit our discourse to SVM internal organization up to this point and continue this thesis on the applications of SVM classifiers in activity recognition.

2.4.3.2.2.2 Application of SVMs in activity recognition

In point of fact, the SVM has been a popular classifier in some activity recognition research [46, 79, 88-90] because it requires a relatively low quantity of data points for training and it performs an optimization process inside the kernel.

In the non-vision work of [89], activities are observed through 20 sensors within a period of 83 days. Ten activities are identified by the expert and for each activity one SVM is trained with the existing samples. In their work, each data set concerning an activity is a sample. Different optimization methods inside the SVM are compared and results show that the kernel fusion optimization method produces the highest precision in their research. Similar work by [90] is performed in order to find the patterns of 14 activities. By comparing all learned patterns and current observations, anomalies are recognized. In this research, the SVM is trained with probable patterns as normal states and improbable patterns as anomalies. The result is that any observation closed to anomaly patterns is identified as an anomaly.

In the visionary work of [88], after a clustering process using a k-means algorithm, clusters are inputted to SVMs for modeling. Once again in the research process, one SVM is considered for each activity. After finding the best parameters for modeling such as the best codebook, camera view and episode duration, the accuracy of reasoning is presented. A similar logic is applied in the work of [79], in which activity patterns are clustered and presented in a hierarchy. In order to find an ongoing activity, a decision tree is used (see Figure 2.12).

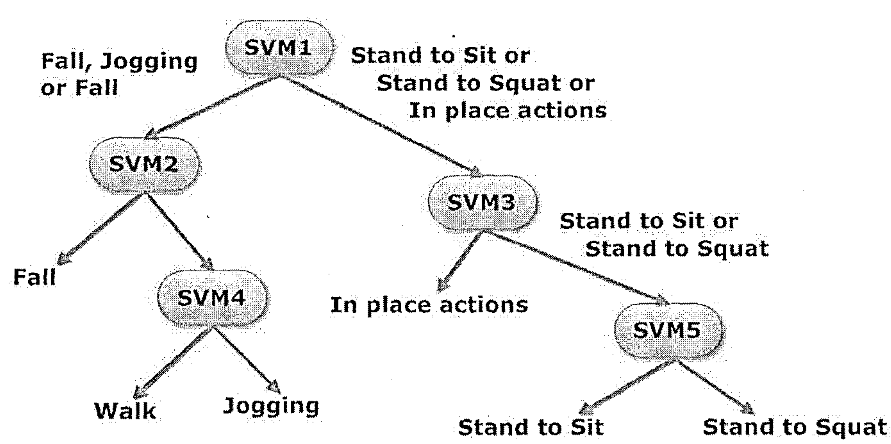


Figure 2.12. Hierarchy of activities

In this research, only five SVMs are applied in order to recognize nine activities. For each node of the decision tree, an SVM is applied and each one contains patterns of two or more activities. The accuracy rate of this study is reported to be more than 80%.

In brief conclusion of SVM application in activity recognition, we recognize a high accuracy rate in most of the concerning research. The SVM is able to handle the imprecision and uncertainty problem in one way or another. An important challenge in SVM utilization is the increase of process complexity and inaccuracy when the number of patterns and data attributes goes up. Despite the mentioned challenge, the SVM is a supervised approach and it depends highly on the expert idea to label input data; however, the most important problem of the concerning studies refers to its inability to perform real-time reasoning. In other words, the mentioned approaches have to wait for completion of activities in order to reason which known activity was accomplished and whether it was done correctly or not. In this case, an expert is required to identify the beginning and ending points of activities regardless of their correct or incorrect realization. Therefore, a pure application of the SVM may lead to inefficient results and rigid rules in the realization of activities at recognition time.

2.4.4 Temporal-pattern discovery

Temporal-pattern discovery [23, 25] deals with the discovery of temporal patterns of interest in time series or temporal sequences where interest is determined by domain and application. One characteristic of frequent observation of the world is that its observed features may be similar to previous observations. In other words, we would see a lot of repetitive data records in the data set. In temporal-pattern discovery, not only are observations taken into account, but the duration of observations in their last state should be considered. Moreover, the discovery of a sequence of events, which means the transition from a machine state to another machine state that happens frequently, would be helpful to predict the observed system's behavior. Anomaly recognition is also another application of such techniques. In activity recognition, by surveying temporal features of activities like duration, beginning time, ending time, intervals and consideration of temporal uncertainty in the accomplishment of a sequence of actions, this provides valuable information and criteria to recognize activities. This can be seen in a wide range of research such as [44]. In this part of the thesis, we will introduce, describe and discuss temporal-pattern discovery methods such as frequent-pattern mining and algorithms which are applied in temporal-pattern discovery such as SHIP, ALZ and Apriori [25, 50].

2.4.4.1 Frequent-pattern discovery of activities

In [44, 90], temporal pattern discovery based on Allen's temporal relations is applied to discover interesting patterns and relations on Smart-Home data sets. These studies describe a method for discovering temporal relations in Smart-Home data sets and applying them to perform an anomaly-detection process on frequently – occurring events. These works attempt to recognize Allen's thirteen temporal relations through data by analyzing the lengths of time during which sensors are stable in their statuses (see Figure 2.13). After this, the temporal intervals of activities are calculated. In the second step, frequent activities or events, which occur during the day, are identified. The Apriori algorithm [91] is applied to proceed with the mentioned task. During third phase, out of the thirteen Allen's temporal relations, only the *"before", "contains", "overlaps", "meets", "starts", "started by", "finishes", "finished by" and "equals"* relations between activities are calculated. These relationships are indicated based on observed probability in training data. Finally, an anomaly can be detected if a very improbable event occurs. Anomalies can be recognized in activities sequences and suspicious states. Here are, in short, the process steps taken from this research:

1. Identification of daily activities or events which happen frequently;
2. Identification of observed temporal relations between events;
3. Calculation of evidence of event occurrence that can be used for calculating anomalies.

Temporal relations	Visualization	Interval constraints
X Before Y **Y After X**	X Y	Start(X)<Start(Y); End(X)<Start(Y)
X During Y **Y Contains X**	X Y	Start(X)>Start(Y); End(X)<End(Y)
X Overlaps Y **Y Overlapped-by X**	X Y	Start(X)<Start(Y); Start(Y)<End(X); End(X)<End(Y)
X Meets Y **Y Met-by X**	X Y	Start(Y)=End(X)
X Starts Y **Y Started-by X**	X Y	Start(X)=Start(Y); End(X)≠End(Y)
X Finishes Y **Y Finished-by X**	X Y	Start(X)≠Start(Y); End(X)=End (Y)
X Equals Y	X Y	Start(X)=Start(Y); End(X)=End(Y)

2.13. Thirteen possible temporal relations between activities

Anomaly in [23, 90] is defined based on probability theory and it is indicated that if the probability of an event is based on the occurrence of other events which themselves rarely occur, then the evidence supporting the occurrence of the event is not as strong and using probability approach, a case study to measure anomaly has been introduced. A key-point on impracticality of this work is the reason that an event be recognized as abnormal refers to quantity of its occurrence in training test. It is obvious that there are many normal actions that we do rarely and the low quantity of their accomplishment does not mean that they are abnormal. Moreover, in [23], daily time and duration of activities is not considered and this approach is not functional on large data sets. In the introduced work, the Apriori algorithm was applied to find frequent patterns that may occur.

2.4.4.1.1 The Apriori algorithm

Apriori is the most widely used algorithm for the discovery of frequent itemsets (patterns) and association rules [25]. The main algorithm of Apriori is as follows:

Input:

- S learning data set; minimum support and confidence thresholds

Output:

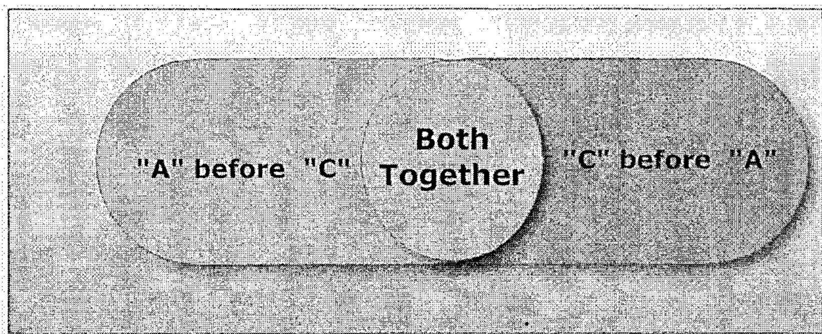
- Set of frequent itemsets

1. Any subset of a frequent itemset is a frequent itemset
2. The set of itemsets of size k will be called C_k.
3. The set of frequent itemsets that also satisfy the minimum support constraint is known as L_k.
4. C_{k+1} is generated by joining L_k with itself.
5. L_{k+1} is then generated by elimination from C_k, those elements that do not satisfy the minimum support rule.

Algorithm 2.4. Apriori

Any recovered association rule by Apriori is expressed with two statistical elements of information that are *confidence* and *support*. The *confidence measure* refers to the quantity of observation of the recovered rule without consideration of the sequence of predicates in the training test set; however, support refers to the quantity of observation of the recovered rule with consideration of the sequence of the predicates. A main consideration in the Apriori algorithm is that any subset of a frequent itemset must be frequent.

For example, by selecting the minimum support and confidence setting as 50%, we would like to find all rules $X\&Y \to Z$ from an activity database. In this example, we intend to verify the existing relations between two activities, the A *and* C. These relations are: A occurs *before* C, or C occurs *before* A, or both of them occur together (see Figure 2.14).



Time	Ongoing Activity
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

2.14.: Association rule recovery from database that holds transactions of activities

By support we mean the probability that a transaction contains $\{X, Y, Z\}$ and by confidence we mean the probability that a transaction having $\{X, Y\}$ also contains Z. In this example, considering the problem constraints, the rule $A \to C$ is recovered by support of 50% and confidence of 66.6%, and we illustrate it is as being $A \to C$ (50%, 66.6%). Another rule is recovered as $C \to A$ (50%, 100%). The occurrence of C while A being accomplished (if C has never occurred before) is a normal and correct chain of activities. The prediction is another application of temporal pattern discovery. In the next part of this thesis, we will review research that aims to make predictions in the Smart Home by applying time-series data mining.

2.4.4.1.2 The SHIP algorithm used to make predictions in the Smart Home

Prediction in the Smart Home is a key point in achieving the ultimate goals of Smart Home design. In quantitative approaches, prediction refers to the most probable events that would be observed in the future according to previously observed sequences of events. In [50], the SHIP algorithm is proposed to find the most probable future-happening events. The logic behind Apriori is consequently applied in the SHIP algorithm; however, this time, instead of looking for relationships between two previously determined events, they search for recent sequences of events that occur frequently. Therefore, in contrast to Apriori, the passing-through data is repeated for each possible combination-of-events set. At the core of the SHIP algorithm, the sequence of events that are valid in the following formula would be $R_t(s,a) = \alpha\, l_t(s,a)/\sum_t l_t(s,a_i) + (1-\alpha)\, f(s,a)/\sum_t l_t(s,a_i)$, in which $l_t(s,a)$ is the length of the longest sequences that end with action a in state s at time t. The function $f(s,a)$ is the number of times that action a has been taken from the current state. The SHIP algorithm matches the most recent sequence of events with sequences in collected histories. When the inhabitant issues a command to a device, it is recorded in the inhabitant history. A match identifies a sequence in the inhabitant history that matches the immediate event history. A match queue is maintained to ensure a near-linear run-time. The SHIP algorithm consists of two steps. First, the match queue is updated when a new action is recorded. At time t in state s, we compute $l_t(s,a)$ and $f(s,a)$ from the current state. It returns action a corresponding to the greatest $R_t(s,a)$ value as its prediction. The user can refine the algorithm by specifying a decay factor for old history or allowing an inexact match. A key disadvantage of SHIP is the fact that the entire action history must be stored and processed off line, which is not practical for large prediction tasks over a long period of time. We classify the SHIP algorithm as a quantitative sequence-mining algorithm, therefore except for the need of a high amount of training data, reasoning in prediction would depend on a quote of the quantity of the observations (regarding each other) in the training set.

2.4.4.1.3 The ALZ algorithm to make predictions in the Smart Home

In [50] it is proposed that the Active LeZi (ALZ) algorithm can be more accurate than the SHIP algorithm in frequent-pattern mining. ALZ uses information-theory principles to process historical action sequences. By predicting inhabitant actions, the home can automate or improve on anticipated events that inhabitants would normally perform in it. To make these predictions, the ALZ algorithm calculates the probability of each event (inhabitant action) occurring in the parsed sequence, and then predicts the action with the highest probability. To achieve optimal predictability, it uses a mixture of all possible higher-order models (phrase sizes) when determining the probability estimate. Specifically, they incorporate prediction by partial-match strategy of exclusion to gather information from all available context sizes in assigning the next symbol its probability value. ALZ does its task based on the following algorithm:

Input:

- Sequence action, minimum support and confidence thresholds

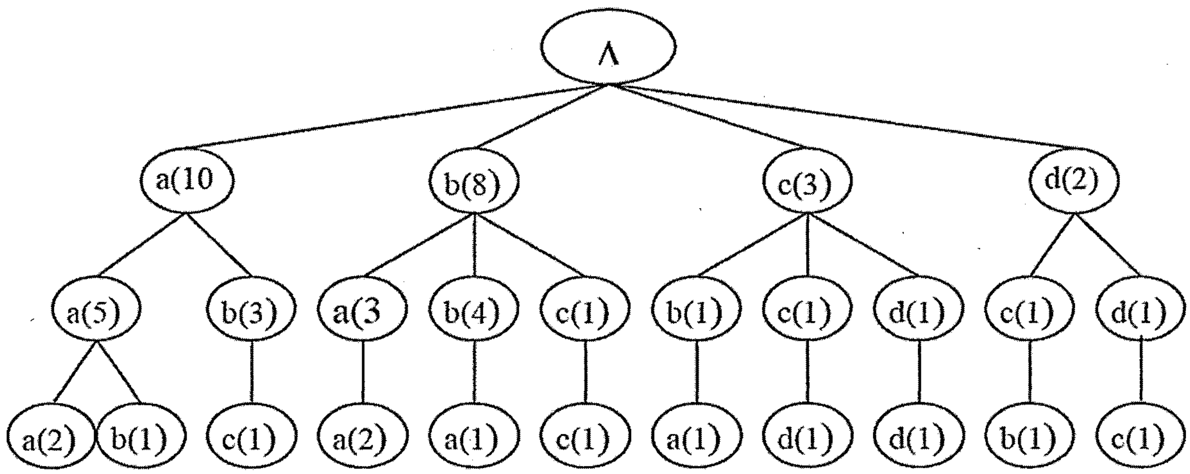
Output:

- Probabilistic action tree

1. Initialize
$dictionary = null;\ phrase\ w = null;\ window = null;\ \max\_lz\_length = 0$
2. Loop
3. Wait for next symbol v
4. $If\ (w, v)\ \in dictionary\ then\ w = w.v$
5. $Else\ add\ (w, v)\ to\ dictionary$ update $\max\_lz\_length$
6. $w = null\ endif$
7. $Add\ (w, v)\ to\ window$
8. $If\ (length(window) \geq$ $\max\_lz\_length$ delete $window\ endif$
9. Update frequencies of all possible contexts within window that includes v
10. Forever

Algorithm 2.5.: ALZ

For example, considering "$aababbbbaabccddcbaaaa$" as a series of actions in the Smart Home, the window maintained by Active LeZi is the set of contexts used to compute the probability of the next symbol. Here, the last phrase "aaa", which is also the ALZ window, is used. Within this phrase, the contexts that can be used are suffixes within the phrase, except itself (i.e. "aa", "a" and the null context). Suppose the probability that the next symbol is an "a" is being computed. From the following Figure 2.15, we see that an "a" occurs two out of five times that the context "aa" appears, the other cases producing two null outcomes and one "b".



2.15.: Tree formed by the ALZ parsing of the string "$aababbbbaabccddcbaaaa$"

Therefore, the probability of encountering an "a" in the context "aa" is $2/5$ and we now fall back (escape) to the order-1 context (i.e. the next lower order model) with probability $2/5$. In the order-1 context, we see an"a" five out of the ten times that we see the "a" context, and of the remaining cases, we see two null outcomes. Therefore we predict the "a" at the order-1 context with a probability of $5/10$, and escape to the order-0 model with a probability of $2/10$. At the order-0 model, we see the "a" in ten out of 23 symbols seen so far, and we therefore predict "a" with probability of $10/23$ resulting in a null context. The blended probability of seeing an

"a" as the next symbol is therefore $(2/5 + 2/5(5/10 + 2/10(10/23)))$. Similarly, let us compute the probability that the next symbol is "c". In this case, the order-2 and order-1 contexts do not yield "c". Therefore, we escape to the order-0 model and predict "c" with a probability of 3/23. In this case, the total probability of seeing "c" would be; $(0/5 + 2/5(0/10 + 2/10(3/23)))$.

2.4.4.2 Frequent episode discovery

Frequent Episode Discovery or FED [25] deals with the problem of finding temporal patterns of interest within the framework of discovering frequent episodes in a temporal-data sequence. FED is justified in a context where there are important data-mining application areas in which the data to be analyzed consists of a sequence of events. One basic problem in analyzing event sequences is finding frequent episodes. In this context, an episode refers to the collection of events that occur relatively close to each other in a given partial order and a temporal pattern indicates the interval or duration that the events are expected to be probably observed. Recovery of frequent episodes considering temporal constraints is the result of this algorithm. The overall goal is to analyze sequences of events and discover recurrent episodes. Regarding figure 2.16, which is a sequence of events where each event has an associated time of occurrence; note that it is considered as input for FED.

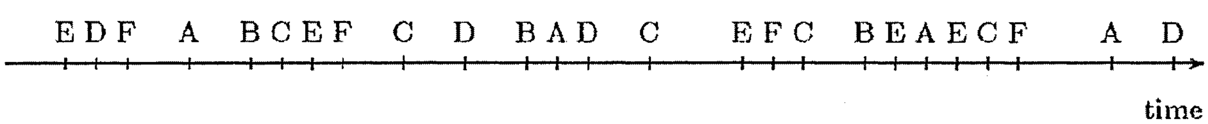
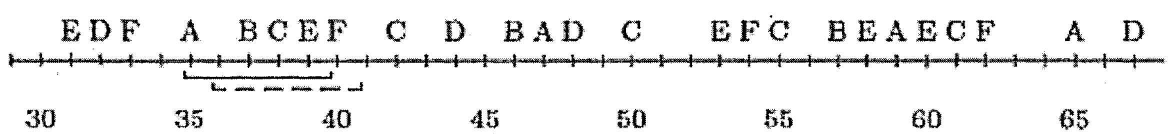


Figure 2.16.: Sequence of events on time line

Given a set E of event types, an event is a pair (A, t) where $A \in E$ is an event type and t is an integer, the occurrence time of the event where time is considered discrete. The event type can actually contain several attributes; for simplicity, we consider here only the case where the event type is a single value. An event sequence s on E is a triple $< s,\ T_s,\ T_e >$ where s is a set $\{(A_1, t_1), (A_2, t_2), \ldots, (A_n, t_n)\}$ of ordered events such that $A_i\ \in E$ for all $i = 1, n$ and $t_i < t_{i+1}$ for all $i = 1, n-1$. Further on, T_s and T_e are integers where T_s is called the starting time and T_e the ending time, and $T_s \leq t_i \leq T_e$ for all $i = 1, n$.

Figure 2.17 illustrates the event sequence $\langle s, 29, 68\rangle$ where $s = \{(E, 31),\ (D, 32),\ (F, 33,),$ $(A, 35),\ (B, 37),\ (C, 38),\ \ldots,\ (D, 67)\}$. In the analysis of sequences, we are interested in finding all frequent episodes from a class of episodes. To be considered interesting, the events of an episode must occur close enough in certain span of time. The user defines how close is close enough by indicating the width of the time window within which the episode must occur. The time window is defined as a slice of an event sequence, and then considered an event sequence as a sequence of partially overlapping windows. In addition to the width of the window, the user specifies in how many windows an episode has to occur to be considered frequent.



2.17.: An example event sequence and two windows of width 5.

Formally, a window on an event sequence $< s,\ T_s,\ T_e >$ is an event sequence $<w, t_s, t_e>$ where $t_s \leq T_e$ and w consists of those pairs (A, t) from s where $t_s \leq t \leq t_e$. The time span $t_e - t_s$ is called the width of the window w, and it is denoted width (w). Given an event sequence s and an integer win, we denote by $w(s, win)$ the set of all windows w on s such that width $(w) = win$. By definition, the first and last windows on a sequence extend outside the sequence, so that the first window contains only the first time point of the sequence, and the last window contains only the last time point. With this definition, an event close to either end of a sequence is observed in just as many windows to an event in the middle of the sequence. Given an event sequence $< s,\ T_s,\ T_e >$ and a window width win, the number of windows in $w(s, win)$ is $T_e - T_s + win - 1$.

For example, in figure 2.17, two windows of width 5 on sequence s are illustrated. A window starting at time 35 is shown in solid line, and the immediately following window, starting at time 36, is depicted with a dashed line. The window starting at time 35 is $< \{(A,35),(B,37),(C,38),(E,39)\}, 35, 40 >$. Note that the event $(F,\ 40)$ that occurred at the ending time is not in the window. The window starting at 36 is similar to this one; the difference is that the first event $(A,35)$ is missing and there is a new event $(F,40)$ at the end. The set of the 43 partially overlapping windows of width 5 constitutes $w(s,5)$; the first window is $< \varphi, 25, 30 >$, and the last is $< \{(D,67)\}, 67,72 >$. The event $(D,67)$ occurs in 5 windows of width 5, as does, for example, event $(C,50)$.

This method performs similarly to the Apriori, but instead of surveying all of the subsets of a sequence, only predefined ones are surveyed and the output would be, similarly, the statistical information about the chance of occurrence of any mentioned subsequence.

2.4.5 Summary on temporal data mining algorithms

Clustering algorithms try to provide basic classes in which the data of data sets can be classified. The clustering process decreases the role of the expert in the temporal data-mining process and innovation here would lead to an even greater decrease in the role of the expert which is a general desired task of artificial intelligence. Clustering can be done based on internal relationships between data points: it is then called relative clustering. C-means is an algorithm that accomplishes the clustering task based exclusively on the internal relationships between data points. The second strategy of clustering refers to methods that exclusively concern external parameters. For example, these do not take into account if a relatively large number of data points are concentrated around a special position, a strategy known as absolute clustering. K-means is an instance of this group of algorithms. The third strategy of clustering refers to consideration of both the internal and external relationships of data points, and fuzzy subtractive clustering is an approach which accomplishes this task better than other ones.

Classification algorithms classify data input in a way that we define as being how a target variable would be met. Therefore, these algorithms would make a temporal data set become a chain of cause and effect. Therefore, we can predict future events by applying these algorithms. Each of the classification algorithms work in a special context, but we believe that C4.5 is the best classification algorithm for activity recognition because, regardless of its accuracy, it acts quickly enough while a limited number of real-world features are observed (less than 5000 attributes).

Temporal pattern-discovery methods deal with the interesting matter of prediction in huge data sets that have a lot of data records. They translate a large data set into a probabilistic cause and effect model and provide facilities to distinguish probable events from improbable ones. Each of the temporal pattern-mining algorithms has particular advantages and disadvantages. Apart from the particular disadvantages of FED, ALZ, SHIP and Apriori algorithms, a common weakness of these algorithms is the fact that they do not predict the time of event occurrence and, accordingly, they cannot verify if an activity is realized correctly or not. A second problem is that they should receive observations in predefined and rigid structures, so that they cannot be applied in reasoning for the multi-state sensors such as temperature, RFID, or light measurers. Moreover, for cases in which the environment is observed through several sensors, these algorithms would require a lot of time for pattern mining. The last difficulty is that the input type of the mentioned algorithms is "action" or "event", and processing to infer the events or actions from data is a complicated task. In all of the research considered here, it is the task of expert to apply his or her mind in order to provide the input.

2.5 Discussion on qualitative vs quantitative approaches

In this section, we intend to briefly review the fundamental differences between quantitative and qualitative machine-learning approaches. Basically, quantitative approaches are more appropriate for problems that contain a limited number of data sources on one hand and, on the other hand, when there are a lot of observations or data concerning the problem. In such approaches, the final result is indicated with success-probability and confidence measures. Quantitative approaches face less imprecision if more observations are provided during the training phase. As consideration of more attributes causes more machine states, a minimum of training samples is needed to calculate the probability of transition between states; then, training complexity would be increased. Furthermore, uncertainty is probably increased as the contradictive data that should be neglected would probably increase.

In contrast to quantitative machine learning approaches, qualitative approaches better fit problems that contain both a limited number of observations and several data sources. In such cases, the final result is indicated with similarity degrees that are calculated according to target concepts. Qualitative approaches face less uncertainty if more data sources are provided during the training phase. Consideration of more attributes causes more criteria for observation classification and leads to greater better uncertainty in reasoning while assigning observations to the right categories. If more observations are provided to train a qualitative model, then imprecision in observations can be resolved. Statistical methods are applied in quantitative data mining algorithms. In the methods mentioned in this thesis, probability theory is applied to calculate the likelihood of occurrence of events. One important point to note is that the input of any probabilistic approach (like quantitative data-mining approaches) should have random distribution. In our perspective, this means that probability law does not allow the intentions of experts to interrupt training tests, which also signifies that the quantity of occurrences would affect generated understanding and that the target agent (Smart Home resident) being surveyed for activity recognition research *should behave completely normally.*

One interpretation about the fact that the input of a probabilistic approach should have random distribution is that, by applying quantitative approaches, the expert is not able to impose his or her *knowledge* to such models. Furthermore, the expert should make sure that his observations obey normal distribution, which means that at least a minimum of is needed to find the concerning probability of a state. In [29] it is indicated that 250,000 observations are required to calculate only the probabilities of activities beginning times is done. Another interpretation is that statistical approaches are applicable to the problems for which the expert has absolutely no knowledge and, by counting techniques (existing in probability theory), he tries to make superficial knowledge (in contrast to deep knowledge and data structure or data pattern). We reason that the main goal of activity recognition is to learn possible activity patterns and recognize activities depending mostly on the ways that these are done. In fact, activity recognition is a pattern-mining problem; however, quantitative approaches output the chance of realization of activities based on environmental conditions. For example, at 12 o'clock, the probability that the "Eating" activity will begin is 0.6 in [29].

We use the term qualitative approach with those that are preoccupied with the quality of activities realization and able to verify the normality of the world state. In the research of [21], which applies the lattice theory, the possibilities of next actions and states are discussed. Alzheimer's patients are investigated to understand what kinds of errors they make and these patients are classified based on their disease advancement degree. This approach is a great introduction to the necessity of the Smart Home and one finds valuable information in cognitive assistance provision which is not part of the current research project. In [51], an approach which applies the possibility theory is introduced. In short, it can be said that this research explores the possible contexts in which activities may be realized, are surveyed and assigned possibility distribution degrees. Finally, activities that can be conducted in different contexts are ranked based on possibility and necessity distribution. The aforementioned approach is logic-based, and it tries to manage existing uncertainty and imprecision in data. It is assumed that activity recognition is a kind of problem that has partial information and its uncertainty is caused because of this fact. The aforementioned approaches are logic based, and the source of knowledge is not data-driven. To improve these qualitative approaches, we proposed our research project in order to discover the fuzzy temporal mining that can explain observations and recognize the normality of world state as its ultimate result.

2.6 Conclusion

Recently proposed works on activity recognition show effective but *unreliable* results. These are still dependent on the expert's knowledge in both learning and recognition steps; on one hand, researchers presume that activities are realized in an ambient environment, but on the other hand, they recognize each activity by consideration of only a few special attributes. Therefore, they do not propose a reasoning system that is capable of interpreting all of the possible events that may occur every time and everywhere in the ambient environment. The result is that they cannot verify *correct realization of activities*.

One more major reason that makes these approaches impractical is that their reasoning system is not flexible enough to handle existing uncertainty in input data for modeling, especially since it is not capable of distinguishing in what context and which input arguments may play more important roles in activity recognition. In other words, activities are expected to be performed in standard and rigid structures in order to be recognized. Furthermore, even if this limitation is met, the approach will not be able to output a certain decision and the uncertainty in this output (its unreliability) is a permanent characteristic of these approaches.

Fundamentally, quantitative probabilistic approaches propose consideration of some specific *data points* (as the core of knowledge) that differentiate between input states and those that can predict output states. In other words, they discover points that can establish a linear or nonlinear relation between input and output states, but the limitation is that the input argument must completely match the mentioned determinative points in order to be able to predict output. On the other side of the spectrum, there are qualitative and possibilistic approaches [13, 51] that draw a possible *space* as the core of knowledge for observed scenarios. These relate the input space to the output space and indicate positioning in the part of the input space that would lead to the activation of certain possible outputs. More similarity degrees in known input states would result in a better output prediction. They discover that a correct scenario must be realized in what space and the input states would define the status of the output states in the sense of the possible ongoing activities.

3 Fuzzy temporal mining model for activities recognition

3.1 Introduction

In the current chapter, we will present two contributions in order to bring solutions for the improvement of activity recognition and data-mining techniques. The first contribution refers to the modeling of the activities and proposes an innovative method of perceiving these. As a second contribution, a fuzzy-logic based mining model is offered to reason world state normality and events that may occur in the future. This model considers the temporal features of activities in recognition through application of the fuzzy-time concept.

Several viewpoints on the activity recognition, Smart Home and the activities entities will lead to different take offs from these subjects. In the context of ambient environment, embedded sensors provide primary data about *home state*, and it is presumed that when an activity is realized, the embedded sensors are actuated. Thus, their states depend on the realization of the activities. Embedded sensors provide data about objects' locations in the home, home appliances' states, electrical devices' states, residents' locations, doors' states and environmental parameters such as light and temperature, etc. In this thesis, activities are regarded as series of events in the sense of changes in world-state quality that occur in the environment; the way that world quality is changed is the subject of activity recognition. Activity recognition acts as a sort of learning agent that analyzes observations, discovers what activity is going on or what is intended to be realized by the Smart-Home resident, predicts world states that will be met in the future and distinguishes between normal and abnormal world states. Hence, the realization of activities is recognizable through the identification of activity statuses. These statuses do not have clear, definite and certain specifications or borders. Quantitative and traditional data-driven machine-learning approaches like the Hidden Markov model [49] and Bayesian networks [31] try to find absolute and certain activity statuses; however, activities do not follow definitive and certain patterns of realization. For example, in [31] to recognize the activity of "drinking water", the glass should be taken from a definite point, called "initial state". If the initial state of activities is not recognized, then the activity as a whole is not recognizable. In [29] some cases, only 13% of confidence from inferred knowledge is expected which is clearly an insufficient percentage to attain reliability. We suggest consideration of the fuzzy event as the forming element of activity models. A fuzzy event can be defined as a switch from one fuzzy state to another. For example, if an object is moved from a table area to a cabinet area, its distance from the table switches from "near" (fuzzy state) to "far" (fuzzy state). This distance switch is a fuzzy event. Therefore, a fuzzy event describes statuses of activities that are defined depending on the existing interrelations of variables [12]. One property of fuzzy events is that they not only indicate the origin state of event occurrence, but also they indicate the future state of activities caused by the occurrence of the fuzzy event.

An important point about this particular subject is that sensors' states indicate the home state, but the home state is different from the *activities' state*. In fact, by analyzing the way that the home state is changed, we are able to recognize activities. Considering the aforementioned point, we can now see why approaches that equalize home state with activity state do not show reliable and practical results in activity recognition [23, 31, 90]. Activities are the entities which characteristically depend highly on space and, especially, the time factor, but a typical activity may be realized in dissimilar times and spaces. In other words, in an ambient environment similar realizations of a special activity may be captured by non-similar sets of sensors (realization in different locations) and, even if these are repeated in the same location, sensors may output dissimilar values. Moreover, the temporal features of activities such as an activity's beginning and ending time and, also, the delays between actions may be dissimilar to other repetitions. Therefore, we can imagine the dynamic characteristic of these activities; however, uncertainty (caused by the intelligence source) and imprecision (caused by sensors) constitute their noticeable features.

3.1.1 Activities' recognition as complex mining problem

In the Smart Home, humans actuate world features non-linearly in order to achieve their goal(s). For example, by tracking an object in the environment (sugar) in the course of realization of the "coffee-making" activity (Figure 3.1), we are able to see non-linear observations.

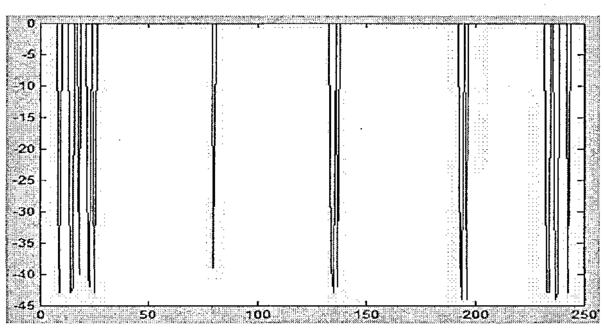


Figure3.1.: Two hundred and fifty observations from the position of "sugar" in the realization of the "coffee-making" activity . RFID antennas observed the RFID tag attached to the sugar to observe the position of the sugar.

Considering that in LIARA more than 500 world attributes are observed, it can be inferred that activity recognition is dependent on many variables that act non-linearly. Imprecision of sensor observation and uncertainty in the realization of activities are part of the reason why human behavior seems non-linear; so even if a human being behaves linearly in order to achieve his or her goals, the perception from an observer's viewpoint is non-linear, making it difficult to predict the future states of the Smart Home. For activity-recognition agent, an intended goal may be a special world state out of the possible normal world states; the final result of activity realization is to keep the world state normal. In [7, 12, 92], we define activities as being a sequence of actions and operations in which each action changes one or more world features in order to achieve an intended normal world state. In some cases, we cannot catch, sense or capture the occurrence of some events or we cannot observe some world attributes, but we can find evidence of the occurrence of these events and infer or calculate these unobservable properties. For example, when at first we feel thirsty (unobservable event for the Smart Home), then we drink water. In order to do that, we take a glass, open the cabinet door, open the faucet, etc. Therefore, a series of observable events have occurred as a consequence of the *thirst* event. After realization of the "water-drinking" activity, we are no longer thirsty (unobservable world state). Although we cannot observe what happens inside a human mind such as "feeling thirsty" or capture some special world states such as "water going down the throat", we can infer such mentioned events and properties through analysis of observable events and properties which may provide us some evidence about "invisible" events and world states. As a result, we consider the realization of an observable-actions' series (activity) equal to the system's goal in the proposed reasoning system, because a survey of how these actions are accomplished would represent the occurrence of unobservable events and realization of the whole activity would indicate the state of unobservable attributes. By modeling these activities, we can recognize them and. if an observation does not match at least one activity model, then anomaly is inferred. In order to assist the resident of the Smart Home, we need to know how successful he or she was in achieving his or her intended goal (s) and, finally, the appropriate assistance for the resident can be inferred. In order to judge normality of the world state, two criteria are considered. One criterion is verification of the possible *context* in which an activity can be accomplished [11] and the other is that of checking if the activity is realized in a correct way [7].

The aforementioned descriptions could better reveal the difficulties of the activity recognition problem and justify the proposal of a new data mining approach for this problem. In this model, we presumed that an activity is a concept that may typically be realized at many times and in numerous spaces with several possible temporal features; the structure of this concept may be discovered through analysis of raw data. The perception of a typical activity is a *conceptual structure* [93] that includes a set of *rules* which justify the occurrence of special events in temporal data sets. In other words, the conceptual structure of a special activity confirms the observations of the sensors that have witnessed activity realization. We use the term *fuzzy conceptual structure* because the rules behind the data are the fuzzy rules discovered by fuzzy temporal data mining techniques. A further description is that a typical activity's fuzzy conceptual structure describes a *virtual space*; this gives more recognition possibility to observations that better match the mentioned space or model. On the other hand, it gives a lower possibility degree to the observations that do not completely match the structure. Finally, according to possibility measures assigned to every existing concept, activities are ranked and the one with the highest possibility degree is the most possible intended activity.

In this thesis, two viewpoints on activities are proposed. In the first viewpoint, activities are regarded as a series of *fuzzy events* that occur in the environment. A fuzzy event is inferred when an action is accomplished in the environment. In this event-driven viewpoint, it is expected that realization of similar activities leads to the occurrence of a similar event series. Therefore, in order to perform activity recognition, we propose an activities-mining model for finding a series of events that is similar to current observations. The second viewpoint on activities refers to a sort of complementary extension of the first viewpoint. Through this viewpoint, we regard activities as fuzzy concepts in the form of a multiple-regression function in which their forming entities are fuzzy events; each fuzzy function of an activity is dependent on a set of role-playing variables which are part of the fuzzy events occurring during the realization of activities. By this viewpoint, we are able to reason in occurrence of two or more simultaneous concepts (activities) [46]. In order to perform activity recognition, the mining model is applied so as to find possible combination(s) of known concepts that may better explain the observations. In other words, around a single reality several hypotheses are made and the hypothesis which better matches reality is selected as being the most possible ongoing activity concept.

3.2 Fuzzy event mining activities

As it was mentioned earlier, observation of activity features provides primary data about realization patterns of actions, operations, activities, plans, objectives and generally any goal that an individual may intend to achieve. In applying extensions of fuzzy logic [6, 61, 94], we have modeled activities as a chain of fuzzy events that occur in observations [12, 92]. In temporal data sets, an activity is mapped as a chain of fuzzy events. These events represent real events that occurred in an environment. The resident partook in these in order to achieve a goal, which is a special world state. For example, when the activity of "coffee making" is performed, this results in eight fuzzy states (see Figure 3.2).

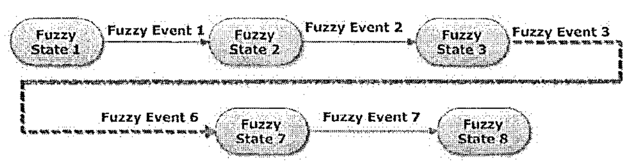


Figure3.2.: "Coffee-making" activity modeled as chain of fuzzy events

By modeling activities, we could estimate the intention of the individual and predict events that may occur in the future when a few elementary actions of a known plan or activity are seen [7]. The second viewpoint about the activity is to consider it as being a type of fuzzy-dynamic conceptual system. In fact, it is presumed that an intelligent system directs the realization of activity concepts in the virtual world of temporal data sets. In order to better explain this viewpoint, we refer to the system theory [95], in which a system is defined as a set of interrelated objects that collaborate together in order to achieve a goal; a system has a boundary with its environment; it takes input from its environment, processes it and gives output to its environment; Furthermore, it directs this output according to feedback from the environment. In a system, we can imagine machine states and a hierarchy of systems inside a system (subsystems) can be imagined. Here, the term "*conceptual system*" refers to a system that is composed of non-physical entities, i.e. ideas or concepts, a concept being an abstract idea or a mental symbol typically associated with a corresponding representation in language or semiology [96]. As a conclusion, let us say that a conceptual system is simply a conceptual model [95, 97].

An activity is a conceptual system because it respects the systems' specifications. It consists of a set of interrelated variables which represent world attributes and, especially, objects' locations. The activity is realized in order to achieve a goal, especially a world state; it has a boundary with its environment, which is defined through fuzzy state and fuzzy context; it takes input from its environment by performing observation, processing it and giving output to its environment by accomplishing an action in order to change a world attribute. For an activity we can define fuzzy states [92] and a hierarchy of concepts such as actions that can be imagined inside it [51]. Because this system (activity) depends on time, so an activity is a dynamic conceptual system. Moreover, because fuzzy mathematics are applied to model it, an activity is regarded as a sort of fuzzy dynamic conceptual system.

An activity is a sort of concept, so it is comprised of a set of rules that would justify the occurrence of a collection of observations. In fact, an activity concept draws on a logical virtual space in which an activity can be realized. In the case that more than one concept justifies a single collection of observations, then the one which finds the observation closer (having a greater degree of similarity) to the conceptual structure would be ranked as the most possible activity (among other possibilities). For each concept (activity), a structure with a space that the activity can be valid in there is defined. In other words, we formalize activities through the definition of their conceptual structures. The important point here is that it is desirable to formalize this concept in a data-driven manner without application of the expert idea.

3.2.1 System viewpoint on the Smart Home

Over the course of our research, we have observed the training of an intelligent system of activity recognition, anomaly recognition and assistance provision in normal world states. This training data resulted when the normal behavior of a human being in the Smart Home was observed. In doing this, much detailed information about activities could be included in the training data. In figure 3.3, a schema representing a systematic approach of activity recognition while using a data-mining strategy is presented. In this schema, one can see that the Smart Home acts as an intelligent system, while we find the general properties of a typical system in it; for example, a typical system should coordinate with its environment by taking input and giving output, and it should control the environment by receiving feedback [46, 95, 97, 98].

We selected the (intelligent) system viewpoint on the activity recognition, anomaly recognition and assistance provision problem because of a major reason which is that we can divide the Smart Home ambient environment into two major groups: that of the intern system and the extern system. The system collaborates with its environment by inputting data and outputting world actuation. Further explanation on this subject is that we consider the Smart Home as an activity observer and this data indicates the behavior of an intelligent system. "Normal world state" is the final goal of this system and it tries to achieve this final goal through world actuation. Therefore, the Smart Home is a big data warehouse that contains the intelligent system's behavior and, through the use of data-mining techniques, the behavior of this system can be modeled [12, 13]. It predicts future events that will happen and provides information about resident intention in order to assist him or her (rational agent).

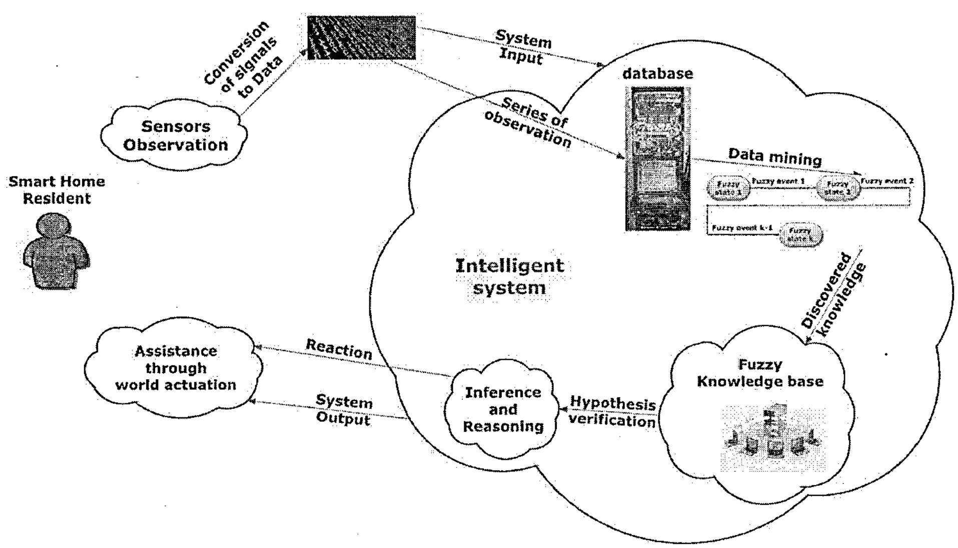


Figure3.3.: System viewpoint on activity recognition and assistance provision

In the next sections, we will discuss not only that a Smart Home is a system, but also that we can recognize a system viewpoint in the concept of activities so that we propose to consider an activity as a sort of conceptual system. In this way, fuzzy contexts and fuzzy states of activities would constitute the conceptual system's intern and extern. To do so, we begin by identifying fuzzy-context and fuzzy-state concepts, and then in figure 3.4 a general schema on this conceptual system is proposed.

3.2.2 Fuzzy contexts of the activities

Contexts are the surrounding conditions in which activities are realized [11]. The fuzzy context refers to a set of variables in which a stable interrelation is maintained while activities are realized. At recognition time, any change in context is interpreted as abnormality of world state. For example, if a human wakes up at 6 o'clock, then it indicates a normal world state for the activity recognition problem, but if he or she wakes up late at 2 p.m., then it can be inferred that he or she is sick and that an anomaly is present.

One of the other benefits of the consideration of contexts refers to help in the identification of similar activities. When similar activities are performed in different contexts, they represent different concepts and, in this way, we can distinguish these different concepts. For example, if a human eats food in the morning, it means that he or she is taking his "breakfast", but the same activity (eating) at 12 o'clock noon means that he or she is having "lunch". In the next section of this thesis, we will deal with the formalization of the fuzzy context.

In our point of view, context is a fuzzy term and it can be applied to multi-variable problems such as ambient environments, in which multiple features of scenarios are observed. In real-world problems, any sensor data may vary, even partially, while activities are realized and, sometimes, these changes should be taken into account because this variation could be significant, but sometimes they should not be taken into account because the variations of sensor data are not significant in recognizing the activity. We apply a fuzzy-logic based clustering approach in order to survey different levels of details of occurring events in different levels of certainty and survey the activity models in their own contexts.

3.2.3 Fuzzy states of activities

A fuzzy state represents a general and brief description about the current status of the world. When an activity is realized, the world observes the transit of a chain of fuzzy states (see Figure 3.2); however, this transition proceeds in a special fuzzy context. Each activity is regarded as a sequence of fuzzy states (see Figure 3.1). In fact, when an activity is performed, the world observes the transit of a chain of fuzzy states, and the system achieves its goal while activities are being realized. Considering that an intelligent system is assigned in order to direct the realization of an activity, the perception of fuzzy states and fuzzy contexts may indicate how to repeat the realization of this activity. In this system, the fuzzy context represents the environmental and external conditions required for realization of the activity, but fuzzy states represent the procedures or actions that should be performed by the system in order to realize the activity. In fact, fuzzy states represent the internal states of this system and the events that may occur inside it (see Figure 3.4).

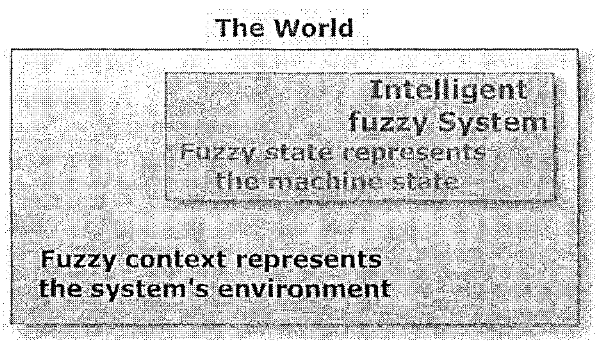


Figure3.4. Fuzzy state and fuzzy context required for the realization of an activity

In Figure 3.4, we can see that although the types of the fuzzy contexts and fuzzy states are similar, the objective of their consideration is different; in fact, one represents the conditions in which a scenario can be realized by an intelligent system, and the other represents the way that the scenario can be realized in the world. Therefore, it is presumed that in order to realize an activity, the world is divided into two sections which are the intern system and the extern system. The world features that should remain invariable during activity realization are the fuzzy-context members (extern system), and the attributes that are varied or would be used for activity realization; these would be taken into account as being fuzzy-state elements (intern system). The boundary between an intern and extern system is not a fixed, stable or definitive border. This logical boundary may dynamically change and new attributes join the system environment (fuzzy context) after a few steps of activity completion.

In order to relate Figure 3.4 to activity recognition, the *world* may be recognized as a temporal data set containing observations from an activity. Fuzzy states are known as the knowledge (pattern) extracted from some activities' scenarios, and the fuzzy context represents the conditions in which the learned knowledge is valid. Hence to explain system dynamicity which we distinguish from system border, we propose to consider fuzzy subtractive clustering in the definition of fuzzy states and fuzzy contexts. The reason is that this algorithm discovers clusters by considering both the cluster's insides and outsides. Therefore, discovered fuzzy clusters would define system fuzzy states respecting fuzzy contexts. In order to correctly illustrate this issue, we propose the following example. Let us consider the realization of an activity in the world through the observation of six variables, which are indicated in Table 3.1. Then, the fuzzy context and fuzzy states of this activity are sought out:

3.1.: Observation of Six World Attributes in Twenty Stages
(synthetic data)

observation number	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6
1	1	1	1	10	12	10
2	2	1	2	9	11	10
3	3	1	1	4	10	10
4	4	4	2	10	9	10
5	5	4	7	19	8	10
6	6	4	8	7	8	10
7	7	7	7	7	9	10
8	8	7	8	5	10	10
9	9	7	12	17	11	10
10	10	9	13	18	12	10
11	11	9	12	20	12	10
12	12	9	14	18	11	10
13	13	20	20	2	10	10
14	14	20	19	5	9	10
15	15	20	18	2	8	10
16	16	16	19	19	11	10
17	17	16	15	14	10	10
18	18	16	14	12	9	10
19	19	12	15	5	9	10
20	20	12	13	1	8	10

In order to break world observations into two groups of context and machine states, we apply the subtractive-clustering method [99] and then verify results. In Table 3.2, we have illustrated how the world is perceived. We have shown that by selecting different influence ranges (IR), different cluster sizes result; the bigger an IR is selected, the fewer number of fuzzy states are created. In contrast, a smaller IR would cause smaller cluster sizes and, thus, more fuzzy states would be perceived by the world. Each line represents the cluster center of a fuzzy cluster, and the symbol © is used in order to indicate which variable is taken as a fuzzy context member. Here, we see not only that the world is divided into two intern (fuzzy states) and extern (fuzzy context) sections, but also that applying the proposed data-mining activities model provided an approximate summary about the ongoing events in the world. By selecting a big IR, the world is seen as a single fuzzy context which shows no dynamicity; however, by selecting smaller amounts of IR, the world would be seen more dynamic, and more fuzzy states would be inferred. When a very small IR is selected (0.4), then the world is seen almost as being like the observations. Increasing or decreasing the IR, we can see different fuzzy states and, probably, different fuzzy contexts. This means that in requiring different details from temporal data sets, different perspectives for each set of fuzzy states and fuzzy contexts are inferred. As the result, the defined boundaries of activity states with their contexts depend on IR, so it may be said that this boundary is a dynamic boundary.

3.2.: Observation of Six World Attributes in Twenty Stages (synthetic data)

Fuzzy State Number	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6	Influence Range
1_1	8.0	7.0	8.0	5.0	10.0	10.0	IR=2
2_1	7.0	7.0	7.0	7.0	9.0	10.0	IR=1.5
2_2	17.0	16.0	15.0	14.0	10.0	10.0	IR=1.5
3_1	12.0	9.0	14.0	18.0	11.0	10.0	IR=1
3_2	7.0	7.0	7.0	7.0	9.0	10.0	IR=1
3_3	14.0	20.0	19.0	5.0	9.0	10.0	IR=1
3_4	2.0	1.0	2.0	9.0	11.0	10.0	IR=1
4_1	12.0	9.0	14.0	18.0	11.0	10.0	IR=0.9
4_2	7.0	7.0	7.0	7.0	9.0	10.0	IR=0.9
4_3	14.0	20.0	19.0	5.0	9.0	10.0	IR=0.9
4_4	2.0	1.0	2.0	9.0	11.0	10.0	IR=0.9
4_5	20.0	12.0	13.0	1.0	8.0	10.0	IR=0.9
5_1	12.0	9.0	14.0	18.0	11.0	10.0	IR=0.8
5_2	7.0	7.0	7.0	7.0	9.0	10.0	IR=0.8
5_3	14.0	20.0	19.0	5.0	9.0	10.0	IR=0.8
5_4	2.0	1.0	2.0	9.0	11.0	10.0	IR=0.8
5_5	20.0	12.0	13.0	1.0	8.0	10.0	IR=0.8
5_6	5.0	4.0	7.0	19.0	8.0	10.0	IR=0.8
6_1	10.0	9.0	13.0	18.0	12.0	10.0	IR=0.7
6_2	7.0	7.0	7.0	7.0	9.0	10.0	IR=0.7
6_3	14.0	20.0	19.0	5.0	9.0	10.0	IR=0.7
6_4	2.0	1.0	2.0	9.0	11.0	10.0	IR=0.7
6_5	17.0	16.0	15.0	14.0	10.0	10.0	IR=0.7
6_6	20.0	12.0	13.0	1.0	8.0	10.0	IR=0.7
6_7	5.0	4.0	7.0	19.0	8.0	10.0	IR=0.7
7_1	10.0	9.0	13.0	18.0	12.0	10.0	IR=0.5
7_2	7.0	7.0	7.0	7.0	9.0	10.0	IR=0.5
7_3	14.0	20.0	19.0	5.0	9.0	10.0	IR=0.5
7_4	2.0	1.0	2.0	9.0	11.0	10.0	IR=0.5
7_5	17.0	16.0	15.0	14.0	10.0	10.0	IR=0.5
7_6	20.0	12.0	13.0	1.0	8.0	10.0	IR=0.5
7_7	5.0	4.0	7.0	19.0	8.0	10.0	IR=0.5
7_8	16.0	16.0	19.0	19.0	11.0	10.0	IR=0.5
7_9	19.0	12.0	15.0	5.0	9.0	10.0	IR=0.5
7_10	3.0	1.0	1.0	4.0	10.0	10.0	IR=0.5
7_11	4.0	4.0	2.0	10.0	9.0	10.0	IR=0.5
8_1	10.0	9.0	13.0	18.0	12.0	10.0	IR=0.4
8_2	7.0	7.0	7.0	7.0	9.0	10.0	IR=0.4
8_3	2.0	1.0	2.0	9.0	11.0	10.0	IR=0.4
8_4	14.0	20.0	19.0	5.0	9.0	10.0	IR=0.4
8_5	17.0	16.0	15.0	14.0	10.0	10.0	IR=0.4
8_6	20.0	12.0	13.0	1.0	8.0	10.0	IR=0.4
8_7	5.0	4.0	7.0	19.0	8.0	10.0	IR=0.4
8_8	12.0	9.0	14.0	18.0	11.0	10.0	IR=0.4
8_9	16.0	16.0	19.0	19.0	11.0	10.0	IR=0.4
8_10	4.0	4.0	2.0	10.0	9.0	10.0	IR=0.4
8_11	19.0	12.0	15.0	5.0	9.0	10.0	IR=0.4
8_12	3.0	1.0	1.0	4.0	10.0	10.0	IR=0.4
8_13	13.0	20.0	20.0	2.0	10.0	10.0	IR=0.4
8_14	15.0	20.0	18.0	2.0	8.0	10.0	IR=0.4
8_15	1.0	1.0	1.0	10.0	12.0	10.0	IR=0.4
8_16	6.0	4.0	8.0	7.0	8.0	10.0	IR=0.4
8_17	16.0	16.0	14.0	12.0	9.0	10.0	IR=0.4
8_18	8.0	7.0	8.0	5.0	10.0	10.0	IR=0.4

We can see several hypotheses that explain the observations which are formed when different cluster sizes are desired. Each cluster is represented by a cluster center and the set of cluster center represent a fuzzy state. At the point of each fuzzy state, we can imagine that each variable is stable and, by occurrence of the fuzzy event (which may be a multidimensional event), the world transits to a new fuzzy state. For example, one hypothesis is that an activity transits into four fuzzy states, coded as 3-1, 3-2, 3-3, 3-4 in Table 2, and the sixth variable indicates the fuzzy context [11] of this activity, symbolized by © in Table 3.2. In figure 3.5, we have illustrated this perception observed from the presented world.

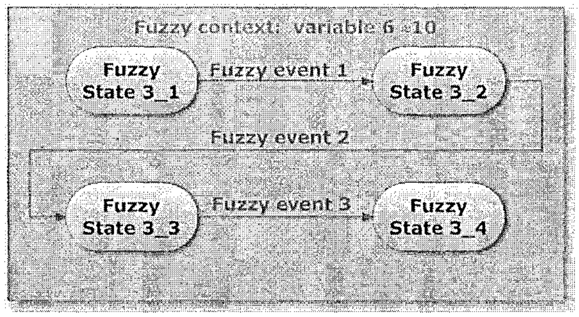


Figure3.5.: The world is modeled as chain of fuzzy events

In figure 3.5, the occurrence of fuzzy events illustrated will cause the transition from one fuzzy state to another in the world. A fuzzy event may be a multidimensional event; for example, when the first fuzzy event occurs, the first variable state decreases from approximately $12\ to\ 7$, the second variable from $about$ $9\ to\ 7$, the third variable decreases from $14\ to\ 7$, the fourth variable decreases from $18\ to\ 7$ and the fifth variable decreases from $11\ to\ 9$. This whole perception is valid only if the influence range has been selected as 1, and if the sixth variable stays approximately at the 10 value level [13]. In the section concerning concept viewpoint pertaining to activities, we will discuss that all of the discovered machine states can be crossed by a curve or line through a regression operation and the calculated equation will represent the activity function [8, 13].

3.2.4 Modeling of activities as a series of fuzzy events

Here, we introduce the proposed framework about the modeling process established in order to learn activities. The main function of activity modeling is to be able to make predictions in the Smart Home and reason about correct realization of activities. Moreover, the Smart Home should be able to judge if the world state is normal, or if the Smart Home resident needs assistance. Fuzzy conceptual structures are proposed to be applied in modeling of activities and, finally, a hierarchy of concepts may be drawn useful in demonstrating the system's perception of the world. Consequently, we introduce some definitions that are applied in our mining model. The world in the proposed learning problem is observed through a set of applied sensors. The symbol $s_{i=1,n} \in S$ represents the i^{th} sensor from the set of applied sensors S where n refers to the number of sensors or variables and a refers to a typical activity. Goal G is achieved when a is realized, so in most cases, we can imagine that goal achievement is equivalent to activity realization.

3.2.4.1 The world

The world in the activity recognition problem is the ambient environment in which activities are realized. It is desirable for a Smart Home to cover all features of activities through its embedded sensors. Therefore, when an activity is realized, all sensors or a subset of these will be actuated and the Smart Home state (in short, we call it the "home state") will change, so the home state is dependent on the activity state. For the Activity Recognition Reasoning System (ARRS), the closed world [100] of ambient intelligence is perceived through the observations taken from the sensors. For example, these sensor observations indicate the spatial locations of the objects applied in activity realization. Frequent observations of all sensors from the real world are recorded in a temporal data set and this temporal data set constitutes the elementary input of the ARRS. Therefore, the world is observed by sensors and they map the real world to virtual world. If the world contains any feature which remains unobserved, then the ARRS will not perceive it and, naturally, it does not take it into account; therefore, the perception of the real world is limited to observatory facilities, which equal the virtual world or temporal data set. One important property of this collection is that each of its members (each sensor) observes one property of a single reality at a particular moment. In other words, it is presumed that sensors perform their observations synchronously.

> **Definition 3.1** (*virtual world*). Presuming the reality is the state of the world attributes as they actually are, the virtual world is the collection of attributes that are observed from the world accompanying their observed values. Formally, the virtual world is a set of tuples $World = \{(s_i, v_{ti}, t) \in \mathrm{S} \times \mathcal{R} \oplus \{0,1\} \times T \ / \ |\mathrm{S}| = n, v = val(s_i, v_{ti}, t)\}$.

in which s_i observes the i^{th} attribute of the real world out of the observed features and we refer to s_i at time t by v_{ti}; however, T is the number of times that observation is done. The $val()$ function captures the value of sensor s_i at time $t \in [1, T]$. The observations are made through n sensors for T times. RFID sensors or, generally, any kind of sensors that generate any amount of values $v \in \mathcal{R}$ or the ones that generate 0-1 values are the data types that are accepted. Because of the fact that activities are realized and observed in an ambient environment, we expect the observed world to be unaffected by any event occurring out of the ambient environment. Moreover, we presume that all possible world states are observed within T observations. Therefore, in this thesis, the ARRS supposes that the world is closed or, in other words, it benefits from Closed World Assumption (CWA), which is a presumption that what is not currently known to be true is false. For example, if no explanation for an observation is found, then we infer that the world is abnormal or an erroneous activity has been realized.

3.2.4.2 Observation

Observation is the process of gaining significant details from world attributes in order to be analyzed by artificial-intelligence techniques and discover high-level information to explain the world. In other words, observation is the process of sampling the signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. When an activity is realized, world quality is changed significantly and the observation process informs the ARRS about these basic events. Technologically speaking, during the observation process world qualities are transformed into electrical signals through the application of sensors and these electrical signals are again transformed into digital numeric values, which are understandable for the computers. In total, observation includes three main steps which are performed by two sorts of hardware devices:

- Sensors that convert physical parameters of the world into electrical signals;
- Signal conditioning circuitry that converts sensor signals into a form that can be transformed into signal values. From a logical viewpoint, this hardware may be an Analogue to Digital converter (ADC).

In this thesis, we do not deal with the electrical conversion procedure which occurs inside the observation process. However, when we refer to raw data, we are in fact referring to the digital numeric values that are received from digitizers and registered by the database. In ambient environments, world observation is conducted frequently through a definitive frequency and world quality is estimated/measured for each observation and registered automatically inside a temporal data set. Each time the world is observed, a line of data record is created in order to save this world quality for future analyses.

Definition 3.2 (*momentum observation*). Let $O_t = (v_{t1}, v_{t2}, \ldots, v_{tn})$ the observations vector taken at time t on all sensors $s_i \in S$ of the Smart Home, where v_{ti} is the imprecise value of the sensor s_i at time t. The observations of an activity $a \in A$ concerning some goal G are represented by matrix $O_a = [v_{ti}]$, where $1 \leq t \leq T,\ 1 \leq i \leq n$.

$$O_a = \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{T1} & \cdots & v_{Tn} \end{bmatrix}$$

Therefore, each row or line of this matrix is an observation of momentum in the world and each column represents the observations of the concerning sensor during activities realization. This is to say that every attribute of an observation refers to a single reality or each attribute explains a property of a single object. An example of observation is the temporal data set, which is formed from frequent registration of the output of the Smart Home's sensors. For example, while the activity of "coffee making" was realized, the distance of the RFID tags attached on the sugar, glass and coffee to the two RFID antennas was captured as seen in Table 3.3:

Table 3.3.: Observation of six attributes of the world during the first five seconds of the "coffee making" activity in LIARA.

	Time	Sugar Vs. RFID Antenna 1	Sugar Vs. RFID Antenna 2	Glass Vs. RFID Antenna 1	Glass Vs. RFID Antenna 1	Coffee Vs. RFID Antenna 1	Coffee Vs. RFID Antenna 1
Observation 1	Second 1	0	0	0	37	34	0
Observation 2	Second 2	0	31	0	36	39	0
Observation 3	Second 3	0	34	44	0	33	0
Observation 4	Second 4	0	40	39	39	40	0
Observation 5	Second 5	0	43	0	0	35	0

In Table 3.3, we can see that time, like any other world attribute, is observed and taken into account as an activity feature that may provide useful information for activity recognition. In latter parts of this thesis, we will discuss the fact that multi-attribute observations are considered as being the input objects of the proposed activity-recognition reasoning system.

3.2.4.3 Sensor (variable) state

In previous sections, we discussed that sensors' observations are transformed into digital numerical values, so it can be inferred that sensors are mapped as variables in the ARRS. Furthermore, we discussed that sensors' generated values are dependent on activity realization. Therefore, by surveying these values, we may discover the patterns that a sensor (variable) might obey when activities are realized. Regarding time, sensors' generated digital numerical values can be formalized as a time series. Therefore, each column of Table 3.3 is a time series; however, all of these columns describe different aspects of only one reality (activity). During realization of activities, we can see sensors staying *temporary* or *permanently* at a definitive value. This stability at a definitive value causes consideration of sensor (variable) state. It is expected that variables should act in a feasible space, defined by variable states. In other words, it is expected that when an activity is realized, variables act within predicted states.

This value is given to a variable according to the measurement that a sensor takes from the concerning attribute. For example, in Table 3.3, we can see five states for each variable in each column. For instance, the coffee-container antenna is $\{34,\ 39,\ 33,\ 40\,,35\}$ centimetres away from the first RFID antenna. The important point here is that definition of a variable or sensor state is dependent on time, so by the elapsing of time, new data records are created and new variable states can be created. Considering the role of time, we face two groups of variable states. The first group of states refers to values in which the variable stops at several different moments and causes a relatively *stable* state; there are also *transition* states that indicate the transition of a variable from a stable state to other stable states. In order to distinguish these two - *transition* and *stable* states in a data-driven manner, we first consider the time factor. Consequently, when a variable stays at a definitive value for a *relatively long* time, this value is a sensor (variable) state and, if a variable remains for a *relatively short time* on a definitive value, then it is a transition state. When we talk about variable state in this thesis, we are in fact referring to a stable sensor state.

Definition 3.3 (*sensor state*). Let O_a an observation matrix on activity a, sensor state is formally defined as set $State_i = \{v_{ti} \in O_a : \exists\, \varepsilon \in [1, T],\ i \in [1, n] \rightarrow v_{t-\varepsilon i} = v_{ti} = v_{t+\varepsilon i}\}$ where i represents the i^{th} sensor in which ε is the minimum delay for stay of a value in order to be recognized as a stable state.

For each action that a Smart Home resident performs, in order to complete the realization of an activity, the state of one or more sensors may be actuated, so their digital-monitoring numeric value may be changed. Therefore, accomplishment of simple actions in the Smart Home is mapped as a series of events in temporal data sets. During realization of activities, we can see each sensor stay *temporarily* or *permanently* at definitive values. This stability causes consideration of a world state. World state represents a short estimation of world quality. Depending on the observed attributes, world quality is estimated. If more attributes in the world are observed, then a better estimation of world state is provided and the difference between similar world states is more easily distinguishable. In Table 3.3 for example, considering that the world is observed frequently, thus each record of gathered temporal data represents an instantaneous estimation of world quality. In other words, each record of the observation matrix represents world state momentum at the concerning moment.

The important point here is that definition of world state is dependent on time. Consequently, through passing of time, new data records are created and new world states may be created. Considering the role of time, we face two main groups of world states. The first group of states refers to the moments when entire variables stop in their old values and cause a relatively *stable* world state. There are also *transition* states that demonstrate the transition of a world state from a stable state to other stable states. In order to distinguish these two *transition* and *stable* states in a data-driven manner, we must consider the time factor. Consequently, when a world remains at a definitive set of values for a *relatively long* time, then it causes a world state and, if a world keeps a definitive set of values for a relatively short period of time, then it is a transition state. For example, we can view the world represented in Table 3.3 by this expression:

"The world state in which the distance of sugar to RFID antenna 2 is high and the distance of glass to the RFID antenna 2 is high."

In this statement, we pointed out the second and fourth record of the observation matrix. Therefore, world state is a record or a group of records from the observation matrix which are subjected to variable limitations.

3.2.4.4 Fuzzy cluster centers of observations

Performing fuzzy clustering on an observations' matrix and fuzzifying the world state concept would lead to the creation of a *fuzzy-state* concept. The most important characteristic of this task is that we are able to handle existing uncertainty in observations and make new classes to explain individual observations. In order to fuzzify temporal observations, we benefit from temporal interpretation performed by the fuzzy subtractive-clustering method, which was originally proposed by Stephen L. Chiu in 1994 [99]. This algorithm is an efficient method for extracting fuzzy-classification rules from high-dimensional data such as Smart Home temporal data sets. By this method, we can transform nonlinear input-output relationships to be expressed by a set of qualitative "*if-then*" rules [75]. One important feature of this method is that it allows knowledge extraction when the expert cannot express his or her knowledge explicitly. In the Smart Home, we hope to be completely fair about activities and, having no expert idea or prejudgment about the ongoing events in the Smart Home, we desire to recognize these activities. In doing this, we can be hopeful about proposing an automatic recognition method which does not rely on experts' help at running time.

The way that our method works is that at first, it quickly estimates the number of clusters and cluster centers from a collection of data points. Then, initial fuzzy rules with rough estimates of membership functions are obtained from the cluster centers; finally, membership functions and other rule parameters are then optimized regarding some output error criteria.

From a perspective of data points, a cluster is a set of similar objects where similarity is defined by some distance measurement. Clustering is a critical task for temporal data mining in the Smart Home because the similarity between objects changes temporally and there should be defined temporal and dynamic criteria for distance measuring. This dynamicity in similarity measure is dependent on two general reasons. One reason for this is that occurrence of events is dependent on realization of previously performed actions or activities. The other reason concerns context or the environmental conditions of a system. In some special contexts, occurrence of an event is more possible than in other contexts. Therefore, behavior of a system from an external observer viewpoint is dependent on both the system's external (environmental) and internal variables.

The proposed cluster-center estimation method estimates the clusters between data points according to both data points' internal relations and external criteria. Here, by the expression "external criteria", we refer to the state of data points from the perspective of a variable regarding other parameters that do not change under any circumstances. For example, when the distance of an object to a particular point[10] in the environment is measured, then the position of this particular point is considered an external parameter (absolute criteria) in data mining. But, if we can find two objects that have a similar distance to a particular point, then, from that particular viewpoint, it can be inferred that these objects are close to each other. This second observation reveals the relative criteria for discovery of internal relations between data points.

Some authors work with fuzzy-clustering methods in the product space of input-output space in order to detect interaction between input and output variables. Others have extended the use of fuzzy clustering to detect multidimensional fuzzy sets in the product space of the input variables to identify the premise of fuzzy rules and, then, assign a linear consequent to each rule. The identification of fuzzy models can be improved by using these multidimensional reference fuzzy sets [75]. Hence, fuzzy clusters give rise to local regression models. The idea of fuzzy clustering is to divide data space into fuzzy clusters, each of these representing one specific part of system behavior. After projecting clusters onto the input space, antecedent parts of fuzzy rules can be found. The consequent parts of these rules can then be simple functions. For example, in activity recognition, these fuzzy rules are expressed by linguistic rules like: *If object 1 is near to point1 then it might be fuzzy state1 of activity1*.

[10] In the proposed validation, it is the location of RFID antennas that observe RFID tags in the environment. Their position is fixed in the environment and it can be said that they have absolute positions in the environment.

Using a fuzzy-clustering algorithm, membership functions can be determined according to two possible methods. In the first method, clusters are projected orthogonally onto the axes of antecedent variables, and membership functions are fitted to these projections. The second method uses multidimensional antecedent membership functions, i.e. the fuzzy clusters are projected onto the input space [75]. In order to symbolize the aforementioned process, we use the *subclust* function.

Definition 3.4 (*fuzzy-cluster center*). A fuzzy-cluster center of set of sensors S on activity a is an observation vector $CC_{\bar{t}} = (\bar{v}_{\bar{t}1}, \bar{v}_{\bar{t}2}, \ldots., \bar{v}_{\bar{t}n})$ representing a group of observations $\{O_t: 1 \leq t \leq T\} \subseteq O_a$ that are similar to each other, where $\bar{t}$ is a fuzzy time and $\bar{v}_{\bar{t}i}$ is a center cluster on sensor $s_i \in S$.

Each $cc_{\bar{t}}$ is discovered through the clustering function *subclust()* by specifying the radius of range IR. We note the matrix of center clusters $[CC_{\bar{t}}] = substract(O_a, IR)$. In the temporal subtractive clustering process, the cluster centers are discovered based on two parameters:

- Similar observations of a single sensor are clustered and, for each cluster (containing similar data points), a cluster center is discovered in order to represent its concerned cluster members.
- Similar momentum observations are clustered and, for each cluster (containing similar observations), an n −dimensional cluster center is discovered in order to represent its concerning cluster members, where n is the number of observing sensors.

One important parameter that the proposed clustering method depends on is the cluster radius IR. A cluster radius parameter defines up to what distance a cluster center may represent its neighbors. If the data is normalized in the sense that data points are indicated in a range between 0 and 1, then in Table 3.4 we represent interpretations for the IR factor:

Table 3.4.: Selection of different cluster radiuses or influence ranges (IR)

Cluster Radius (r_a)	Radius size	Interpretation of the cluster center
IR = 1	Biggest possible	The data point, which is the most similar to all of the data points is the cluster center.
IR = 0.9	Very big clusters	A few data points which represent general dataset specifications are the cluster centers.
IR = 0.5	Fair size clusters	Data points that are at least half-similar to their neighbors are the cluster centers.
IR = 0.1	Very small clusters	Several data points that are very similar to their neighbors are the cluster centers.
IR = 0	No cluster	All the data points are the cluster centers.

In Table 3.4, we present a selection of a bigger IRs which would lead to consideration of bigger clusters. Thus, few cluster centers representing general characteristics of the temporal data set would be discovered (see Figure 3.6).

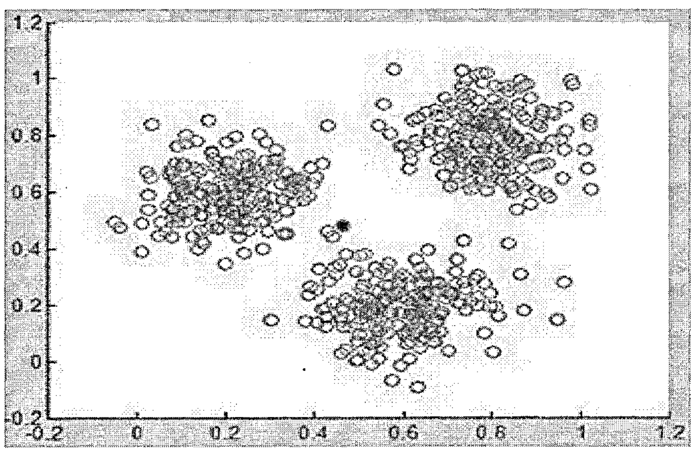


Figure3.6.: At high rates of influence range (here IR=1), the discovered cluster center would represent general characteristics of the data set

When a smaller IR is selected (here $IR = 0.5$) then more cluster centers that represent local characteristics of the data set are discovered. The discovered cluster centers are more similar to the cluster members (see Figure 3.7)

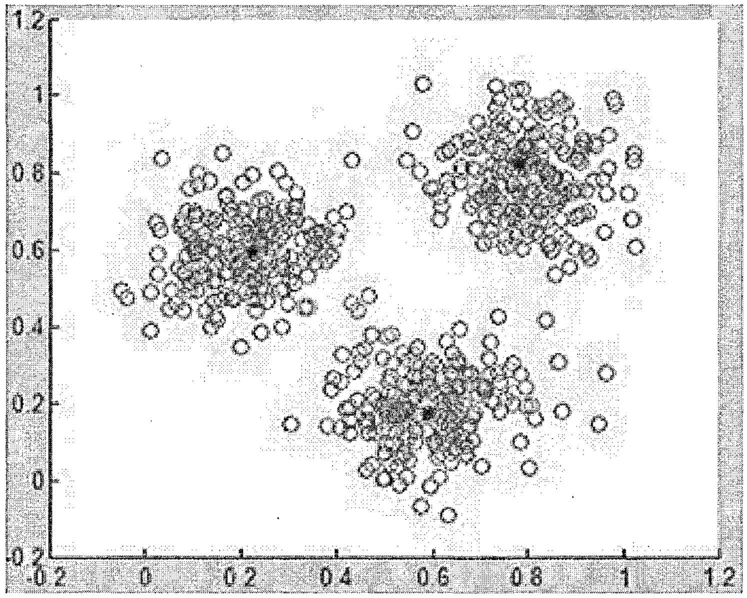


Figure3.7.L By selection of IR between 0 and 1 (here IR=0.5), discovered cluster centers would represent detailed characteristics of the data set

In Figure 3.7, we can see that the discovered cluster centers take on most similarities from their cluster members. This was illustrated by the selection of an $IR = 0.5$; the cluster centers here are fifty percent similar to their cluster members and accept a difference of up to 50% in order to accept a data point to be included in their clusters. By selection of smaller IR rates, less similarity degree is desired; therefore, more cluster centers are discovered. If we select the minimum IR rate, which is $IR = 0$, then every data point would be considered as a cluster center (see Figure 3.8).

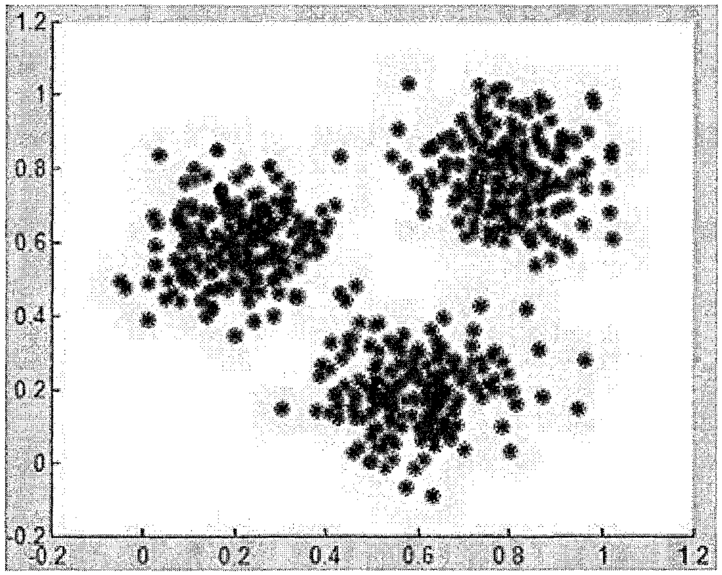


Figure3.8.: The less cluster radius is selected, the more clusters are defined. The maximum number of classes would correspond to the number of individuals.

In Figure 3.8 we can see that a maximum number of details are inferred from the observation when raw observations are taken into account. Finally, it can be concluded that the influence range factor defines how far one wishes the cluster centers be from each other. It can be inferred that the mentioned process is a one-pass quick algorithm and it needs no expert knowledge concerning the number of fuzzy states of variables or the structure of fuzzy classes. In this process, in order to generate such knowledge, each element of the observation matrix O_a is compared to other elements; the positions that data points are mostly concentrated around them are calculated as the fuzzy cluster centers.

3.2.4.5 Fuzzy sensor state

Fuzzy sensor state is the value in which a sensor is stable for a *relatively* long time. The quality of the term "relatively long" depends on two factors: firstly, it depends on the delays of this variable on its other values; secondly, it is contingent on the delays of other variables stable on their values. These criteria are already considerable in the *subtract()* function, when time is observed and inputted like other variables. Therefore, the data-driven interpretation of fuzzy time can explain the criteria for discovery of sensor states.

Definition 3.5 (*fuzzy sensor state*). Let $O_{s_i} = \{v_{ti} \in O_a, 1 \le t \le T\}$ a set of the values of the sensor $s_i \in S$, a fuzzy state of sensor s_i, noted $\bar{s}_{i\bar{t}}^{k} = (\bar{v}_{\bar{t}i}, \bar{t}) \in O_{s_i} \times [1, T]$, is a $k^{th} \in \mathbb{N}$ group of similar observations on s_i related to fuzzy time $\bar{t}$, such that it exist a cluster center $\bar{v}_{\bar{t}i} \in subclust(O_{s_i}, IR)$, where $IR \in [0,1]$.

IR is the desired detail/generality from data points where the set of cluster centers ordered by fuzzy time $\bar{t}$. IR is the influence range or the cluster radius rate, which defines the clusters' sizes; therefore, cluster centers would represent similar data points in which the similarity criterion is the cluster radius. Influence range factor is a relative factor and it depends on the range in which data points are. An important point to note is that selection of different influence ranges in cluster center estimation would lead to different interpretations from sensor observations; accordingly, different data points with different quantities of cluster centers are proposed as sensor states. Therefore, for each temporal data set, different sets of data records representing the total data set may be discovered. In the next part of this thesis, we will discuss combinations of fuzzy sensors' states which will lead to the creation of fuzzy-world states.

3.2.4.6 Fuzzy world state

A fuzzy world state is a set of attribute values in which the world is stable for a *relatively* long time. The fuzzy time concept is applied in order to discover fuzzy world states. In this way, when the world remains in a state for a *long* time (which is a fuzzy term), then it will be recognized as a fuzzy world state; however, the world state itself is a fuzzy state. In other words, the sensors' values are *relatively* and *approximately* stable while they are in a fuzzy world state.

Occurrence of a fuzzy event which represents the occurrence of a real event in the world may lead to the transition of a world from one state to another.

Definition 3.6 (*fuzzy world state*). Let $O_a(T, n) = [v_{ti}]$ the observation matrix of an activity a where n is the number of the sensors and T the learning time, let $IR \in [0.1]$ the fixed influence range, the fuzzy world state, noted $FS_{a,k}^{IR} = (\bar{v}_{\bar{t}1}, \bar{v}_{\bar{t}2}, \dots, \bar{v}_{\bar{t}n}, \bar{t}) \in O_{s_i}^{n} \times [1, T]$, is the $k^{th} \in \mathbb{N}$ group of similar observations on n sensors related to fuzzy time $\bar{t}$, such that it exist a cluster center vector $(\bar{v}_{\bar{t}1}, \bar{v}_{\bar{t}2}, \dots, \bar{v}_{\bar{t}n},) \in subclust(\mathrm{O_a}, IR)$:

A fuzzy world state may include one or more rows of matrix O_a. For example, on the data of Table 3.1, we are able to apply the fuzzy clustering process on data points in order to extract the points (cluster centers) around which the data is concentrated. Thus, they represent different existing qualities of these data points, which are similar to the majority of data points. The result of this process is shown in Table 3.2. It can be observed that if the cluster radius is selected as $IR = 0.7$, then the world would be divided into seven fuzzy states. If at running time a relatively high similarity between current observations and learned fuzzy cluster centers is observed, then it can be inferred that the observations might belong to the realization of a surveyed activity or fuzzy activity.

3.2.4.7 Fuzzy activity

The fuzzy activity refers to a set of fuzzy world states at the influence range of IR, that are transited in order to realize activity a. It demonstrates the realization of an activity in transition of fuzzy states through observation of the i^{th} observer sensors at the influence range of IR. To make this concept, the observed data of activity realization O_a is fuzzified at an influence range of IR.

Definition 3.7 (*fuzzy activity*). Let a set of m fuzzy states $FS_{a,k}^{IR}$ and IR the fixed influence range, a fuzzy activity, noted FA_{IR}, is defined as set of transition on these fuzzy states, $FA_{IR} = \{FS_{a,k}^{IR} \in subclust(O_a.IR) : \forall k, IR \in [0,1],\ \exists fuzzyEvent(FS_{a,k}^{IR}, FS_{a,k+1}^{IR})\}$ through clustering the matrix observation of the habitat, where each couple of fuzzy state are related by fuzzy event transition.

in which a is the mined activity; IR refers to the considered range of influence or the relative similarity degree and m refers to the number of fuzzy states that the world is transited in realization of a, so we have: $0 \le k \le m \le T$ and $FS_{a,k}^{IR} \in FA_{IR}$. In order to calculate the FA_{IR} in matrix format, we perform the temporal subtractive clustering on the observations matrix, noted $[FA_{IR}]$:

$$[FA_{IR}] = subsclust(O_a, IR) = \begin{bmatrix} \tilde{t_1} & cc_{1,1} \cdots & \\ \ldots. & \ddots & \vdots \\ \widetilde{t_m} & \cdots & cc_{m,n} \end{bmatrix}$$

This matrix contains the cluster centers of all sensors (including time) resulting from a comparison process between data points. CC_{ki} represents the kth cluster of the ith variable (sensor). Each data row in this matrix represents a fuzzy world state. If n is the number of columns (variables or sensors), then it can be inferred that activity a would m times change the world state to achieve goal G.

3.2.4.8 Fuzzy event

Accomplishment of simple actions or operations in the Smart Home would lead to switch the world state from a one to another in the virtual world. A *fuzzy event*, which refers to the transition between two world states maps actions accomplished in the real world to the inferable significant events occurring in data sets. Therefore, recognition of fuzzy events would lead to inference of actions that may be a part of a greater plan such as an activity.

Definition 3.8 (*fuzzy event*). Let $\{FS_{a,k}^{IR} \in FA_{IR}\}$ a set of fuzzy world states, a fuzzy event is a couple of fuzzy world states $(FS_{a,k}^{IR}, FS_{a,k+1}^{IR}) \in FA_{IR} \times FA_{IR}$ ordered by a fuzzy time.

Fuzzy events are the constituting elements of a typical activity, if activities are defined as a sequence of actions. In this way, each fuzzy event represents an action in the temporal data set and activities are modeled as a series of fuzzy events [7]. A main point in order to perceive the concept of the fuzzy event refers to its relation to real world observable events that occur in the real world. The sensors of ambient environments capture the events that may occur in the real world such as standing on a carpet, opening a door or turning a light on, but on one hand, there are many non-physical events that cannot be captured and, on the other hand, there are many non-physical events that need a preprocess to be detected. For the inference system, a fuzzy event is a proper tool to imagine such events as well as real events. For example, detection of "delay between events" is itself a sort of event on which we should perform some preprocess in order to detect it. Thus, when we put an object on a table, some preprocess on the RFID data is needed in order to recognize this event. Therefore, although observable events may lead to inference of fuzzy events by the mining model, there are sometimes cases in which a fuzzy event is inferred even when an unobservable event has occurred. Through this description, we can say that a fuzzy event provides a framework for perception of the system behavior.

3.2.5 Activity Realization Pattern

In order to model activities as a series of ordered actions, we calculate their representing fuzzy events and make a set of the mentioned entities that are ordered by time.

Definition 3.9: (*activity realization pattern*). Activity realization pattern (ARP) is defined as the set of couples constituting from fuzzy events and their occurrence order $ARP_{a,IR} = \{(FS_{a,k}^{IR}, FS_{a,k+1}^{IR}) \in FA_{IR} \times FA_{IR} : \exists\, t \in [1,T] \rightarrow t - \varepsilon \leq k \leq t + \varepsilon\}$.

ARP simply represents the chain of events that may occur in the world when an activity is realized. These events may be caused directly by the activity performer, or they may occur as consequence of the accomplishment of actions. ARP represents a possible correct realization of a typical activity. An important point about interpretation of ARPs [7, 12] concerns the uncertainty in observation of the events in both learning and recognition steps. This question may arise in cases in which the same events observed at the learning phase are not observed during the recognition or running step. An example to this effect is that in some cases, most of the events may be observed in the learning phase, but during the recognition step, some of the expected events may not be observed by sensors. Conversely, it is possible that at recognition phase more events are observed rather than what was observed during the learning step.

In order to resolve the aforementioned problem, we should consider the ARP that would be made for live observations at recognition time. This ARP would be compared to the previously learned ARPs and the similarity measures between these two models calculated in order to detect if the activity was correctly realized [7, 12]. Therefore, if the measured similarities are high, it could be inferred that live observations obey a known normal model (pattern); in contrast, high dissimilarity would result in inference of an abnormal state or a different activity pattern.

3.2.5.1 Activity Prediction Pattern

One important objective of ARRS design is to predict the intention of the Smart Home resident when he or she accomplishes a few actions concerning a bigger task. In other words, it is desirable that, by observation of a few actions, the ongoing activity be recognized, so we can verify which services and assistance can be provided. Moreover, anomalies can be detected if none of the expected actions are accomplished. Activity prediction patterns (APP) refer to the models that at first indicate the activity that the resident intends to realize and, secondly, indicate possible events that may occur in future [7, 12]. In order to calculate the APP, all of the ARPs are compared together and, according to the entropy of their constituting elements, fuzzy events are ordered and put in a decision tree, of which its leaves are intended activities and its nodes are fuzzy events [12]. The result of the process of fuzzy event classification based on the information entropy factor is put in a decision tree which includes the following steps:

Input:

- E set of fuzzy events

Output:

- Decision tree

1: Let $e \in E$ as fuzzy event, k as metadata and a as target.
2: Calculate the information gain of all $e \in E$.
3: Put the fuzzy event $e \in E$ which has the highest information gain as the new root.
4: Calculate the entropy of all event qualities (increase/decrease).
5: Put event qualities as branches originating from the root from the highest entropy on the left to the lowest entropy on the right.
6: Write the event order in a list of metadata next to the branches of the concerning event.
7: If there is no more event $\in E$, then write the concerning activity a as the terminal.
8: Regardless of the current node, repeat the steps from 3 onward.

Algorithm 3.1.: Activity pattern recognition

The result of the mentioned process is a decision tree through which we can distinguish what activity is possibly going on and at what step of the process we find ourselves. When possible explanatory scenarios are discovered, we can refer to their concerning ARPs and make predictions. Therefore, the APP is applied for recognition objectives and it may tell us in which possible world states we are. If the tree is sorted based on the events order k, then it can be applied directly for prediction objectives. In algorithm 3.2, we can see how the ARP would be sorted by k.

Input:

- E set of fuzzy events

Output:

- Decision tree

1: Let $e \in E$ as fuzzy event, k as metadata and a as target.
2: Calculate the information gain of all $e \in E$.
3: Put the fuzzy event $e \in E$ which has the highest information gain as the new root.
4: Calculate the entropy of all event qualities (increase/decrease).
5: If $First\ Order(e) \in aNextRoot(e)$, then continue.
6: $Remove(e)$ temporarily from the calculations and put it in a new list.
7: Put fuzzy event $e \in E$, as being the new root.
8: Calculate the entropy of all $e \in E$ qualities (increase/decrease).
9: Put event qualities as the branches originating from the root from highest entropy on the left to lowest entropy on the right.
10: If there is no more event $e \in E$, then write the concerning activity a as the terminal.
11: Return the temporary list items to the main list.
12: Regardless of the current node or event $e \in E$, repeat steps from 3.

Algorithm 3.2.: Activity realization pattern

The activity prediction pattern expresses the most important parameters and values for recognition of activities. In fact, its consisting elements are the fuzzy events that are ordered based on their information entropy being accomplished in ARPs of all activities. Each ARP may be a hypothesis to the fact that its live observations obey its pattern. In fact, the live observations are interpreted as fuzzy events that are a part of an ARP. In this way, APP is the collection of all ARPs (hypotheses) that are ordered (indexed) based on information entropy of their concerning fuzzy events. ARP can be created during a classification process and represented by a decision tree [101]. In here, we consider information entropy as the classification criterion necessary in order to make a short and brief knowledge for recognition objectives. This prediction pattern, which is inferred from the decision patterns, represents the most important signs and information required to recognize the intention of activity realization. The fuzzy event that has the greatest entropy would be placed in the highest node of the decision tree and the lower nodes of the tree would have lesser importance to recognize the intended activity. By accomplishment of simple actions (operations), different activities (scenarios) would be realized. In fact, actions are subsets of activities and one action concerning a special activity can be found in the realization of other activities too.

3.2.5.2 Fuzzy context

As it was mentioned earlier, there are several variables that are analyzed in order to recognize activities. These variables in data sets represent the sensors' observations. Out of all observing sensors, only a limited number of variables play significant roles in recognition of activities and, in our experience, most of the variables do not play a noticeable role in both the realization and recognition of activities. For example, when the Smart Home resident performs the activity of "cooking", the sensors located in the bathroom do not play any role in both realization and recognition of the "cooking" activity.

In order to concentrate efficiently on role-playing variables for the recognition objectives, we propose to apply the fuzzy concept [11]. According to this idea, world variables are divided into two general groups, which are fuzzy context variables and role-playing variables. Fuzzy context variables are the ones which do not vary significantly and the role-playing variables vary significantly while the activity is realized. The concept of fuzzy time is applied here once again in order to distinguish noticeable changes in variables.

Definition 3.12 (*fuzzy context*). A fuzzy context, noted $\tilde{C}_{a,IR}$, is the set of sensors $\{(s_i, \tilde{v}_i) \in S \times O_a: \forall t \in [1, T] \rightarrow v_{ti} = \tilde{v}_i\}$ that do not play any significant role in both realization and recognition of activity a.

in which the sensor s_i during the time of the activity a realization does not vary significantly and it is fixed to value $\tilde{v}_i$ and this value is calculated through the cluster center discovery process. Fuzzy context indicates the surrounding circumstances in which scenarios or activities are realized. The fuzzy contexts of activities indicate the conditions and presumptions in which activities' models are valid; a change in fuzzy context may cause invalidity of the system's perception of the activities; thus, it will be taken into account as a new activity model. Therefore, any knowledge extracted from observations is valid only if a similar context is met.

Up to here, we presumed that activities constituting series of actions were transformed into fuzzy events in data sets, which are directed by the Smart Home resident in order to achieve a goal. This viewpoint describes the activity as a temporal entity and finds recognition criteria in the quality of world dynamicity. This approach has already been published in [7, 12]. In the next part of this thesis, we will discuss that although this modeling process is applied for recognition of correct realization of activities, it can nonetheless be improved to solve the following problems:

- A one-time activity may be modeled through different interpretations for the fuzzy event, especially when different IR values are selected, so that each activity may be viewed in a perspective of different granularity in the sense of generality/details of data-mining. In other words, neither definitive structure nor clear definition of the activities is proposed; however, ARP is just one possible pattern for activities;

- The beginning and ending points of activities should be cleared during the training phase;
- If more than one activity is realized (simultaneous activities) it does not recognize these correct activities as normal world states. Moreover, interruption of activities cannot be surveyed;
- Unseen world states cannot be predicted. In other words, the *trends* of world states cannot be predicted and predictions can target the learned world states. Therefore, no inference process (creativity) is conducted to foresee the future;
- Reasoning in recognition of activities can happen only when an action is performed in the world, so it does not reason in normality of the current momentum observations. Therefore, we desire the ability to conduct real-time reasoning.

The aforementioned items are the main motivation for the proposal of a new complementary viewpoint on activities in order to consider these as fuzzy-temporal concepts. This new viewpoint stands on the previously introduced event-driven viewpoint and extends it in order to more functional but resumed knowledge from the temporal data sets of Smart Home.

3.3 Modeling of the activity as a fuzzy-temporal concept

As it was mentioned earlier, the principal objective of observing intelligent system behavior in this part of thesis is to model the intelligence behind world actuations; inferred fuzzy events are applied as the forming entities of this model. Therefore, in this part of our text, we use and extend previously presented definitions and formalizations in order to broaden event-driven perception from activities and propose a fuzzy conceptual structure (FCS) for the activities. Through this viewpoint, we will regard activities as fuzzy temporal concepts in which fuzzy events are the constituting individuals of this concept and for which a structure is formalized that can describe both dynamic and static features of activities. Fuzzy conceptual structures are proposed to be applied in modeling of activities and, finally, a hierarchy of concepts demonstrating the system's perception of the world may be drawn. We will consider two assumptions so as to make the proper inferences. The first assumption is that an activity is a temporal concept as well as any other temporal concepts such as plans, behaviors or simple actions that may be realized in order to achieve a goal. The second assumption is that only one source of intelligence directs the occurrence of events in the environment; therefore, even if two or more people perform one, two or more activities in the Smart Home, then we will consider one universal comprehensive source of intelligence that directs occurrences of events.

Before we present modeling of activities as fuzzy temporal concepts, we will discuss the underlying reasons for this selection and the significance of a typical fuzzy temporal concept: In [92], we discussed the fact that ARRS is an intelligent system and, typically, an intelligent system needs to be fed knowledge in order to perform its recognition-related tasks. One way to discover and create knowledge for intelligent systems is to make concepts. In other words, an intelligent system perceives the real world by creating concepts that may explain occurring events [93]. If the concepts made are inferred from the analysis of temporal data, we then refer to the concept by the term "temporal concept".

The real world is perceived through application of concepts. In 1984, Novak [102] defined the "concept" as a perceived regularity or pattern in events or objects, or records of events or objects, designated by a label[11]. A concept is indicated through its properties and regarding its relations to other concepts that have common properties. It is impossible to characterize any concept without its relation to other concepts [93, 102, 103]; for example, in description logic, applying a semantic network the concepts are correlated. In the current thesis, as the knowledge is resulted from analyze of temporal data, the perceived concepts are named temporal concepts. Furthermore, in this work, concepts are represented in a hierarchy and each concept is indicated with respect to its higher and lower-level concepts. In fact, the extracted knowledge is represented by these concepts, and more specifically by fuzzy conceptual structures. In the current thesis, in profiting from fuzzy-event formalization, ARRS could perceive the world as a sequence of fuzzy events, which are the forming entities of temporal concepts.

For each discovered concept, we will consider two items which are *conceptual structure* and *flexibility space:* these are required in order to formalize a concept. Thus, any individual who may be subsumed by a concept should be confirmed by the concept structure while respecting the defined flexibility space of it. Conceptual structures [93] represent the cognition function of the world's ARRS system which perceives the environment and its occurring events. In other words, it represents the ARRS's perception of reality[12]. Around a single reality, several hypotheses can be made, or different concepts may be inferred, because this reality can be interpreted according to different criteria and methods. In this context, conceptual structure

---

[11] Label refers here to the "name" or the "word" with which a concept is called.

[12] Reality is the state of the world that actually exists, rather than as it may appear or might be imagined or perceived 102. Joseph D. Novak, D.B.G., *Learning How to Learn.* 1984: Cambridge University Press. 199 .

indicates totalitarian cognition of reality. For example, in Table 3.2 we can see that around a single reality (observations mentioned in Table 3.1), several hypotheses are made to explain it. Thus, we need a summarizer that encapsulates hypotheses around an activity's observations so as to make this activity structure. In the next part of this thesis, we will formalize recently proposed concepts.

3.3.1 Fuzzy temporal concept

Regardless of concept structure, a concept itself is an entity. Any conclusion from the temporal observation itself is a temporal concept because we use fuzzy theory in the process of concept discovery. Then, we call the discovered concept a *fuzzy temporal concept.*

> **Proposition 1** (*Fuzzy temporal concept*). Let set of m fuzzy states $FS_{a,k}^{IR}$ and IR the fixed influence range, a fuzzy temporal concept, noted $FTC_{IR=\epsilon}^{IR=\lambda}$ is a subset of fuzzy states $FS_{a,k}^{IR}$ that are discovered in a range of influence IR $\in [\varepsilon, \lambda]$. Geometrically, $FTC = \int_{IR=\epsilon}^{IR=\lambda} FS \, dIR$ is an integral which describe a fuzzy space of the set of fuzzy states points.

In this proposition, λ indicates high value of the influence range (IR) and ε represents low value of IR. In a normalized data set, λ=1 and it returns a single data point as a result. Also, ε represents the level of desired detail to be considered in modeling. In a normalized data set, $\varepsilon = 0.5$ would represent the fact that data points inside a cluster are at least fifty percent regarding total data points, and similar to the cluster centers. Therefore, fifty percent of dissimilarity to cluster centers is tolerated. Hence, high and low boundaries of IR represent the range of desired detail required in order to be included in the concept. A key point here is that the step lengths of IR count. If we could count the steps with the smallest possible ones then we would have: $FTC_{IR=\epsilon}^{IR=\lambda} = \int_{IR=\epsilon}^{IR=\lambda} FS_{a,k}^{IR} \, dIR$ in which FTC is the collection constituting of all possible cluster centers discoverable from a temporal data set. Technically, we are not able to load continuous values of IR into a clustering algorithm; however, we can load discrete values of IR. For example, at implementation time, we can select $dIR = 0.1$ as the step length for calculation of hypotheses which explain activities. One result that is caused by inference from *Proposition 1* refers to the hierarchy of concepts in the Smart Home. Presuming that we have observed all of the possible events in the world and then, by mining the observation data, we may infer a concept hierarchy from these observations in which the desired level of events defines the characteristics of FTCs. For example, if we look for highly detailed information from the observations, then we would select a very small value for ε. The result would be a set of observations because all data points would be selected as cluster centers; thus, every point in itself represents a concept quality. By increasing the value of ε from lower values to higher ones, we can see that actions are inferable and, by continuing the decrease of desired detail level, we are able to see higher-level concepts such as activities, plans or strategies that can be characterized. In Figure 3.9, we have shown a typical schema of concept hierarchy that may be discovered from the Smart Home data warehouse when different levels of detail in extracted knowledge are desired. Naturally, the quantity of discovered cluster centers would decrease when higher values for ε are selected because a smaller surface would be affected by the integral. This is why we expect that the hierarchy of discoverable concepts is presentable in a pyramidal form.

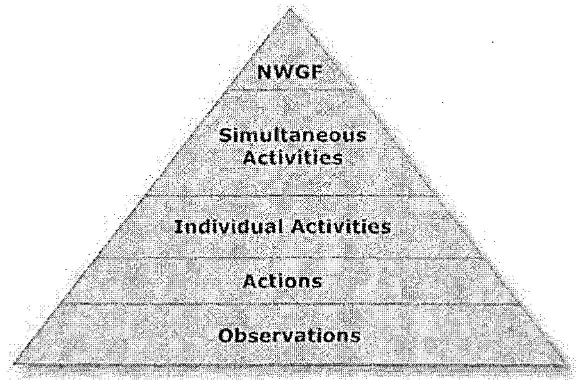


Figure3.9.: Hierarchy of concepts inferable from temporal data in the Smart Home

In Figure 3.9, we can see that the highest level of knowledge is called normal world generic function (NWGF), which describes the general characteristics of a normal world. In order for a behavior or strategy to be normal, it needs the NWGF to confirm it and accept it as a collection of normal world states as well as a typical activity, which needs to be confirmed by its higher-level concepts. Figure 3.9 specifies a hierarchy of concepts that a human might imagine for ADLs in the Smart Home. In this hierarchy, each higher-level concept is more generic than its lower-level concept. In description logic [104], when a more generic concept includes a more particular concept (like the generality relation between "female" and "mother" in the rule: "mother is female"), then it is said that the more generic concept subsumes the more particular concept. There are several concepts such as actions and activities that are situated between raw observations and NWGF. We call these middle-level concepts (MLCs). In the hierarchy of concepts concerning the Smart Home, we presume that each higher-level concept subsumes its lower-level concept. Except for the unsupervised approach, MLCs can be formed in a data-driven manner by performing a fuzzy-clustering operation on observations. To carry out this task, we mine observations from the world in various IR rates. Further explanation is when a very big IR value (λ) is selected; then, total observations would be transformed into a single fuzzy state, which indicates one cluster center per each variable. Then, by decreasing the IR value, these cluster centers would be broken into two or more cluster centers, which would cause consideration of two or more fuzzy states for observation and demonstrate details pertaining to the dynamicity of the concepts. In consequence, by decreasing the IR value, the cluster centers of each fuzzy state would be broken into more partial states so that more details would be mined. This process can be continued down to the step in which desired details about the world are obtained (see Table 3.2). If the desired details about activity are obtained at $IR = \varepsilon$, then in Figure 3.10 we can see how the concept of a typical activity could be inferred in a data-driven manner. This figure summarizes a data-driven approach for discovery of the concepts that a human mind may imagine.

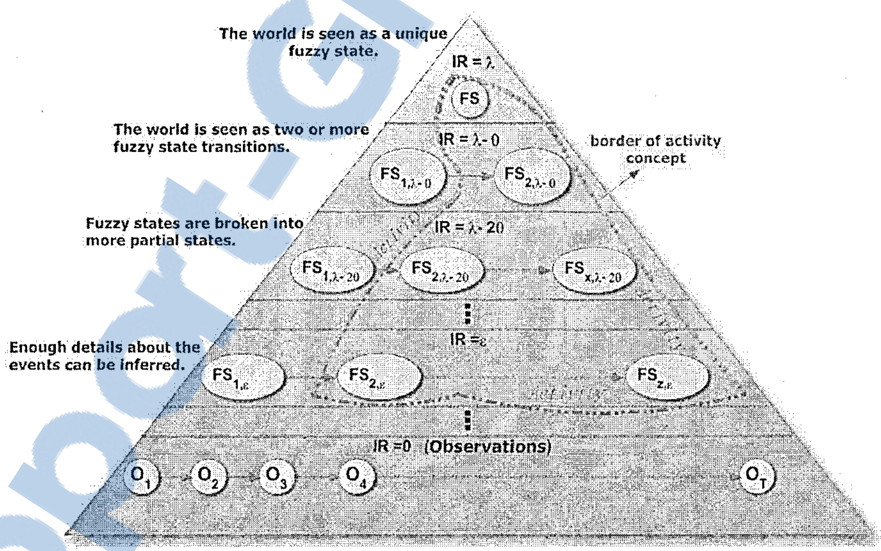


Figure3.10. Data-driven hypotheses for perception of concepts such as an activity

In Figure 3.10, it can be seen that at the lowest level of concept hierarchy, there are raw observations that describe world transition from one observation to another. By discovery of FTCs, we can achieve higher-level viewpoints describing world transitions. We can observe that a typical activity at $IR = \varepsilon$ can be modeled; however, if we select higher values for IR, then higher-level concepts such as plans, strategies and, at the highest level, NWGF could be discovered, too. In this figure, we affirm that a typical activity is a part of a normal world and other high-level MLCs. Hence, it can be said that in order to verify if an individual (object or data point) is an instance of a special concept, we would need to know the general characteristics of this concept at a glance. In other words, we need a model which briefly describes the possible qualities of conceptual individuals. Discovered similarity is once again the criterion for both concept structuring and concept recognition. FTC represents the set of important data points which represent the concept's general characteristics. As it was mentioned earlier, activity is itself a temporal concept and FTC represents the fuzzy conceptual structure of the activity. During recognition phase, by selection of the ε value, we can configure the ARRS and how much deviation from activity models are expected in activity realization. Hence, a key point concerns the activities concept and the way that we can recognize their borders, as shown in the last figure with the dotted line, in a data-driven manner from a huge temporal data warehouse.

3.3.1.1 Concept border recognition

In a huge temporal data set, which includes data about ADLs gathered by continuous observation of these activities over a relatively long period of time, the borders of activities should be known or labeled in order to automatically model activities. Through application of clustering, we could collect similar data points and, also, represent the activity as a single world state at a definitive influence range. Therefore, each activity constitutes a special temporal quality out of all of the observed world temporal qualities. This singular world state is a part of NWGF. In consequence, although activities from a bottom-up view are discovered from observations' analysis, they should be subsumed by higher-level concepts such as NWGF. If the influence range (in cluster estimation) on the total data set is within a range from zero to one [0-

1], then at a special cluster radius range, let us take for example $IR = q \in [0,1]$ in which q is proposed by an expert or it may be mined, we may find cluster centers that each represent a special world quality or an activity. Therefore, data-driven interpretation of activity a is every cluster center discovered at $IR = q$ which is the desired similarity rate between observations in order to be recognized as fuzzy-activity function.

3.3.1.2 Activity function

As it was mentioned earlier, one of our contributions in this thesis is to propose a mathematical multivariable function in order to model an activity and to recognize it. The goal of this function is to recognize activities and, to do that, it transfers observations to the modeling space, which is the activity space. This function is noted by y_a, in which index a refers to the surveyed activity. For instance, if we consider the positions of the "glass" and "sugar" objects in realization of the "coffee making" activity, then the $y_{coffee\ making}$ transfers the observations of the concerning sensors to the activity space in order to verify how similar it is to the coffee-making activity (see Figure 3.11).

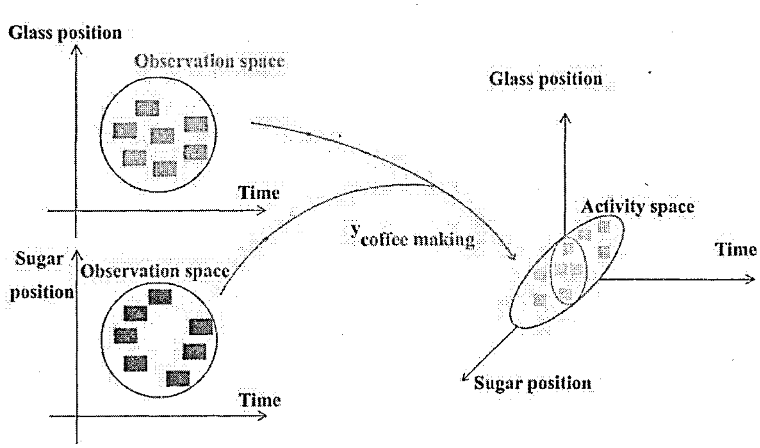


Figure 3.11.: Activity function

The way that this function works is based on discovered similarities between observations and activity structure. Therefore, it is expected that this function should reason in both momentum observations and series of observations that recognize ongoing activity. In order to discover this function, we calculate the equivalent of a curve or line that cuts across activities' cluster centers. When an observation is recognized as being similar to at least one of the activity's fuzzy states, then it can be inferred that it may possibly justify realization of the activity. The sensors are the variables that activity function depends on and, based on their generated numerical values, we can model or recognize activities. Presuming that we intend to calculate a line that goes across the cluster centers of the activity structure (FTC), then a resultant line is formalized as a combination of the values of the sensors involved in the activity.

Proposition 2 (fuzzy activity function). A fuzzy temporal concept $FTC_{IR=\varepsilon}^{IR=\lambda}$ describes a fuzzy activity function a iff. $\exists q \in [\varepsilon, \lambda]$, such that y_a is a multiple regression of the role playing sensors, formulated as $y_a = \alpha_0 + \alpha_1 v_{t1} + \cdots . + \alpha_n v_{tn}$

where v_{ti} is the value of the i^{th}sensor at time t in which α_i indicates the factor of the role played by the i^{th}sensor in activity a, obtained by regression or smoothing methods [64, 70]. y_a is interpreted as a similarity value that a typical realization of activity a would cause possibly which world states. In other words, it indicates the space in which observations of a can be valid. The logic in this reasoning is the estimated inferred similarity degree between observations v_{ti} and activity function y_a. In order to calculate $y_{a,}$ at first we must estimate the line or curve that goes across FTC members, then we calculate the function that proposes a similarity degree to this line.

Property 2.1 (*smoothing FTC members*) The line or curve that traverses the fuzzy states of activity a may result from a linear smoothing technique such as linear regression, polynomial estimation or Savitzky-Golay filter and it is symbolized by $z_a = smooth(\{x \in FTC_{IR=\varepsilon}^{IR=\lambda}\})$.

Generally, these methods are different in the way they treat existing noise in data and linearity of the smoothing curve.

Property 2.2 (*similarity to* z_a). The reasoning in membership of an observation to a member of FTC is conducted based on the distance factor. The closer an observation is to an FTC_a member, the more similarity to z_a results, so more certainty in recognition of a is inferred. The distance of an observation to z_a is estimated according to this formula:

$$distance(v_{ti}, z_a) = \underbrace{min}_{t}\left(\sqrt{\sum_{i=1}^{n}(FTC_a - v_{ti})^2}\right) = \underbrace{min}_{t}\left(\sqrt{\sum_{i=1}^{n}((\int_{IR=\epsilon}^{IR=\lambda} FS_{a,k}^{IR}\, dIR) - v_{ti})^2}\right)$$

In Property 2.2, the distance of all FTC rows (members) to the observation is calculated and, out of these distances, the minimum distance represents the similarity that a is getting realized or, in other words, $v_{ti} \in a$. Regardless of the distance of observation to z_a, there is another parameter that affects the possibility of membership to an activity: it is closeness to bigger cluster centers. The reason for this is the fact that a bigger cluster center itself represents a set of more data points, so it justifies assignment of more possibility degrees to the points that are close to the bigger cluster centers.

Property 2.3 (ranking of points similar to z_a) If an observation is close to a bigger cluster center, then it would take more possibility degree than a point which is closed to a smaller cluster center. We propose to establish a direct relationship between cluster radius and distance to z_a indicated by:

$$rank_a(v_{ti}) = \underbrace{max}_{t}\left(\sqrt{\sum_{i=1}^{n}((\int_{IR=\epsilon}^{IR=\lambda} FS_{a,k}^{IR}\, dIR) - v_{ti})^2}\right)$$

Here, $rank_a(v_{ti})$ represents how much provides certainty in recognition of activity a as a candidate out all of the activities in A.

In order to calculate y_a, which represents the fuzzy-activity function or the space of activity a validity, a function indicating the relation between $rank_a(v_{ti})$ and $distance(v_{ti}, z_a)$ is established by $y_a = smooth(rank_a(v_{ti}), distance(v_{ti}, z_a))$. The smoothing function may include linear or polynomial regression methods or other famous smoothing methods such as the Kalman filter [105].

3.3.1.3 Simultaneous activities

Regardless of the quantity of activity performers, we often see two or more concepts getting realized. From another viewpoint, we can see that activities are the set of smaller entities (such as actions) that are simultaneously accomplished in order to achieve an activity realization goal. Therefore, parallel activities can be regarded as the consisting entities of larger concepts (such as social activities). By using set theory, it is possible to combine or analyze the constituting elements of concepts and, as a result, we can recognize high or low-level concepts rather than individual activities. The application of this contribution is the data-driven recognition of plans, behaviors, strategies, simultaneous activities and, also, lower-level concepts such as actions; therefore, we can reason for the activities interferences.

In recognizing simultaneous activities, we should analyze fuzzy world states in order to discover realization of which activities may possibly cause the surveyed world state. Applying traditional set theory, we would apply the union ∪ operation to find constituting elements. When two singular activities $a_k$and a_u are realized together, then we have $FA_s = FA_a \cup FA_u$, in which FA_s refers to the simultaneous activity. Here, we would treat fuzzy entities (fuzzy clusters) with traditional fuzzy set theory [6, 94]. The result is analyze/combine of the learned concepts. In order to analyze FA_s, we apply the set theory algebra based on known activities; which will be indicated in the following manner: $FA_s = FA_a + FA_u, -(FA_u \cdot FA_a)$. Therefore, presuming a_k refers to the known activity and a_u refers to the unknown activity. A key point here is that the fuzzy context of both activities should be respected in order for both activity models to be validated. Therefore, the resultant world state should be subjected to the fuzzy context $\tilde{C}_{a_s} = \tilde{C}_{a_u} \cap \tilde{C}_{a_k}$.

> **Proposition 3** (*simultaneous activity*). Let a_1, a_2 tow activities with respectively fuzzy state activities FA_{a_1}, FA_{a_1}. a_1, a_2 form a simultaneous activity, if only if it exits a fuzzy context $\tilde{C}_{a_1} \cap \tilde{C}_{a_2}$ such that $FA_{a_1} \cup FA_{a_1}$is a valid fuzzy state activity where the constraint $FA_{a_1} \cup FA_{a_2} = FA_{a_1} + FA_{a_2} - (FA_{a_1} \cdot FA_{a_2})$ is respected.

in which $\cap$ refers to the fuzzy AND operation and $\cup$ refers to the fuzzy OR operation on the cluster centers of activities a_1, a_2. Regardless of the quantity of activity performers, when a combination of two or more *concepts* is inferred, then a simultaneous activity is recognized. Usually, a simultaneous activity is realized in order to achieve a social goal. A social goal may be a desired world state that satisfies achievement of a set of individual goals. Thus, constraints of each of the running simultaneous activities should be realized in a logical space that satisfies all of the known goals. This space is created by using fuzzy logic in such a way that each activity saves its general structure and has partial flexibility if it faces deviations from the previously learned structure.

3.4 Anomaly recognition and assistance provision

In order to recognize anomalies, the Anomalies Recognition and Assistance Provision Reasoning System (ARAPRS) should distinguish normal states from abnormal states. It should justify observations by finding explanations for these. If there is anomaly, it should calculate what is missing. In the next step, ARAPRS should reason what the best reaction is in reaction to the discovered anomaly. The ultimate goal of ARAPRS is to return the world to normal state. Therefore, it can be inferred that deviation from a normal state and recognition of anomalies are the basic reasons of the world actuation for assistance provision.

ARAPRS requires some patterns according to which it could distinguish the normality of a world state. In applying extensions of fuzzy logic, we calculate similarity degrees to normal states and dissimilarity degrees to abnormal states. These criteria are used for accomplishment of automatic reasoning. In order to form the required knowledge for ARAPRS, information provided by the activity-recognition model is taken as input. A problem with this knowledge is that it does not indicate what an abnormal state is. In other words, this knowledge identifies which state is more similar to normal states (similarity measures to activity models), but it doesn't indicate a criterion for abnormality. In order to provide an abnormality criterion, the *error* concepts representing the opposite of normal concepts are calculated and formed. These opposite concepts would represent abnormal states. The main idea behind this process is that possible abnormal world states are states that are the relatively farthest concepts to normal states. The more an observation is similar to a normal concept, the further it is from its opposite concept then the observation is in a normal state. In contrast, the more an observation is similar to an abnormal concept, the further it is from a normal concept, so the observation is in an abnormal state. For implementation of the aforementioned idea, we calculate the fuzzy not of the knowledge hierarchy regarding to one or more activity attributes. The result of this process is symmetric positions of normal world states, which represent abnormal world states. These are the farthest world states as compared to known world states regarding available knowledge. In Figure 3.12, the symmetry of a known normal concept regarding an activity attribute applying "fuzzy not operator" is calculated by using the fuzzy membership function $\mu()$ of the sensor.

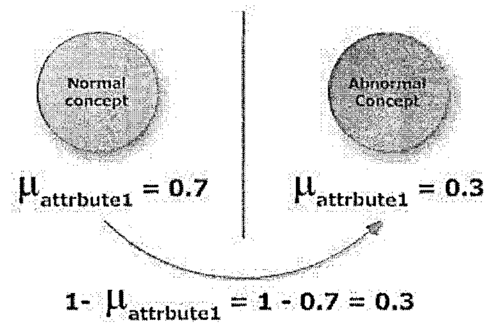


Figure3.12.: Discovery of abnormal concepts

At recognition time, proximity of live observations to proposed concepts would lead to recognition of world state normality. After an anomaly is recognized, it is time to find the best reaction to it because recognition of anomalies is not sufficient in the Smart Home and ARAPRS should react in real time to the recognized anomalies in order to return the home state to normal. This reaction is, in fact, assistance provided for the Smart Home resident. Reaction of intelligent systems such as the Smart Home depends initially on available facilities and embedded actuators. This is as well as the dependency of a Smart Home recognition power on the observatory facilities. We can imagine a day when robots will actuate a world by realizing automated activities; however, up to now we have not seen any research concerned with a robotic-based approach for assistance provision in the Smart Home. Although, we do not mainly intend to discuss a robotic-based approach for assistance provision, we hope that the proposed idea in this section be helpful for future researchers who will work in this area. For now, in the Smart Home project, it is presumed that there already is a resident which can accomplish activities, plans and actions in order to return the home state back to normality. Therefore, it is technically possible for this problem to be mapped to selection and play the best previously provided movie for the Smart Home resident in order to guide him correctly in his behavior. In other words, the role of actuators is to guide the resident to perform appropriate recovery activities.

In order to implement the automatic reasoning system for assistance provision, we need to find the activities that realize opposite functionalities, and then place these in front of each other. For example, the "cooling" action should be set against the "heating" action or the "dish washing" activity put opposite to the "cooking" activity. This task can be performed by applying a symmetrizer operator; however, this time the symmetrizer criterion is the concept that has the greatest similarity degree to all of the concepts. A Fuzzy Symmetrizer Operator (FSO) is proposed as an operator that performs on fuzzy numbers based on available knowledge. It calculates symmetric qualities of inputted concepts. This proposed operator applies "fuzzy not operator" on each attribute (sensor) of fuzzy multi-attribute objects in order to calculate symmetry of fuzzy concepts.

Proposition 4 (*absolute fuzzy symmetrizer operator*). Considering a set FTC_{IR} contains concepts that each of them has n fuzzy attributes $\bar{v}_{ki} \in \bar{s}_{i\bar{t}}^{k}$, then the symmetry of any fuzzy concept $\{O_t = (v_1, v_2, \dots v_n) \in FTC_{IR} \ / \ \mu_{v_i}(\bar{v}_{ki}) > 0\}$ would be calculated by performing the "fuzzy not operator" on each attribute: $AFSO\big(O_t = (v_1, v_2, \dots v_n)\big) = \acute{O}_t(1 - \mu_{v_1}(\bar{v}_{k1}), 1 - \mu_{v_2}(\bar{v}_{k2}), \dots, 1 - \mu_{v_n}(\bar{v}_{kn}))$

The symmetrizer criterion, in the sense that the concept that all of the concepts are symmetrized around it, is the concept which has the maximum similarity degree to all existing concepts. In other words, it is the cluster center of the FTC set. In the following example where we analyzed 10 observations from a world attribute reported in Table 3.5, the cluster center of these observations is discovered by applying the subtractive clustering method at an $IR = 2$.

Table 3.5.: Ten Normalized Temporal Observations from a World Attribute

Time of Observations	Value	Normalized Time	Normalized Value	AFSO(Time)	AFSO(Value)
1	12	0.0510	0.2472	0.9490	0.7528
2	17	0.1019	0.3502	0.8981	0.6498
3	14	0.1529	0.2884	0.8471	0.7116
4	17	0.2039	0.3502	0.7961	0.6498
5	20	0.2548	0.4120	0.7452	0.5880
6	10	0.3058	0.2060	0.6942	0.7940
7	13	0.3568	0.2678	0.6432	0.7322
8	15	0.4077	0.3090	0.5923	0.6910
9	16	0.4587	0.3296	0.5413	0.6704
10	17	0.5096	0.3502	0.4904	0.6498

In Table 3.5, we have illustrated that the tenth observation is the cluster center of the mentioned observations; around this concept, one notes that observations are symmetrized:

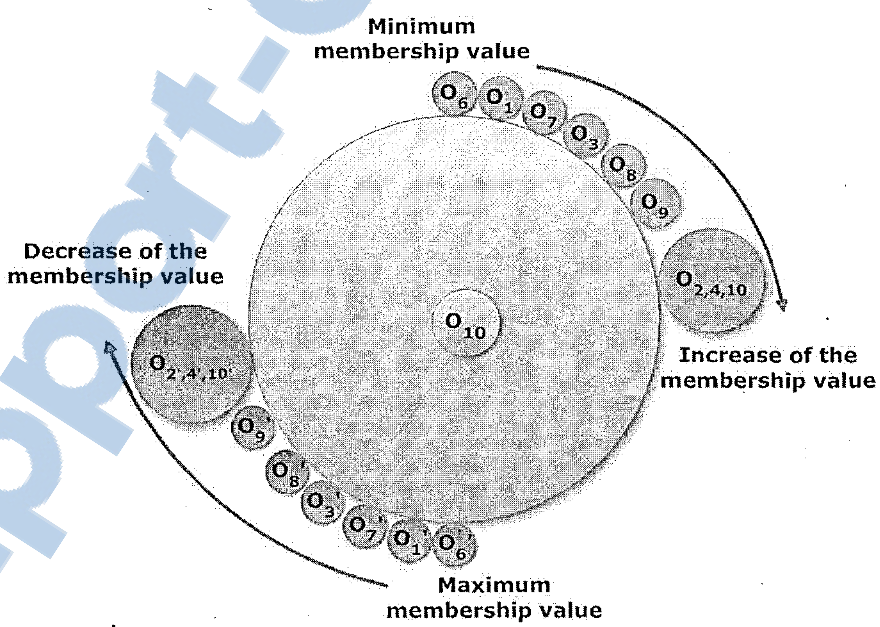


Figure3.13.: Symmetrizing concepts learned(indicated in Table 3.5) around the tenth observation

We have illustrated that concepts learned from system behavior can be symmetrized and normal states can be distinguished correctly from the abnormal world qualities. The abnormal states are colored with red and the normal states are colored with green. The tenth observation of this data set is the most similar concept to both normal and abnormal world state definitions. Anomalies can occur at every level of concept and we propose to perform anomaly recognition on each level of concept. The deeper the level is selected, the more detail about activities are considered and, generally, the more sensitivity is applied on the standard learned form of activity realization. If we wish to verify multiple levels of anomalies, we can imagine a typical symmetric fuzzy system such as the one illustrated in Figure 3.14.

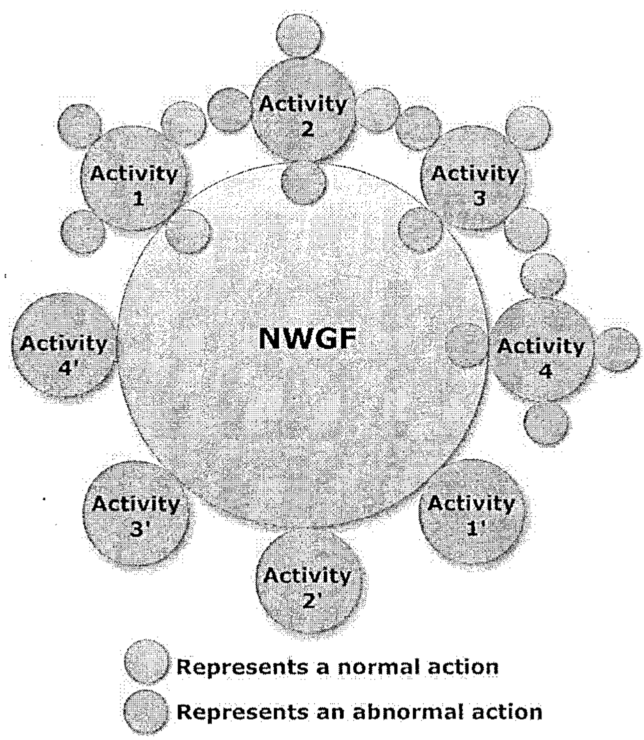


Figure 3.14.: Symmetrizing concepts at different levels

In Figure 3.14, not only details such as simple actions which can be symmetrized are presented, but also more complicated concepts such as activities that may also be symmetrized.

For each concept, many criteria can be used in order to find its symmetry because we may discover several cluster centers when more detail in fuzzy conceptual structures of activities is requested. In doing so, we may symmetrize concepts for each cluster center. Therefore, the ultimate fuzzy conceptual structure of the reasoning system can contain its symmetric schema. In order to perform anomaly recognition, we propose to carry out this process for each concept level by a top-down strategy. For example, considering the knowledge that we discovered at first, we begin with verification of the NWGF; if the world generally seems to be normal, then correct realization of activities are verified. When an anomaly is recognized, it is time to perform world actuation in order to return the world state to normal.

Similarly to what was done during the anomaly recognition stage, for the assistance provision step we will try to make a symmetric fuzzy conceptual structure, but we will not take the imaginary opposites concepts into account. Here, we put each normal concept face-to-face with other real normal concepts. Thus, we calculate the realization of the known action or activity that may return the Smart Home to its normal state.

Proposition 5 (*relative fuzzy symmetrizer operator*). Considering the set W contains concepts that each of them has n fuzzy attributes $\bar{v}_{ki} \in \bar{s}_{i\bar{t}}^{k}$, and let CC_{ki} the k$^{\text{th}}$ fuzzy cluster center ordered by the fuzzy time $\bar{t}$ of the set $\{\bar{s}_{i\bar{t}}^{k}\}$ then the relative symmetry of any fuzzy concept $\{O_t = (v_1, v_2, \dots v_n) \in FTC_{IR} \,/\, \mu_{v_i}(\bar{v}_{ki}) > 0\}$ would be calculated by performing the fuzzy extraction operator on each attribute $RFSO\big(O_t = (v_1, v_2, \dots v_n)\big) = \acute{O}_t(\mu_{v_1}(\bar{v}_{k1}) + 2(CC_{k1} - \mu_{v_1}(\bar{v}_{k1}),\ \mu_{v_2}(\bar{v}_{k2}) + 2(CC_{k2} - \mu_{v_2}(\bar{v}_{k2}), \dots, \mu_{v_n}(\bar{v}_{kn}) + 2(CC_{kn} - \mu_{v_n}(\bar{v}_{kn})\,))$

The symmetrizer criterion here is once again the concept that is most similar to all other concepts; however, this time we will take only real concepts (not imaginary ones) into account. In other words, the RFSO will try to put real concepts facing each other.

In the following example where we perform subtractive clustering on the ten temporal observations presented in Table 3.5, the value 15 is selected as the concerning cluster center. Based on this criterion, observations are set facing each other. In Figure 3.15, we have illustrated how these concepts are symmetrized around this cluster center.

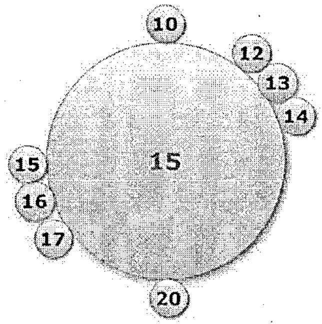


Figure 3.15.: Symmetrizing the normal concepts of Table 3.5 around the eighth observation

We have illustrated that normal concepts can be symmetrized. This operator would let us discover the correct activities which must be realized in order to return the Smart Home to its normal state.

3.5 Conclusion

In the current chapter, two complementary data-driven interpretations of activities are proposed. In both of these viewpoints, the discovered similarity between data points resulting from observation of activities is the axis of the reasoning in the concerning classification and clustering processes of data analysis. In the first viewpoint on activities, we discussed the fact that an activity is a downright dynamic entity and it can be recognized through inferable fuzzy events caused by activity realization. This approach is said to be "event-driven". In order to perform a data-driven process so as to discover fuzzy events, we proposed dividing the world into two general parts; thus, one of them represents the static characteristics of activities (fuzzy context) and the other represents the characteristics that change dynamically while activities are realized (activities' fuzzy states). As a result, it is proposed to perform a classification process to group together the common fuzzy states of all activities in order to succinctly provide all of the learned knowledge in a decision-tree format.

The second viewpoint proposed activities as being sorts of fuzzy temporal concepts that can be formalized as a multivariable mathematical problem. In this part of the thesis, we reasoned that an activity can be viewed as a highly dynamic entity if a high level of detail is desired and, in contrast, can be viewed as a singular entity (non-dynamic) if very general characteristics are required from it. Considering this fact, we discussed that several interpretations of fuzzy events can be made in a data-driven manner and all of these point to a single reality (activity observations). The activity-recognition reasoning system concerning this reality may produce several explanative hypotheses or concepts in order to perceive it. Then we proposed how generate all possible hypotheses about activities in a data-driven way. These hypotheses are synthesized in a smoothing curve or line and, at recognition time, the activities' functions check how much observations are similar to their smoothing curve. Finally, activities are ranked based on inferred similarities to these observations.

The proposed contributions provide a useful framework in order to deal with, analyze and solve many problems of the Smart Home through data-driven means. Recognition of activities in an automatic data-driven manner, recollection of simultaneous activities, reasoning about interruptive activities, reasoning about unseen inexperienced world states, prediction in Smart Home and many more applications are proposed based on the aforementioned approach. These will be discussed in the fifth chapter.

4 Validation

4.1 Introduction

In this chapter, we present the validation efforts and experimental results of the proposed contribution. We shall begin this chapter by briefly describing the validation context of our research, which took place in the infrastructure of the LIARA laboratory. Thereafter, we will present the objectives of the validation process; thirdly, we will explain the methodology of validation; then, we will describe the programming environments and software implementations. Finally, we will submit the results of experimentation conducted with the proposed model and give some statistical information, followed by a comparison with other existing approaches. At the end of the chapter, a conclusion about the validation and experimental results is presented.

4.2 Laboratory infrastructures

This thesis project was conducted at the *Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités* (LIARA) of the *Université du Québec à Chicoutimi* (UQAC). The LIARA laboratory possesses a new cutting-edge Smart Home infrastructure that is about 100 square meters and has around one hundred different sensors and effectors. Among the sensors, there are infrared sensors, pressure mats, electromagnetic contacts, various temperature sensors, light sensors and eight RFID antennas. Figure 4.1 shows some examples of LIARA's sensors.

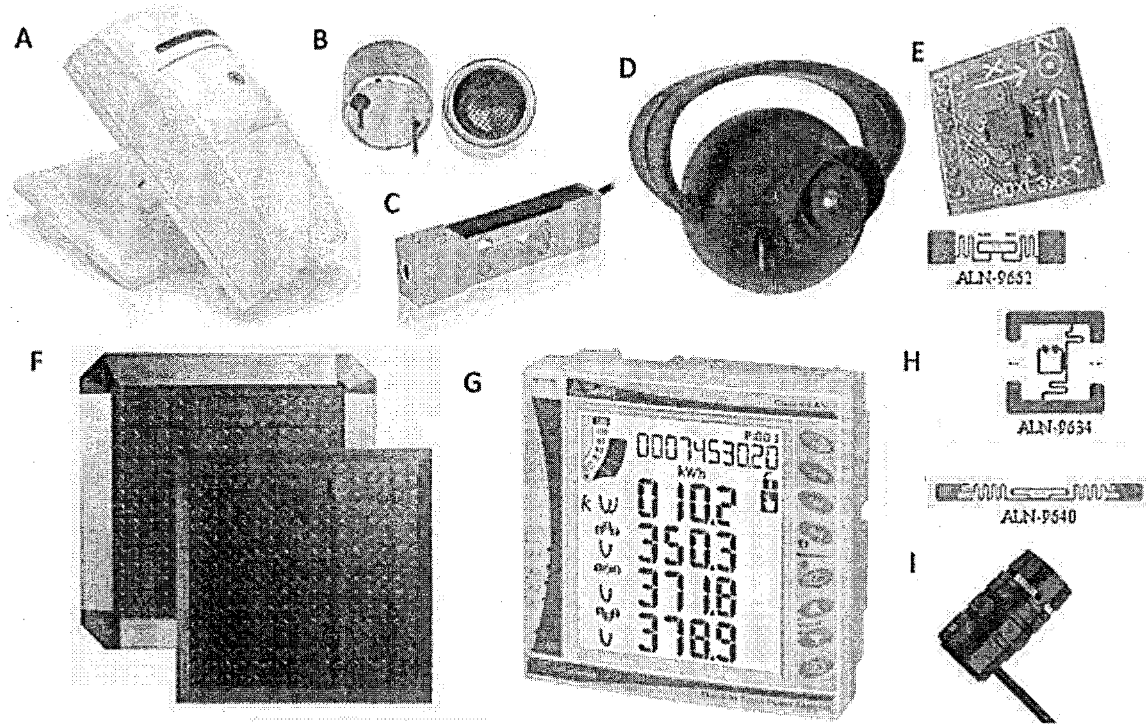


Figure 4.1. (A) IR motion sensor - (B) Ultrasonic sensor - (C) Load cell - (D) Video camera - (E) Accelerometer - (F) Pressure mat - (G) Smart power analyzer - (H) RFID tags - (I) Microphone.

The laboratory infrastructures has been designed and deployed following these guidelines [19] to ensure rapid AI prototyping. We also have many effectors, including an Apple iPad, IP speakers around the apartment, a flat screen HD television, a home theater and many lights and LEDs hidden in strategic positions that we can control at distance. Figure 4.2 shows a cluster of images from different parts and angles of our Smart Home.

The main image on this figure shows the kitchen. At the bottom, from left to right, you can see: a tagged cup (RFID tag), the dining room, an RFID antenna and the HD television. From the top right to the bottom, the following can be seen: the server, bathroom and library. The server is a Dell industrial blade computer, and it is in charge of processing information. We also have an AMX© system to control multimedia hardware such as the DVD player, television and IP speaker. As shown in Figure 4.2, the iPad is embedded in the refrigerator. It controls the habitat during experiments, while testing equipment or assisting the resident with the help of videos when he is in the kitchen. Speaking of assistance, the television can also be remotely controlled from a computer (or AMX) for this same purpose. Moreover, we can see inside the apartment with the mirrored windows mirror which have been specially designed for observation.

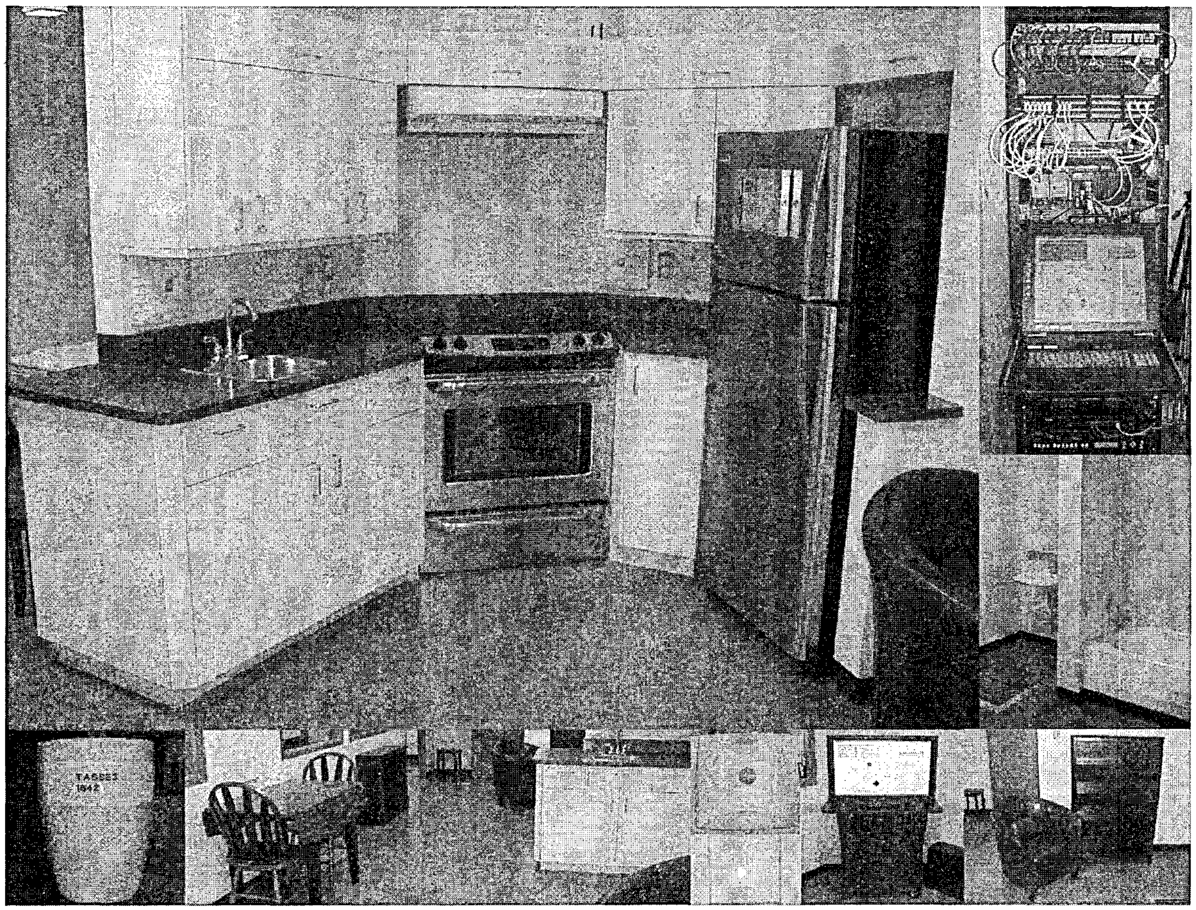


Figure 4.2. Pictures of the LIARA Smart Home

In order to facilitate testing, the lab is equipped with visualization software. A screenshot of this software showing the Smart Home overall can be seen on Figure 4.3. The graphical interface of this software allows us to see different parts of the Smart Home or the overall picture. In each of these interfaces, we can see the state of many sensors such as infrared sensors, light sensors, etc. We also can see the approximate location of the objects in the Smart Home, rounded rectangle, by proximity method, only appears if RFID antennas are activated and the current position of the resident is located in front of the kitchen counter on the right portion of Figure 4.3. These functionalities are very useful when conducting experiments since we can analyze what went wrong by reproducing sensors' activation and double checking if the material works properly. In addition, this allows for manual testing of the Smart Home's effectors, including the television, oven and audio system.

One of the most useful applicative features of our system is certainly the scenario-recording functionality. In fact, in the visualization software, we can create real-life scenarios with the touch of a single button (see the bottom right corner of Figure 4.3).

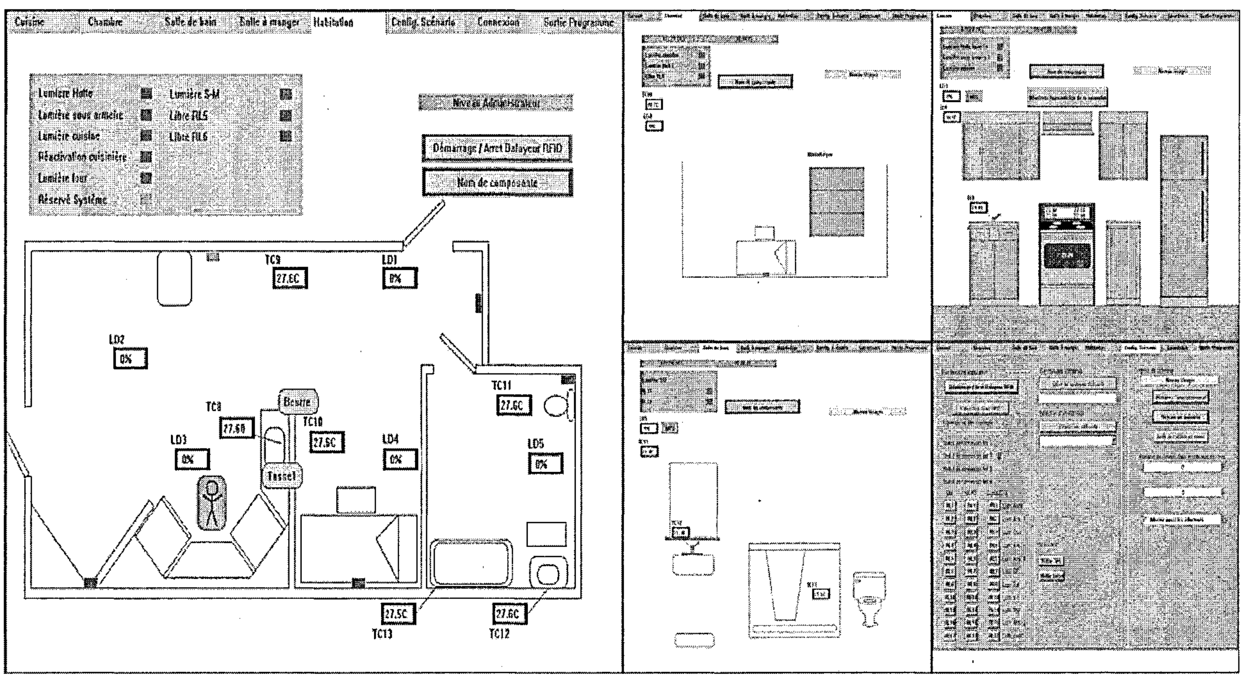


Figure 4.3.: Screenshot of the Smart Home controlling software

To do so, we only have to turn on the recording feature and ask a participant to perform the desired ADL normally in the apartment. When recording is activated, the Smart Home central application copies the data gathered to a third database layer that is identical to the main one. This feature has been used to gather the necessary data for validating the model proposed in this thesis project.

4.2.1 RFID technology

An important addition to LIARA'S Smart Home infrastructure is RFID technology [106]. Since this is the most important hardware related to a thesis project where the proposed model is mainly using the data about the position of the different objects, it seems important to explain a bit more about it. Our system uses eight antennas and passive, class 3 RFID tags. This kind of tag is self-powered and possesses no memory. It relies solely on antennas to work correctly. These antennas send a pulse in the form of radio waves every few milliseconds; then, they listen to receive any response from the surrounding tags. A passive tag is in a dormant state and it is the fact of receiving one of these waves that "wakes" it. Thereafter, it draws energy from that same wave to emit in turn a radio wave containing information on its ID. Of course, compared to their active counterparts that use their own power to emit and that are always awake, passive tags have reduced range and precision. However, they possess many desirable properties for our context, such as their small size and low cost (often less than a penny). These two qualities are what allowed us to tag all objects in the Smart Home. The kind of tags we used is so robust that the objects tagged are washable without causing damage to these when they remain on the object. RFID antennas have a detection zone ranging from less than a meter to around 3 meters depending on system adjustment. The basic implementation of our infrastructure rests on a proximity algorithm in order to roughly locate objects around the Smart Home. To do so, the system is adjusted to 38 db power and sensitivity of 18 db. The goal is to have coverage bubbles that do not overlap. In parallel with this thesis project, we are working on the improvement of our objects' localization system [106] to obtain higher precision which is essential to the success of this project. Our goal is to create a hybrid between proximity-based localization and a trilateration algorithm.

4.3 Validation approach

As it was mentioned earlier, this thesis proposes a contribution to the artificial intelligence, data mining and activity recognition fields of science by new proposals in modeling activities as series of fuzzy events, extraction of activities' fuzzy conceptual structures, modeling activities as multivariable equations, modeling simultaneous activities and anomaly recognition for simultaneous interruptive activities.

Validating all of the proposed elements of this contribution was not easy. In order to be able to implement proposed procedures, steps, definitions, formalizations and propositions, we decided to use MATLAB software [47] as a simulation platform to test the proposed model. Moreover, a Visual Studio.Net environment and Microsoft SQL server were also used. The reason for select MATLAB as the main simulation environment is that by transforming the format of the initial primary data into a matrix, we could benefit from the matrix operators of MATLAB and extract knowledge in matrix format, which let us apply and represent this knowledge in various forms.

The major part of the validations and experimental results of this thesis were already presented and published in the form of two papers in the *Springer Journal of Ambient Intelligence* and *Humanized Computing* [12, 13] and in the proceedings of the *AAAI STAMI* workshop [8].

4.3.1 Experimental objectives

One of the main objectives of the validation phase was estimation of complexity in the data-mining process, ensuring the practicality of proposed solutions and obtaining experimental results for statistical analysis.

Process complexity is an important factor in the estimation of efficiency in any algorithm. Inefficiency of algorithms may lead to unpredictable problems pertaining to runtime. For example, if an algorithm processes input slowly, then process overload will occur. Thus, we can predict some contexts in which the system does not react appropriately to context necessities in real time. The incapability of real time reaction may consequently prevent the system to follow up on the scenario, so more consequences would be expected to happen. Except for quickness, one other property of process complexity is the rate of success in estimation. In other words, we measure how accurately an algorithm can successfully predict training samples in order to estimate approximate success rate at runtime. In this thesis, we verify both the success rate and quickness factors of the proposed approaches in order to ensure efficiency of the algorithm.

Apart from process complexity, practicality of the proposed approach should be verified. In fact, any proposed modeling approach would intend to perceive the real world, interpret the ongoing events and propose some solutions for some specific cases. When the system perceives the real world correctly, takes into account uncertainty and imprecision which are properties of any real-world problem, then we may expect the practical aspects of the proposed approaches to interpret observations accurately. Except for consideration of uncertainty and imprecision parameters, we should consider the requirements for training the model and adapting it to partial changes. In other words, if the model is not easily trainable or if it depends highly on specific contexts, the proposed approach may fail in runtime. The success rate of the proposed approaches in this thesis reveal an index for comparison of the occasions in which our proposed approaches can operate better or worse than other proposed approaches. Therefore, we provide statistical information about success rate, required training data, efficiency, precision and other aspects of the thesis' project.

4.3.2 Targeted applicative context

The contribution of this thesis aims to be generic in its fundamental form, so that it can be used in multiple applicative fields. However, the main applicative context of the LIARA laboratory is assistance to people suffering from Alzheimer's disease [107] who need continuous and permanent care-taking to be able to stay at home. This motivation directed our focus on the correct realization of activities rather than simple hypothesis generation for probable or possible plans. Therefore, we tried to perform some erroneous tests that simulate the ways in which the mentioned target group would probably perform the same or similar mistakes. Moreover, we intended to verify whether the proposed approaches could help the target group. For example, in our validation, we tried to simulate cases where an object in realization of a specific activity is not applied in a proper order; consequently, we verified if this mistake was recognizable and, if so, the inference of proper assistance to the Smart Home resident. Furthermore, we verified if system complexity enables real-time observation, data interpretation, reasoning and inference making.

Regardless of the aforementioned initial motivation in selecting the target group, consideration of its problems could lead us to verification of some of the worst logical scenarios that may happen in the Smart Home. The result is that even if an ordinary person performs improbable mistakes during his or her activities, the proposed approach may support these situations. Therefore, verification of the target group's problems would also be a proper criterion for selection of proper tests, which challenge the precision in data interpretation, reasoning and inference making.

4.3.3 Chosen activities, modeling and features for simulation

In order to model some representative activities for simulation, we needed to define observable features activities which would be the object of our research. For example, duration, delays between simple actions, beginning/ending points of an activity, objects used and their displacements and environmental features (like temperature or light) are considered for modeling. In other words, we took some spatiotemporal specifications of these activities into account. These features served as input for our learning and recognition process.

The three chosen activities we modeled for testing our approach were "*coffee making*", "*hand washing*" and "*reading*". We chose these activities because they are representative of the common daily activities elderly people may carry out, they can be monitored using the spatio-temporal features that we described, are of short duration, have an appropriate number of steps and are commonly used as examples and case studies in Smart Home scientific literature [108].

4.3.3.1 Coffee-making activity

The *coffee-making* activity is a basic ADL in which several objects are involved. These objects are the *coffee pot*, *hot water* (or *kettle*); *milk*, *sugar*, *spoon* and *cup.* In our case, RFID tags are attached to each of the aforementioned objects (or on their container).

In order to estimate the location of objects in the environment, we applied some RFID antennas, which can capture a distance of up to one meter. In totally, eight antennas were used to obtain localization information during tests.

This activity is typically performed in the following manner (the order may vary): a cup is filled fully with hot water; the Smart Home resident finds necessary objects to realize the activity in the environment's kitchen cabinets and puts them on the table. Then, sugar, milk and coffee are added to the cup. Finally, the Smart Home resident mixes the contents of the cup. In addition to object displacement, Smart Home software estimates the location of the Smart Home resident and captures the opening/closing events of kitchen cabinet doors.

This activity is good for testing because it contains temporal features (order of the task, duration, delay) and spatio-temporal elements (objects' positions, distance, time, etc.).

4.3.3.2 Hand-washing activity

The *Hand-washing* activity is also a basic ADL that may be realized at any time in the home in many locations. This activity faces some limits in its temporal features and its structure is relatively more rigid in its realization than other activities. In addition to RFID sensors, temperature sensors located in pipes are used to monitor this activity. Here, we are dealing with multi-state but non-spatial sensors; however, the temporal features of this activity may be interesting. For example, water temperature should be between some specific parameters (limited degrees); however, the time (duration or delay) for it staying within these temperatures should be controlled.

The *hand-washing* activity is normally done in the following manner: the Smart Home resident enters a zone (the kitchen for instance) and approaches the sink area; then, the resident opens the cold/hot water taps, uses soap, washes his or her hands, dries his or her hands and closes the taps. Finally, the resident leaves the sink area. During this activity, not only should temporal features (such as delay in opening the water valves) be taken into account, but supplementary constraints, such as water temperature and presence of the resident at the sink, should also be considered.

4.3.3.3 Reading activity

The *reading* activity is a simple and very common ADL in which on-off and multi-state sensors are actuated at realization time. Moreover, this activity is normally done in a specific location. Although the duration of this activity is highly variable (there is not a specific temporal constraint), this activity goes on as follows: a book is taken from shelf, then the resident goes to a sitting space with the book; after a while, the book is returned to the shelf.

4.4 Implementation and testing

The model's implementation and simulation mainly relies on MATLAB software. The concerning validation was conducted based on the data of tests performed in LIARA. In addition, we applied the Microsoft SQL server, Microsoft visual studio.net and SPSS in a Microsoft Windows 7 environment. The data types are basically transformed into matrix format (kept in dynamic address translation or .dat files) for data analysis, as well as process results and inferred knowledge. Each selected activity was performed several times. These activities were realized in several different manners depending on recognition objectives. In some tests, we have purposely realized activities in an erroneous manner, in which the order of some actions were incorrectly accomplished or simply omitted.

The implementation and validation of our main contributions contain five major steps [12]. At first, observed data is mined and models for verification of the dynamic features of activities are derived (event-driven modeling). Then, an event-driven recognition system in the form of a fuzzy inference system (FIS) is proposed. During the third step, conceptual structures for activities are calculated and conceptual models are discovered for these. In fourth step, calculated knowledge is applied in anomaly recognition and, in the last step, the best reaction to the anomaly is estimated. Implementation and validation of each mentioned step is presented in the form of case studies.

4.4.1 Validation of the fuzzy event-driven modeling and mining approach

In order to model activities as series of fuzzy events which occur in the ambient environment, we fuzzified our observations and divided these into fuzzy clusters. These fuzzy clusters are estimated according to data specifications and the expert may not impose any idea on the cluster estimation process. The implementation of *subtractive clustering* in MATLAB is called for at this time. In this step, time is like other activity features such as spatial features, and is fuzzified; the fuzzy time intervals representing temporal features are automatically calculated.

The fuzzy time intervals represent the possibility of occurrence degrees of expected actions and delays between the actions. Although, these factors are estimated in a data-driven and automatic manner, we do not need to represent all of the temporal features in as understandable manner for the human mind because the recognition step is also performed in a data-driven way.

We have organized a case study for this section. In this case study, we surveyed a single activity - *coffee making* - and concentrated on pattern recognition arising from it. More than 500 features of this activity were observed in the actual Smart Home infrastructure. After modeling these observations in order to propose a recognition-based approach, we profited from the fuzzy inference system (FIS) implementation in MATLAB.

4.4.1.1 Case Study 1: fuzzy modeling of the coffee-making activity

In this section we will verify and discuss the activity of *coffee making* as a case of pattern recognition. Both correct and wrong realizations of this activity were compared by a Fuzzy-inference system (FIS). For this case, we did not apply fuzzy events and we tried to discover the positions in which objects were mostly located, while this activity was realized. The objective of this case study is to show the potential of fuzzy logic to model and evaluate activities. The approach used in this case study cannot evaluate individual observations and needs complete realization of activities to be evaluated because it needs two completed patterns for comparison.

In the lab, 560 sensor events are observed every 0.25 second. At an influence range of 0.5, we proceeded in the fuzzy clustering of these observations and we reduced data to a 246 in 560 matrix, which contains the cluster centers of these observations. A fuzzy-inference system is made based on the available knowledge about the activity. FIS is also trained with wrong realization of the activity, in which for example sugar has been forgotten to be used. According to the available knowledge, by training the system with one normal and one wrong realization, 22 variables were selected as effective variables to recognize normal realization of the activity. The concerning fuzzy rule is illustrated in Figure 4.4.

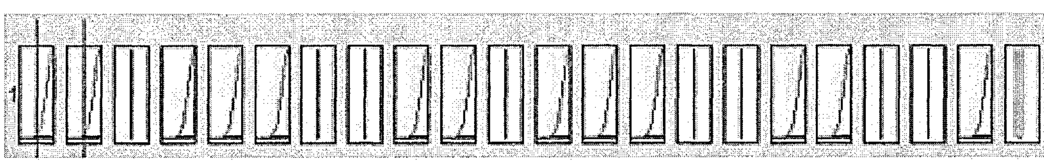


Figure 4.4.: Fuzzy classes for the normal realization of the coffee making activity

To evaluate the reasoning system, we decreased the number of world features from 560 to 246; then, the number of inferred fuzzy classes was reduced to 19 for this state and again, by decreasing to 125 variables, the number of inferred fuzzy classes reduced to 6. See Figure 4.5.

The FIS was tested by evaluating two similar normal realizations of the *coffee-making* activity. The evaluated similarity degrees of the correct realizations and learned patterns are illustrated in Figure 4.5. By decreasing the quantity of training variables, the similarity degree was reduced. The evaluation process was done by using the $evalfis()$ function[13] in MATLAB.

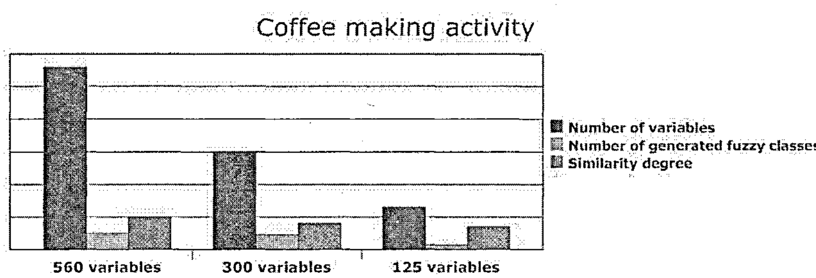


Figure4.5.: FIS at an influence range of 0.5 with a different number of training variables

According to the experimental results, it can be inferred that the quantity of calculated fuzzy classes are dependent on the number of training variables. When more fuzzy classes are calculated, then more world states are considered and so, more precision in reasoning applied to separate the normal and abnormal world is expected.

4.4.1.2 Discussion on Case Study 1

When an object is positioned in a special location for a longer time rather than in other locations, then the data related to object position would concentrate around it. This position is calculated by the fuzzy subtractive clustering algorithm and it is a meaningful signal to recognize

---

[13]http://www.mathworks.com/help/toolbox/fuzzy/evalfis.html

the activity. We expect that the object would be positioned once again in the same area(s) during future repetitions. For example, *sugar* is mostly positioned next to the cup, as well as the table. These positions are significant in recognizing the activity.

In the aforementioned case study, we could model the activity of *coffee making* without use of the fuzzy-event concept and, instead, we would use the possible positions of objects to infer activities. It is illustrated that, as the number of observation variables (sensors) increases in the learning and reasoning process, the more fuzzy classes are defined and the more criteria to judge correct realization of activities are formed. The training process for each of three steps was done in less than one second by a "Windows 7 operating system", Quad core 2.66 GHz processor and 4 GB of RAM. This case study empirically reveals that the proposed approach welcomes more sensors in training and its process complexity is not increased significantly when the quantity of input increases. Furthermore, we can observe that this system is not completely dependent on sensors' positions, and sensors elimination or addition will not cause invalidity of existing knowledge. The problem with this activity modeling is that we can evaluate the activities only when they are completely done and we do not have the possibility to reason during an ongoing situation (at runtime); the order of performed actions is also a limitation. So as to consider world dynamicity, actions' occurrence order and being able to reason at runtime, we should regard the world as being a chain of fuzzy events.

4.4.1.3 Case Study 2: inference of ARP/APP during the coffee-making activity

For this case study, we organized several tests so as to recognize ARPs and APPs concerning the coffee-making activity. In order to broaden the study, we also tested two variations of the *coffee-making* activity: *making tea* and *drinking water*. To do this, we used objects like a *coffee maker*, *tea pot*, *spoon*, glass/cup, *sugar*, *coffee*, etc. RFID tags on the objects were the main sensor events. Each activity was done twelve times in different manners. Sensors observed actions concerning each activity by tracking both the objects and the resident. Observation is the first step in learning patterns of activity realization. Embedded sensors in the environment take care of the observation function. ARPs are inferred by trace-of-object displacements. Objects displacements are inferred if these get close (or far) relatively to special points in the environment. In this way, activities are defined by displacement of objects in space. The position of objects to special geographical points can be described by partial membership of their distances to specific fuzzy classes such as *near*, *intermediate distance* and *far*; however, in the implementation two simple concepts, *far* and *near* are applied. The movement of objects regarding each geographical point can be done in two ways: *getting closer* or *getting farther*.

All of the aforementioned definitions are fuzzy terms and these can be defined through fuzzy functions. Therefore, at implementation time, instead of directly applying the integers representing the distance of objects, we applied fuzzy membership measures of the distances to fuzzy classes. It should be mentioned that we did not take into account the position of objects in x-y scaling or their precise distance, but only their movements and displacements regarding some specific positions, which were in fact antennas' positions.

Implementation of the fuzzy approach lies on the concept of the fuzzy event. In this approach, events are inferred from each meaningful change in position of objects regarding some already known geographical position(s). Observance of special events could mean recognition of a special activity. To infer the occurred event, we apply the following membership function to be prepared for the classifier:

$$\mu(far) = \begin{cases} 1 & if\ d < b \\ \dfrac{d-a}{b-a} & if\ a < d < b \\ 0 & if\ d < a \end{cases}, \quad \mu(close) = \begin{cases} 1 & if\ d < a \\ \dfrac{b-a}{d-a} & if\ a < d < b \\ 0 & if\ d > b \end{cases}$$

$$\mu(get\_far) = \mu^1_{object}(far) - \mu^2_{object}(far)$$

$$\mu(get\_near) = \mu^1_{object}(near) - \mu^2_{object}(near)$$

The aforementioned approach in defining a fuzzy event is valid only if the expert has an idea about the possible positions of objects relatively to the antennas, which are in either far or $near$ positions in the mentioned definition.

In the case that the expert does not have any idea about possible states of data records or at least he intends not to involve his idea into the pattern recognition process, then we can describe the positions of objects comparatively to special geographical points (antenna locations) as partial membership of their distances to specific fuzzy classes such as *near*, *intermediate distance* and *far*. The movement of these objects relatively to each geographical point can be made in two ways, either *getting closer or getting farther*. Here, a tuple combined from couples of events and their occurrence order indicates the realization pattern for an activity named *activity_1*.

$$\begin{aligned} &Realization\ Pattern\ for\ activity\_1 = \{< get\_close(O1\_A1), 1 >, \\ &< get\_far(O1\_A2), 2 >, < get\_close(O2\_A1), 3 >, \\ &< get\_close(O3\_A1), 4 >, < get\_close(O3\_A2), 5 >, \\ &< get\_close(O4\_A1), 6 >, < get\_far(O4\_A2), 7 >\} \end{aligned}$$

In this example, O_i represents an object that is applied in realizing *activity_1*, A_i designates one of the geographical points that are fixed; displacements of objects are defined relatively to these. In this model, non-fuzzy events like *turn oven on or off* can be merged to complement and provide supplementary information. Also, in order to learn spatio-temporal patterns of activities, values that are observed at first are compared to each other and then, the displacements of objects are inferred as events. To derive the order of events, we assign a number to these, according to their occurrence order. ARP is in fact the ordered, inferred set of events concerning activities. The creation of a *prediction pattern* resulting from the classification of activities is one of the goals of our research. APPs can be calculated by comparing all of the ARPs together by using a classification process. For example, through application of the proposed algorithm described in the preceding chapter, a decision tree that predicts the intended activity can be drawn (Figure 4.6).

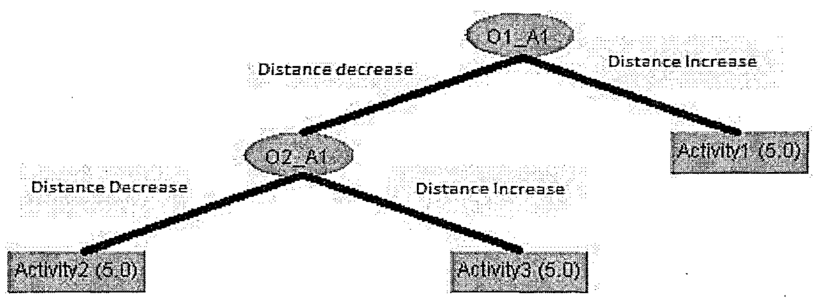


Figure 4.6.: Decision tree illustrating an APP about three *coffee-making* related activities.

Through use of APPs, we can predict the intention of the resident. In this way, when a resident starts realizing activities by accomplishment of a few actions, we would be able to recognize his or her intention concerning his or her final goal. In figure 4.6, each node of the decision tree indicates the relation between an object and its RFID antenna, and each branch indicates the occurrence of a fuzzy event on its upper node. Rectangles indicate possible intended objectives. For example, in this case study, if Object1 (O1-cup) gets nearer (operation performed by the Smart Home resident) to Antenna1 (A1 – at proximity of the tea maker) and then, if Object2 (O2-sugar) gets nearer to Antenna1 (A1), the system then infers that the resident intends to realize the activity of *making tea* (which corresponds to Activity2-*making tea* in the tree).

4.4.2 Modeling and recognition of activities as fuzzy temporal concepts

In the previous chapters, we mentioned some reasons for improvement of the fuzzy event-driven modeling approach. In this part of our research, we applied fuzzy events in order to form fuzzy temporal concepts, which describe the structure of the activities, so that the new approach is an extension of the event-driven approach.

4.4.2.1 Case Study 3: Modeling of activities as a multivariable function

In order to validate these proposed ideas, we tested three selected activities: *coffee making*, *hand washing* and *reading*. The matrix operations were simulated in MATLAB. Each activity was performed four times. Applying the fuzzy-context filter (refer to Chapter 3 for a reminder), we calculated that the sensors presented in Table 4.1 were the ones that play a significant role in realization and recognition of these activities and the rest of sensors should be taken into account as fuzzy-context members. In order to recognize simultaneous activities, we need to apply the proposed model operators on extracted knowledge, so that we may transform the discrete form of the activity model, which are the FTC members; to a continuous form. In this process, all of the aforementioned discrete values are integrated by a traversing line or curve. In fact, fuzzy cluster centers of temporal data are smoothed through a method such as multiple regressions. At recognition time, the similarity (distance) of live observations to this multidimensional function is estimated. Then, activities are ranked based on discovered similarities between live observations and activities' models. As an example, to calculate fuzzy states of the *hand washing* activity, we discovered that this activity is seen as a single world state at λ=7. In step-ward manner decrease of the Influence Range ($\Delta IR = 0.1$), the quantity of discovered transitional world states increases and is more dynamic, but less generic definitions for the activity is taken; at $\varepsilon = 1$, all of the 40-second observations from this activity are transformed to a 29-section scenario, which includes enough detail about the activity. In order to transform the discrete world states into a continuous single form, fuzzy states at higher IR rates take higher-similarity measures as weight for multiple-regression training because these represent more data points in bigger clusters. Here, we have directly assigned the concerning IR rate to fuzzy states calculated with the same cluster radius. For example, at $IR = 0.7$ the single world state is assigned the weight of 0.7, or each fuzzy state at $IR = 1$ is assigned the weight 1, so that we could calculate regression on the fuzzy states of this activity. Equation 4.1 represents the structure of the *hand-washing* activity:

Table 4.1.: Role Playing Variables in Case Study 3

s_i	Variable title	Description
s_1	Time	Daily time
s_2	CE2	Doors open/close
s_3	DB1	Water temperature
s_4	DB2	Water temperature
s_5	CB2	Panel magnetic
s_6	CB6	Panel magnetic
s_7	TC8	Tap temperature
s_8	MV1	Motion sensor
s_9	MV2	Motion sensor
s_{10}	MV4	Motion sensor
s_{11}	MV5	Motion sensor
s_{12}	T_base	System Sensor
s_{13}	CD1	Bookshelf Infrared
s_{14}	CD2	Bookshelf Infrared

$$\begin{aligned} y_{handWashing} = {} & 6.417023379 \cdot 10^{-3} s_1 - 9.353899771 \cdot 10^{-2} s_2 - 2.967082486 \cdot 10^{-1} s_3 \\ & - 1.110231388 \cdot 10^{-2} s_8 + 3.611645773 \cdot 10^{-2} s_{11} + 1.17149016 \cdot 10^{-3} s_{12} \\ & + 1.41532962 \end{aligned}$$

Equation 4.1.: Equation of the *hand washing* activity

In Equation 4.1, time events and sensor events play a meaningful role in modeling and recognition of activities. This formula indicates the structure of the *hand washing* activity by representing a *fuzzy space* indicating the expected sensors behavior over the period of the time line. In this case study, we surveyed the modeling process of activities and the way that we mined the data to achieve summarized knowledge about a big data set. This knowledge briefly describes the general characteristics of activities with their dynamicity properties. During the next step, we surveyed modeling of simultaneous social activity in order to verify how we can distinguish individual activities from social activities.

4.4.2.2 Modeling of simultaneous activities

Simultaneous realization of two or more activities (regardless of who the activity performers are) is considered a social activity. In other words, we can justify observations by a pattern characterizing a *social plan* in which concurrent activities may be realized. In this part of the case study, we verified the simulated, concurrent realization of the *coffee making* and *hand washing* activities. The modeling process of a social activity is similar to individual activity modeling: during the first step, observations resulting from realization of a social activity are modeled as sequence of fuzzy events occurring in the world [12], at a range from $IR_{high} = \lambda = 0.5$, down to $IR_{low} = \varepsilon = 0.2$. Here, the $\varepsilon = 0.2$ represents the level of detail which one wishes to include in the activity model. The discovered activity model is:

$$y_{HWashingAndCoffeeM} = -1.286066617 \cdot 10^{-3}s_1 - 1.6196889 \cdot 10^{-1}s_2 + 0.491954717s_3 - 4.416616668 \cdot 10^{-1}s_4 - 5.479516397 \cdot 10^{-1}s_5 + 2.531096405 \cdot 10^{-1}s_6 + 1.874326049 \cdot 10^{-1}s_7 + 3.428391878 \cdot 10^{-2}s_8 - 1.26879522 \cdot 10^{-1}s_{10} - 1.443386543 \cdot 10^{-1}s_{11} - 1.550049851$$

Equation 4.2.: Equation for coffee-making / hand-washing simultaneous activity

In Equation 4.2, the behavior of the sensors over the period of the time line, during simultaneous realization of the *hand-washing and coffee-making* activity, are indicated.

In this part of our research, we intended to analyze a simultaneous activity model in order to discover forming of individual activities. In other words, in this problem, if the observations O of a simultaneous activity are modeled by fuzzy state function $FA_s(O)$, and a known activity that is one of the forming activities is modeled by $FA_k(O)$, then the subject to discover is the unknown-activity model represented by $FA_u(O)$. The function of the unknown activity is calculated by the following equation:

$$FA_s(O) = FA_u(O) + FA_k(O) - \left(FA_u(O) \cdot FA_k(O)\right) \rightarrow FA_u(O) = \frac{FA_s(O) - FA_k(O)}{1 - FA_k(O)}$$

Equation 4.3.: Relation between observations and individual activities

In Equation 4.3, s refers to simultaneous activity, u refers to the unknown activity formula and k refers to the known activity formula. Considering that we know the model of the simultaneous activity of *coffee making / hand washing*, then we would break it into its forming concepts, which are individual realizations of *coffee making* and *hand washing*. Presuming that the function of *hand-washing* activity $FA_{HWashing}(O)$ is proposed in Equation 4.1., and that the simultaneous-activity function $FA_{HWashingCoffeeM}(O)$ is proposed in equation 4.2, then the unknown-activity function $FA_{CoffeeM}(O)$, which is the activity of *coffee making,* can be estimated in Equation 4.4:

$$FA_{coffeeM}(0) = -4.3800s_1 + 0.025s_2 + 1.0830s_3 - 0.4410s_4 - 0.0130s_5 - 0.5470s_6 + 0.1870s_7 + 0.0560s_8 + 0.1260s_{10} + 0.0720s_{11} - 4.3800$$

Equation 4.4.: - Equation of the *coffee-making* activity

In this equation, the fuzzy space in which the activity of *coffee making* is expected to be valid in is estimated. As it was mentioned earlier, in order for an intelligent system such as ARRS to perceive the real world, it makes concepts out of observations. In this part of the thesis, we propose how the ARRS may represent concepts in a data-driven manner. In order to propose a schema for models that may be derived from a temporal data set, a hierarchy that indicates simultaneous realization of activities may be created. The biggest and the most generic concept is the NWGF, which represents simultaneous realization of all activities; at its bottom level (the lowest level of observations), we are able to model initial data. Figure 4.7 shows the hierarchical organization of concepts corresponding to multiple activities:

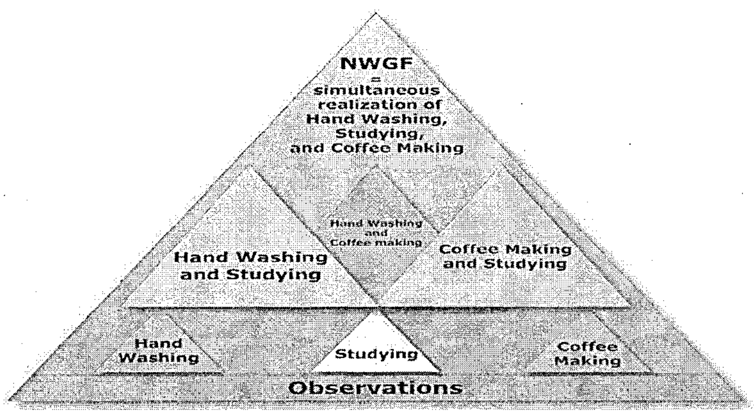


Figure 4.7.: Discovered hierarchy of concepts in the Smart Home

In order to organize the knowledge that we have used in this case study, we propose to draw it in a schema such as Figure 4.7. The ARRS perceives the world and its ongoing events. It then tries to find individual activities and their combined realizations. These concepts which have been learned result in a schema like the one represented in Figure 4.6., and it may be completed by the concept-making and concept-analysis approaches mentioned in Chapter 3. The hierarchy of concepts situates observations and individual activities at lower levels; simultaneous activities are found at higher levels; all of these concepts are categorized. At the highest level there is the

NWGF, which represents the simultaneous realization of all activities in the Smart Home. This high-level concept represents all knowledge at a glance and may be applied for fast reasoning in anomaly recognition [13, 46, 109].

4.5 Experimental results

In this part of the thesis, we present the functionality and effectiveness of the conceptually-driven data-mining and modeling approach. We first performed each selected activity four times in both individual and simultaneous manners in the LIARA laboratory. More than 500 activity features were observed during realization of these activities. Individual realization of the *coffee making*, *hand washing* and *reading* activity was performed; the simultaneous realization of coffee making and hand washing was then performed. At the end of these activities, we verified the estimation of similarity of these activities to the models which had been learned while applying multiple regression.

In our experiments, we saw that the calculated simultaneous activity could approximately replace the real simultaneous model. Other experiments we conducted also confirmed reliable results in this regard. The important point to note here is that we achieve this result through application of the activities' multivariable function which explains the activities' dynamicity at a glance. Considering the previously discussed individual activities, which are *hand washing*, *reading* and *coffee making*, we would randomly select one observation for each activity and test what similarity degrees resulted through these models.

It is desirable that activities' models confirm their own observations with a higher rank than other possible models. The selected observations are the ones that are found during the 35th

second of each activity realization. The result of this calculation, which leads to reasoning in the recognition of an activity, is indicated in Table 4.2:

Table 4.2.: Similarity measures of observations pertaining to activity functions

	Hand washing	Studying	Coffee making
Hand washing sample	1.343215	5.1748	- 153.5411
Studying sample	1.3419	5.8724	- 153.5011
Coffee making sample	1.3431	4.822	- 153.1001

As it is described in Table 4.2, similarities' measures between *observations* and *models* are calculated, which illustrate a schema from recognition results of the ARRS. In this example, we verified that each pattern (model) confirms the concerning *momentum* observations with a relatively high similarity measure. For example, the *hand washing* sample is given a higher rank by the *hand washing* model as compared to the *reading* and *coffee making* models. Consequently, in Table 4.3, we demonstrate the calculated similarity degrees in the case that individual activities concur with simultaneous activities. The tested samples (observations) are found during the 35^{th} second of each activity data set:

Table4.3.: Similarity measures of individual activities versus simultaneous activities

	Coffee making	HW_CM	CM_studying
Hand washing sample	- 153.5411	1.555110386	0.40259372
Coffee making sample	- 153.1001	2.383138	7.05006674794
HW_CM sample	- 153.4338	5.10132771912	0.841671309
CM_studying sample	- 150.7728	2.460629941	1.490270749

In Table 4.3, it can be seen that similar to the latter example, the *inferred* simultaneous and individual models indicate better similarity measures to their concerning observations. An advantage of the proposed model is that not only we can reason about momentum observations concerning the world, but we can also reason about the group of observations in order to assign these to their appropriate concepts. This property would help to reason about the dynamics of the world. In order to do this task, we considered 15 seconds of the realization of the *hand washing* activity; 15 seconds from the realization of the simultaneous activity of *coffee making* and *reading*, and 15 seconds from the realization of the simultaneous activity of all three activities. The aforementioned observations were selected from the 20^{th} second to the 35^{th} second of each activity. Then subtractive clustering was performed on each observation group so as to calculate a data point that represents all of the 15 observations. In consequence, this representative point is tested with the concepts and, in Table 4.4, we have demonstrated some instances of these calculations:

Table4.4.: Similarity degree of a group of observations to their concepts

	Hand Washing	Coffee Making + Studying	Coffee Making + Hand Washing + Studying
Hand Washing sample	1.584921157717	0.417683824238	1.379359698881
Coffee Making + Studying sample	1.577486937353	1.406517318472	- 2.720760198171
Coffee Making + Hand Washing + Studying sample	1.583903960732	7.677138008818	1.509945321716

As it is shown in Table 4.4, these models confirm their concerning observations groups with a higher similarity degree.

4.5.1 Anomaly recognition and assistance provision in the Smart Home

The information provided from the activity recognition process is taken as input for the ARAPRS. In order to detect anomalies at the NWGF level, the concerning fuzzy states are used and the AFSO operator is performed on them. Because we can find different cluster centers for each influence range (IR) in the activity recognition step, thus we are able to identify different criteria in order to symmetrize concepts that have been learned. In this section, we show how we symmetrized these concepts in the concepts' hierarchy, level by level. In order to illustrate this, we considered a circle for each concept surrounded by circles representing lower-level concepts. For example, considering the concept hierarchy illustrated in Figure 4.7, we could draw a symmetric system that places these concepts facing each other. In order to symmetrize these concepts (considering both normal and abnormal concepts), we applied the AFSO operator and, in Figure 4.8, we can see a schema representing the symmetrized concepts. For anomaly recognition, per each concept, we can configure an SVM classifier in order to detect the concerning anomalies of that concept. At each level, if no SVM classifier outputs a normal state (no concept has been correctly realized), then it can be inferred that there is an anomaly at that level.

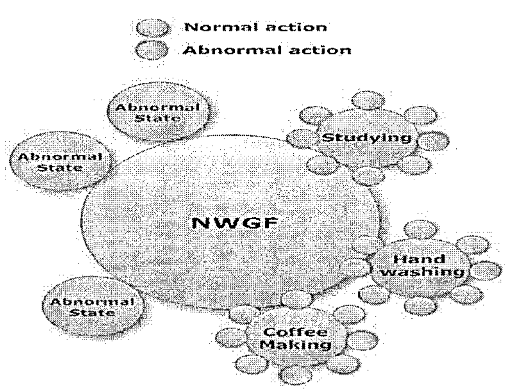


Figure 4.8.: Symmetrizing concepts around cluster centres

In Figure 4.8, we have illustrated a schema from the applied mechanism for recognition of concepts and anomalies. In the anomaly-recognition process, observations are mapped in the proposed virtual space, then considering the similarity degree to the known normal or abnormal concepts, world-state normality will be recognized. In order to locate precisely the *unrealized intended concept,* we would have to find the most similar scenario. In the last step, according to inferred similarity degrees, a proper guidance video or alarm is selected to be actuated.

4.5.2 Testing the system for normal and abnormal activities

An SVM classifier is applied in order to be trained with both normal and abnormal world fuzzy states. After that, the SVM is trained with the entire fuzzy cluster center set (members of the FTC set) and their symmetric values; it can be applied for reasoning about world state normality and to see if world state is generally normal. If for each concept an SVM is assigned, then we can also reason about the correct realization of each activity. In other words, not only we can infer the normality of world state, but also we can reason about the details of the anomaly. In LIARA, we benefit from warning lights, speakers and a TV, which may become active in order to assist and guide the Smart Home resident. Therefore, by finding the anomaly type, we can give the proper assistance and guide the resident who may have forget to accomplish some actions.

For implementing and testing the described support-vector machines, we benefited from MATLAB software and, in applying the *svmtrain* command, an SVM structure is formed and was trained with both the normal activities cluster centers and their symmetric values (as abnormal world states). The structure of the created SVM (for recognition of general world-state normality) is reported as follows:

```
         SupportVectors: [2x7 double]
                  Alpha: [2x1 double]
                   Bias: 2.9837e-016
         KernelFunction: @linear_kernel
     KernelFunctionArgs: {}
             GroupNames: [306x1 double]
   SupportVectorIndices: [2x1 double]
              ScaleData: [1x1 struct]
          FigureHandles: []
```

In figure 4.9, we have illustrated a plot of SVM classification which indicates the relation of the two variables to normal and abnormal states:

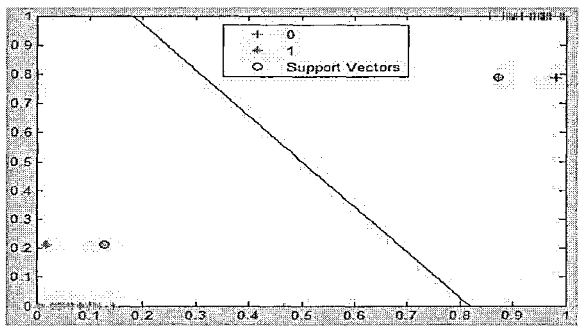


Figure 4.9.: Support vector machine for distinguishing normal and abnormal states

We can see that the two normal and abnormal world states are clearly isolated and distinguishable. Therefore, in order to test the structured SVM for recognition objectives, we applied the $svmclassify$ command in order to measure its ability to distinguish normal and abnormal states.

In order to validate the proposed approach, during the first step, we tested it with *nine* correct scenario realizations (three times for *coffee making* realization, once for *reading*, once for *hand washing* and four social combinations of the individual activities). Moreover, the approach was tested with an erroneous *coffee making* activity realization.

The reasoning system, which detects general anomalies at the NWGF level, confirmed all of the nine correct realizations, but the erroneous realization of the *coffee making* activity was also recognized as part of a normal world situation. The reason for this misrecognition is that at the NWGF level, the SVM concerning the general anomaly recognizer is trained with centers of relatively big clusters; therefore, the partial mistakes made during activity realization (forgetfulness of sugar) will not be recognizable in this level. In other words, at the NWGF level, only the relatively major mistakes, such as entering dangerous contexts, are recognizable. Therefore, it is not necessary to call the caregiver because of a partial mistake. Hence, this partial error in activity realization is recognized by the SVM assigned to the individual activities level. Among individual activity models, the activity of *coffee making* was selected as most possible (similar) *uncompleted intended* activity. In consequence, the concerning actuators (video tape and warning lights) of this activity are called into action.

In the second step, we tested the model with an erroneous realization of the simultaneous *coffee making / reading* activity. In this scenario, the Smart Home resident stopped realization of coffee-making half way through the action and began the activity of reading. The SVM assigned for anomaly recognition of this simultaneous activity recognized the concerning anomaly from the 175th second out of a total of 188 seconds of activity realization.

4.6 Comparison with other approaches

In this section, we aim to compare our proposed methods with other existing ones, especially approaches which deal with activity and anomaly recognition in Smart Home. The objective of this part of the thesis is to show the efficiency of the proposed work regarding current available approaches.

In this part of our research, we have taken several important existing works into account, such as [23, 49, 50, 51, 88], where each of the mentioned scientists follows a strategically different approach to activity recognition. This difference includes the manner of observation (vision or non-vision based), of machine learning (data-driven or knowledge-driven), of reasoning (probabilistic or possibilistic) and the objective pursued in activity recognition. Further explanation on this subject is that in some research, special actions are taken into account and the ongoing activity is guessed; however, in other approaches, activities are ranked based on inferred-possibility degrees. In some research, anomalies are objected instead of correct realization of activities. In the following part of this thesis, we will indicate several criteria for making comparisons between previously conducted researches.

4.6.1 Comparison criteria

Direct comparison of our work with that of others was not an easy task because we followed achievements of some unique goals in this thesis, which are not followed by other existing works. These goals were being data-driven in nature and independent from expert idea, recognizing correct realization of activities and performing reliable anomaly recognition. However, we can mention some secondary goals which were achieved for the first time in our work such as provision of a framework for automatic world actuation (automatic goal setting and automatic realization), data-driven verification of temporal features of activities and integration of multiple data types within a single model. Moreover, in an important contribution concerning temporal data mining, we proposed an algorithm which sums up a huge temporal data set precisely within a brief mathematical equation. Regardless of achieved objectives and the difficulty to directly compare our work with existing approaches, we can measure the differences between the approaches proposed here with other existing ones by considering the following parameters:

1. **Vision or non-vision based approach**: Generally, the observable data in the Smart Home can be classified in two groups of data types based or not on vision. The vision-based approaches include analysis of media data types such as implementation of image processing or voice recognition as a part of activity recognition. The non-vision-based approaches deal with non-media data types such as on-off sensors, RFID sensors or temperature. An important point to be aware of is that because we are prohibited from impeding the resident's privacy, we applied non-vision sensors for observation.

2. **Precision in reasoning**: Precision-in-reasoning approaches indicate the success rate of proposed algorithms concerning correct recognition. These point out the quantity of correctly-made decisions out of the all of the decisions made.

3. **Quantity of machine states**: Each model representing the perception of a real-world problem includes some machine states and the transitions between these machine states. Experimentally speaking, process complexity depends on the total number of machine states, especially if the approach applied is a probabilistic one.

4. **Temporal complexity in model training**: Temporal complexity is a parameter that indicates the time spent in a computer for performing the training or recognition task. Long processing depends highly on the quantity of observations, so although a computer may quickly process a small data set, processing time increases dramatically when the data set size increases.

5. **Quantity of required training tests**: In order to learn the conditions for transition between machine states, we trained the model through training data. The quantity of

required training tests for correct accomplishment of recognition is applied as a criterion for comparison between approaches. This parameter depends on architecture model and the quantity of machine states.

6. **Knowledge validation in case of partial changes to sensory networks**: Partial changes in the Smart Home environment may happen. For example, a sensor may be added to a previously elaborated sensory observation network. Through this parameter, we would indicate whether the previously learned knowledge would remain valid in case a sensor failed or would be added.

7. **Quantity of observed activity features**: By this factor, we indicate on what information sources the inferred knowledge stands. Moreover, it shows how many parameters are verified in order to infer anomalies. Application of more sensors leads to verification of more activities' features; therefore, this *may* lead to a more precise anomaly-recognition approach and more certainty in recognition of activities' being correctly realized.

8. **Consideration of flexibility in realization of activities for modeling and recognition**: Activities may be realized in different manners. They may begin at many different times and in various spaces, and they may end in non-definitive places. Consideration of this issue is a parameter that helps to recognize more realistic approaches.

9. **Ability in anomaly recognition for interruptive activities**: Humans may manage realization of more than a single activity. For example, while cooking she/he may talk

on the telephone. Provision of ability to judge correct realization of parallel activities is a feature that we verified when considering the approach.

10. **Ability in sorting of probable/possible ongoing activities**: Many times there is no certainty in definitively selecting an activity as the only candidate which describes the ongoing events. In this case, instead of selecting a candidate, we may select a group of candidates that explain the ongoing events. In order to construct a reliable reasoning system, we must sort the probable/possible hypotheses and explain the world by descriptions containing degrees of certainty.

11. **Ability in real-time reasoning**: Reaction to anomalies is a critical task. For example, when the Smart Home resident falls down, it is necessary to react quickly and properly. Several approaches require time and wait for more observations in order to make a decision. For example, some approaches wait for the end of an activity before recognizing it.

12. **Knowledge or data-driven machine learning**: Knowledge required of a reasoning system can be provided in two ways. In the first way, the expert transfers knowledge directly to the machine and, in the second way, knowledge is provided through data analysis and AI techniques. A data-driven approach provides a framework for design of automated Smart Homes, which can be customized for the residents; however, with a knowledge-driven approach, an expert can feed the model with sophisticated rules without spending time and energy on model training. A system which benefits from both model-training strategies could be a perfect approach for the Smart Home because on one hand, there are a lot of details in and manners of activity realization

which justifies a data-driven approach and, on the other hand, there are many cases for which expert knowledge is sufficient in order to prevent process complexity in machine learning.

4.6.2 Comparison results

As we have already mentioned, direct comparison with existing models is not an easy task. On one hand, much of the required information about others' works is not available and, on the other hand, the existing research was often conducted in different contexts, with different objectives and with very different formal approaches and constraints. Therefore, in order to compare our results with those of other researchers, we would *analyze* existing approaches by previously proposed parameters.

Five main functional steps of our approach were compared to other existing works; inasmuch as method is concerned, we compared each phase of our approach with other research step by step in order to specific and non-generic comparative results. The strategic steps of our approach are sensory observation, modeling and intention recognition, recognition of correct realization and anomaly recognition. For each phase, we considered a few important works and applied some of the previously mentioned factors for comparison.

4.6.2.1 Sensory observation

We applied over one hundred on/off sensors, RFID tags and temperature and light sensors. Considering that we had eight RFID antennas, so for each RFID tag we had eight states; therefore, we had a total of over five hundred data columns in our database. Each column represents an activity feature. The duration of activities was a minimum of one minute and a maximum of five minutes. Considering that for each second one observation was made, thus on average we had 75,000 data cells for each activity. We applied a non-vision based observation method in order to respect the privacy of the Smart Home resident.

The quantity of applied sensors in the CASAS Smart Home [110] is approximately 39 on/off sensors. Considering that we studied multi-state sensors and a greater quantity of on-off sensors in our approach, thus at the observation level it was highly compatible with other available approaches. In CASA, a typical activity is recorded through 40 data records, and we can infer that each data set in that Smart Home includes 1560 data cells; therefore, we have selected a compatible amount of data for processing.

In [88], five cameras observe activities. In order to achieve the optimal recognition rate, these cameras work in a codebook of 100; therefore, we can imagine that in this project, 500 activity features were observed. We evaluate our approach on the level of sensory observation as being compatible with this work; however, we worked with multiple sorts of sensors.

4.6.2.2 Modeling

Our approach is a data-driven one, which mines the data in order to be trained. Some data-driven approaches are the works of [29, 49].

In the work of Singla, for the activity of *hand washing*, 8 machine states are considered (see Figure 4.10), and in a world where four activities are learned, each activity is performed 22 times and an overall accuracy rate of 88.63% had resulted from this process. One important point in the probabilistic work of Singla is that in order to achieve normal distribution in transitions between machine states, it should provide at least 35 samples for each transition (edge in graph). In the best case, it requires 35 training tests and, in the worst case, it would need 35 in 18 (number of possible transitions) training samples. In other words, it is possible that probabilistic modeling of the *hand washing* activity requires at least 630 stochastic training tests. Moreover, an inaccuracy rate of 11% indicates that some unreliable decisions were made.

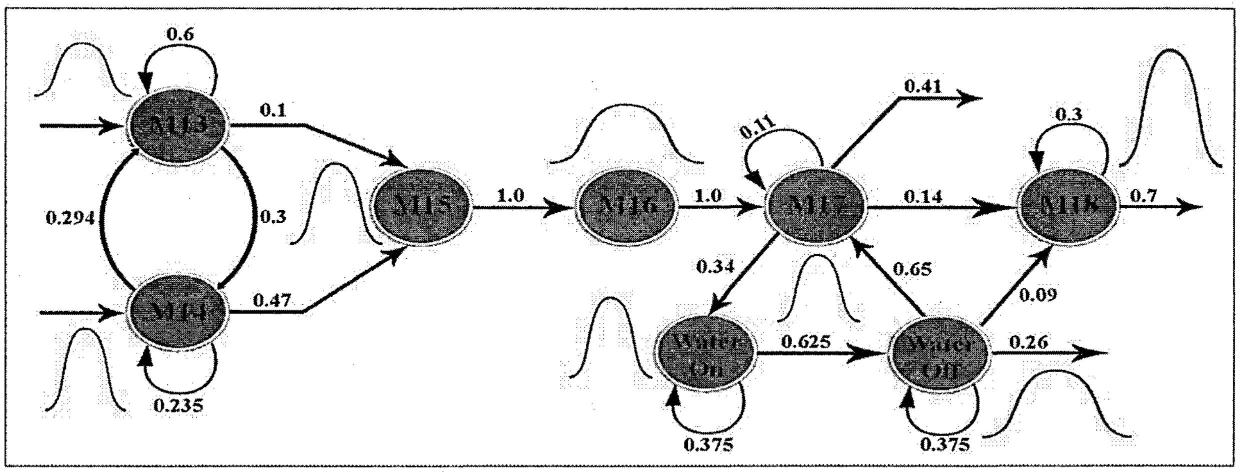


Figure 4.10.: Modeling of hand-washing activity in Singla's work [49]

In probabilistic approaches such as this one, an activity must start from its beginning states; for example, in Figure 4.10, the activity should begin by observation of M14 and M13 and end by M18 or water off. Now if for a reason of imprecision an action is not observed, or if the activity begins from further states, then it is not correctly recognized. The sensors are critical parts of the model and elimination or addition of sensors may cause dramatic invalidity of knowledge. In work presented by Biswas [31], we can see that in order to train a more accurate probabilistic model, it may take up to 72 hours. Accordingly, model training and provision of proper training tests is a challenge in probabilistic models [51].

In the probabilistic vision-based approach of Chen et al. [88], the expert must supervise the modeling process and estimate the optimal state for each model. Not only should the activities be performed in a rigid way, but the discovered knowledge is also valid for specific colors and light measures, which constitutes a constraint for this approach.

In our approach for example, during a typical activity - in this case the activity of *coffee making* - we calculated 22 fuzzy machine states by one-time model training. Uncertainty and imprecision were considered and the model was trained in only a few seconds. Sensor elimination and addition are tolerated here. This knowledge can be manipulated by the expert in a fuzzy knowledge base [12]. Because of the nature of our work, perfect accuracy was achieved.

Our proposed modeling approach is an innovative one which for the first time puts forwards consideration of fuzzy states for modeling activities. Because of the Closed World Assumption (CWA) in our work, any unknown observations are considered as anomalies; therefore, we expect inaccuracy in the case when an activity is realized in an unseen or previously unknown manner.

4.6.2.3 Intention recognition

In the work of Nazerfard [29], the temporal contexts in which activities may be realized in are predicted. An inaccuracy rate of over 10% for intention recognition is expected by the proposed probabilistic method. This approach does not recognize the correct realization of activities and it does not rank the probable explanative activities. It should be mentioned that well-trained probabilistic works always contain 5% of inaccuracy.

In the work proposed by Patrice Roy [51], approximately 70% is attained in terms of precision in intention recognition and possible hypotheses are ranked. Our presented approach is compatible with the aforementioned work; however, Roy has followed a knowledge-driven strategy. As it was demonstrated in previous sections, the proposed approach correctly ranked the most possible ongoing activities.

4.6.2.4 Recognition of correct realization

Recognition of correct realization of activities was one of the main characteristics of our project. A system which accurately ranks hypotheses and welcomes observation of several activity features may perform this task well. Moreover, recognition of contexts in which activities are realized is a critical issue in going about this task. The work of Roy [51] proposed introduction of erroneous activity realizations; however, we do not find this approach practical because, in contrast to normal realizations, abnormal realizations of an activity may be done indefinitely.

Our approach draws a fuzzy space for correct ways of realizing activities and finds suspect observations which are unfamiliar to already-acquired knowledge. We have dealt with this step for the first time in our realizations and have consequentially gotten practical results.

4.6.2.5 Anomaly recognition

Out of the rare works that deal with anomaly recognition in the Smart Home, we can point out the research of Jakkula [23, 44, 90]. In this work, all abnormal sensor-event series are considered as being anomalous. Although this work proposes an interesting approach, it basically considers proportion of normal episodes to improbable ones. Thus, any correct action that occurs would rarely be recognized as an anomaly. Furthermore, this study reveals a relatively high inaccuracy rate. In [29, 49, 88], the reasoning system requires that an activity be concluded in order to be recognized; therefore, they do not show any evidence of real-time reasoning.

Our approach proposes similarity of actions and activities to the normal realizations, so it recognizes an anomaly according to the nature of the actions and activities which are associated to it. Furthermore, we provided a framework for real-time reasoning in correct realization of activities.

4.7 Discussion on concept-driven recognition of activities

In this part of the thesis, we discuss some questions that may be asked about the proposed work. The first subject to this effect is that the beginning and ending points of an activity should be known in order to extract activity patterns; considering that data in a relatively large temporal data set in the Smart Home should be mined, how could we clarify these points in an automatic manner without the help of an expert? To answer of this question, we refer to the idea of general characteristics of an activity, which indicates that an activity is a set of fuzzy events that occur within short spans of time (and short spatial distances), in which "short" is a fuzzy term and regarding to the influence range of the defined fuzzy classes it varies. According to this principal, we do not need to know the exact beginning or ending points of activities; however, approximate position of these points can be calculated if members of the FTC matrix are sorted by time.

A second question that one may ask is how can we use this model for long time anomaly recognition (for example, 24 hours) considering that activity patterns indicate behavior of sensors in activity realization for a relatively short period of time (for example, three minutes) Our response is that we can calculate the similarity of live observations to models regardless of time and would infer the similarity of a current observation to special moment(s) of a model and predict future events by these presumed hypotheses. Moreover, we have discussed that we can estimate a special IR rate in which each activity is put in a cluster. Thus, whenever a series of fast fuzzy events occur, an activity (individual or simultaneous) is realized.

The third subject refers to the question that one may ask about dependency of activities to real (daily) time and duration. In other words, how can we include both of the aforementioned data types in an integrated model? For example, the activity of eating lunch may begin at approximately noon and continue for 20 minutes; at recognition time, how can we reason in both of the mentioned features? When answering this question, we refer to the nature of the clustering method (subtractive clustering) on observed data in which cluster centers are estimated by considering both absolute values (daily time) and relative values (duration). In other words, by the applied cluster-estimation method, the cluster representatives are selected regarding distances between data points (duration) and their aggregation in some absolute positions (daily time). Therefore, approximate daily beginning and ending times of this activity can be inferred as some cluster centers and activity duration is already hidden between the beginning and ending values.

The last discussion refers to spatial reasoning and possible ways that we can model activities regarding their spatial features. In other words, how can we propose a similar approach for modeling activities as series of fuzzy events which may occur over short distances? We propose resorting to an observation matrix based on a proper spatial variable and performing the clustering operation regarding this spatial parameter as an answer to this question. It is expected that world transitions are indicated based on displacements regarding this variable.

4.8 Conclusion concerning validation and experiments

In this chapter, we presented our validation efforts and the experimental results for the contribution proposed in this thesis. First, we have briefly described the validation context of our works, which took place in the infrastructure of the LIARA laboratory. Thereafter, we presented the objectives of the validation process; we explained the methodology of validation and we described the programming environments and software implementation. As mentioned, the validation approach was mainly based on simulation of representative scenarios of common daily activities in the Smart Home infrastructure and then analyzing the data using MATLAB software in order to test proposed equations. Finally, we presented results of the experimentation conducted using the proposed model and gave some statistical information, followed by a comparison with other existing approaches. In summary, we can say that the proposed model gave promising results that can be considered significant advancement concerning several aspects of the learning and recognition process in a smart environment, as seen in the comparative section. It can also be noted that the proposed model, validation and results have led to several scientific publications [7-13, 46, 92, 109, 111], showing the significance of our work and the proposed ideas. Results obtained were well received by the community of peers working in the field.

5 General conclusion

In the proposed thesis, we presented a contribution to the field of artificial intelligence, following in the footsteps of temporal data-mining models and activity-recognition approaches. It allows us to take a step forward by providing answers to the issues raised in the introduction, which are related to providing a way to learn and recognize the activities of daily living in a Smart Home in order to be able to assist cognitively-impaired elders. More specifically, the thesis contribution takes the form of a new unsupervised temporal data-mining model for activity recognition, which allows anomaly detection and assistance provision in the Smart Home. In this model, the role of the expert in modeling, reasoning, data interpretation, inference making and recognition steps is decreased as much as possible; therefore, the proposed approaches of this thesis shed light in some directions for designing stand-alone Smart Homes. Therefore, it can be seen as a step toward the development of an assistive environment which is able to adapt itself to the resident's profile and enable the possibility of independent living with respect to privacy through non-audiovisual observation and minimal need for an expert's intervention.

5.1 Realization of fixed objectives

As described in detail in Chapters 2, 3 and 4, the contribution of this thesis is conceptual, practical and experimental. This contribution was achieved with four research objectives cited in the introduction following scientific methodology: (i) literature review, (ii) formalization of the conceptual contribution, (iii) implementation/validation and, finally, (iv) experiment and comparative analysis of results. We will now summarize the completion of each objective.

5.1.1 First objective: literature review

The first objective, presented in detail in Chapter 2, was to review existing works of the literature related to ambient intelligence, data mining and, more specifically, temporal data mining for activity recognition in Smart Homes. Each related work which was presented has been examined considering contextual challenges, existing advantages and its limitations, in order to position our contribution. Our goal at first was to acquire general knowledge about the field of research by reviewing key books [22, 25, 48] and works [21, 49, 50] on the subject. The second objective consisted of a much more targeted review of fundamental and applied works [51-53] related to the specific problem of temporal data mining in the Smart Home; these were conducted for activity recognition. In this phase, some useful material about the temporal data mining [25], probability theory [112-114], fuzzy logic [66-68] and description logic [96] was studied about reasoning systems, inference-making systems and data interpretation methods. Finally, investigation of why a temporal data-driven approach for activity recognition should be proposed concluded this part of the thesis. This first step served as a basis for elaborating our conceptual contribution.

5.1.2 Second objective: formalization of the new model

The objective was, in the light of the conclusions drawn from the literature review, to propose a new model for a qualitative temporal data-driven modeling approach. At this stage, the main strategies, goals and directions for accomplishment of our thesis's objectives had been identified. Data-driven machine learning from temporal data (i.e. in order to bring automation into modeling), automatic reasoning systems, and reliable inference making and respecting the resident's privacy had been identified as the main characteristics of an approach that considered desirable in this case. Hence, to achieve this objective, we developed a new formal model inspired by the advantages and weak points discovered while conducting the literature review. As pointed out in previous works related to Smart Home technology, temporal information constitutes a key element for correctly identifying activities of daily living, but only a few works [23, 32, 45] made use of this kind of temporal data in the mining process, and most of them only relied on Allen's temporal framework [5], which is very limited. Therefore, we proposed a formal framework which was adapted to our specific context: temporal data mining used in order to recognize activities. As a novelty, we also introduced fuzzy logic and, more specifically, the notion of fuzzy time in reasoning, which allows the taking into account of imprecision in activity patterns and recognition of activities. We modeled these activities and formalized our conceptual ideas and articulated them into a complete model. We then provided a framework for automatic reasoning during simultaneous activities.

5.1.3 Third objective: validation

The third objective was to validate the proposed model in a representative context in order to assess it efficiency. To achieve this, we used Microsoft Visual Studio .Net and an SQL server; then we implemented the formal model presented in Chapter 3 using MATLAB [47] software in order to simulate the equations of the proposed model. The objective realization was presented in chapter 4. The developed application includes a fuzzy-inference system, a subtractive-clustering algorithm, regression and classification processes to interpret and make inferences about observations and a few software tools. MATLAB was chosen as the main programming environment because it allows transforming the format of initial primary data (from sensors) into matrix format, and also in order to benefit from MATLAB matrix operators to extract knowledge; this enables us to apply and represent knowledge in various forms. Modeling, mathematical operations, recognition and inference-making procedures proposed in this thesis have been simulated with this environment. In the Appendix section of our thesis, we have mentioned some applied-code examples. In our simulation, we used data observed in the Smart Home infrastructure of the LIARA laboratory from various sensors such as radio-frequency identification tags, infrared motion sensors, electromagnetic contacts, etc.

5.1.4 Fourth objective: experiment and comparative analysis of the results

The last objective consisted in rigorously testing the model proposed in this thesis by conducting a series of experiments, which took the form of several representative case studies. Consequently to our objectives, we selected representative activities of daily living which were performed in the LIARA infrastructure. These activities were modeled and the developed application tried to learn and recognize these. We then completed a statistical analysis of the obtained results which we compared to the ones presented in former works. These results were very promising and were presented in detail in Chapter 4.

5.2 Review of the proposed approach

The goal of the proposed data-mining approach was to summarize a relatively large temporal data set and extract applicable knowledge from it. The inferred knowledge was then applied and used for activity recognition, anomaly recognition and assistance provision. Following these steps, we proposed a model composed of two parts: (i) knowledge inference (activity learning and modeling) and (ii) knowledge application (recognition and decision making). Mathematical operations and methods were developed to automatically perform the aforementioned processes.

5.2.1 The knowledge-inference process

During the first step of this process, a fuzzy space for realization of activities is formalized. This logical space indicates the behavior of the sensors data in activity realization. In order to calculate this space, the temporal data of the Smart Home is fuzzified and fuzzy rules explaining the observations made during activity realization are inferred. This information is then processed and the supplementary information, used for explaining unobserved world states, is inferred. Trying to estimate sensor behavior during possible simultaneous realization of activities then led us to draw a knowledge base which could explain any possible normal world state (even if it is never observed). We also proposed a pyramidal schema for calculation of this proper knowledge base for Smart Home.

5.2.2 Recognition and decision-making process

The inferred similarity degree to known and estimated normal pattern of activity was the main judgment criteria for activity recognition, anomaly recognition and assistance provision systems. A fuzzy-inference system and fuzzy-logic-based clustering approach were developed for validating this type of reasoning system.

Anomalies are recognized whenever no normal activity pattern confirms a similarity degree to current observations and an abnormal pattern does confirm a similarity degree to that observation. In order to implement this type of recognition system, we trained a support-vector machine with some fuzzy numbers that represent the fuzzy rules which explain normal and abnormal realization patterns of activities.

Whenever an anomaly is recognized, proper actions or activities that enable the return of the Smart Home to a normal state are called. In order to calculate the proper reaction of the Smart

Home, every action learned and each activity are symmetrized around normal world states; then, when an anomaly occurs, the recovery action or activity which is the symmetric concept of the anomaly position is selected as the proper reaction of the Smart Home to the anomaly; however, because we have no robotic or invasive actuators in the Smart Home, we rely on the diffusion of multimedia guidance in the Smart Home. Once again, in the assistance provision step, we applied fuzzy logic and fuzzy-logic-based clustering to conceptualize symmetrization in the Smart Home.

5.2.3 Applied innovative mathematical techniques

In this thesis, we proposed the application of some innovative mathematical operations on temporal data. Smoothing of fuzzy numbers through smoothing techniques, fuzzy summarization and data-driven definition of fuzzy time are some of the innovative mathematical operations that we proposed. Our contribution to data mining is the proposal of a data-driven learning system and that of mathematical equations used for the modeling of a temporal data set. Moreover, we proposed a framework so as to be able to recognize simultaneous activities. Furthermore, we presented an innovative method for anomaly recognition, which is capable of reasoning on the occasions when simultaneous activities are interrupted.

5.2.4 Efficiency and practicality

Efficiency and practicality of the proposed approaches were verified and tested. Our empirical case studies showed low-level process complexity because the methods that were used were designed to consider input data or information only once; thus, it is prevented from iteration. The experimental results described in Chapter 4 showed reliable precision in recognition during both individual and simultaneous activity realization. These were analyzed and compared with other approaches.

The results of the research presented in this thesis can be extended to be applied to some other fields of study. In [115], we have dealt with the problem of network coding and extended the results of the research performed in LIARA to contribute to this newly justified problem. In [109], we proposed some solutions for prediction of social networks. In [92], we proposed some models for assisting the Smart Home resident in order to facilitate his or her activity realization. In [13], we proposed the idea of relative parallelism and application of data-driven fuzzy-time discovery, which may be applied in parallel processing.

5.3 Limitations of the proposed approach

Despite the promising results obtained by testing various ADLs during the experimental phase, the proposed approach faces some limitations. The first kind of limitation refers to restrictions imposed by today's technology and disability of some sensors in precise estimation of attribute states, so full certainty of correct accomplishment of many actions could not be attained. For example, when the Smart Home resident opens the water tap and moves a glass under it, we cannot be sure if the glass is full of water or empty. Another instance that we can quote as an example of this phenomenon is that we cannot be sure that dishes are well cleaned or not. Regardless of inaccurately sourced information, we faced some limitations with existing actuators. In order to automatically assist the Smart Home resident in an emergency, the presence of flashing lights or the showing of a movie from a television set are insufficient. If the Smart Home resident is old, because of difficulties in action accomplishment even in non-emergency cases, the system will have nothing to do. Therefore, the best software for an intelligent-activity recognition approach could not accomplish anything great if it were disabled from actuating the physical environment at required moments.

The second sort of limitation refers to disability in online learning. The proposed approach learns activity models at first; then, it uses the knowledge learned at runtime. We did not propose an online learning technique because we cannot yet distinguish between an *anomaly* and a *new manner of correct activity realization.* A possible solution to this limitation is to begin by describing a border and definition for tolerable anomalies; then, any unfamiliar patterns which do not cross that imaginary border could be taken into account as a new correct activity. This task would require a new definition concerning normality and the proposal of a new creative system.

5.4 Future research

A brief proposal for future research would be one of working on some approaches which model behavior of an Alzheimer's patient. In this way, we could discover the parameters (activity features or world attributes) that affect perceived behavior. For example, we could discover how the patient's mind arranges goal achievement steps; on what critical steps he fails correct realization of activities and, regardless of analytical works, we could propose a customized system for assisting such a patient. This research would mainly target Alzheimer's disease and the result could be discovery of more knowledge concerning this pervasive illness.

Further explanation is that by modeling the behavior of the Alzheimer's patient, we could estimate which world attribute indicates non-linear behavior within expected behavior. Then we would try to find a justification for the observed disorder in five layers which are described in more detail in the following sections.

5.4.1 Data level

At this level, we would try to justify the perceived behavior of the patient by evidence that we would discover during a data sorting step. Furthermore, , observations are sorted according to time in the proposed temporal data mining model, but if the data set was sorted by other parameters such as spatial information relating to an object, then we would face a spatial data-mining problem that could be solved similarly to the methodology employed in the current thesis. As the result, we could model the patient's mind according to a special parameter that justifies best his or her behavior.

5.4.2 Concept-making level

Similarly to the methodology that we described for patience modeling at the data-level, we can analyze behavior of the Alzheimer's patient at the concept-making level. Considering that each concept is represented by a special cluster out of a number of possible recognizable clusters, then we can analyze if the patient's mind can make well the temporal concepts. If we could justify a patient at this level, then we could plan customized assistance such as reminders and guides (in multimedia format for example) that show correct realization way of activities.

5.4.3 Knowledge hierarchy

Sometimes, although a person can correctly perform some actions, he or she may not respect correct accomplishment order. In fact, one possible reason that could justify this behavior refers to the priority of actions or knowledge hierarchy in his or her mind. By modeling patient behavior and discovering disorder level, we could plan in order to provide customized assistance - such.as robotic assistance - for such a disorder.

5.4.4 Anomaly recognition

A possible problem that could be modeled is the anomaly recognition level. For example, when an anomaly occurs, the patient often does not know what the best reaction to it is. For example, he or she does not know when it is cold, but the person should heat the environment. Another example would be the action of of washing dishes after eating lunch. For this problem, we could plan customized assistance through automatic object symmetrization for the patient.

5.4.5 Recovery plan

Sometimes, even though the Smart Home resident knows appropriate simple reactions to simple anomalies, he or she may not be able to estimate a sophisticated recovery plan in order to escape the anomaly. This can be a difficult task even for a healthy man or woman. According to this thesis, we can infer that a recovery plan is a hierarchy of concepts that are symmetrical to an actual anomaly. A robot can be assigned the recovery plan in order to assist the Smart Home resident.

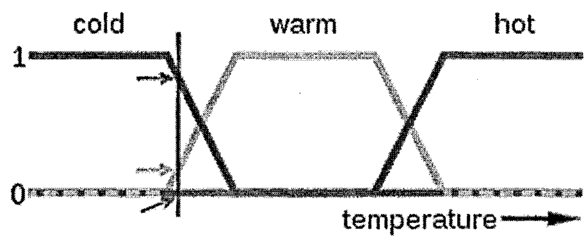
We hope that by this proposal, the possible disorders with the Alzheimer's patient are discovered and new ways of understanding this disease and possible treatments for this illness are discovered. One other idea for future researchers is the proposal of statistical analysis on world attributes and activity functions in order to discover linear and non-linear relations between concepts and world attributes. One result of risk analysis is an increase in precision in the activity recognition system. Moreover, we could provide statistical information for each concept as well as a framework for risk analysis in the Smart Home. This risk analysis would lead to efficient energy management and robotic assistance provision for improvement in realization of activities.

6 Appendix A: Fuzzy logic

Fuzzy logic is a form of many-valued probabilistic logic; it deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values) fuzzy-logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions. Classical logic only permits propositions having a value of truth or falsity. The notion of whether 1+1=2 is absolute, immutable, mathematical truth. However, there exist certain propositions with variable answers, such as asking various people to identify a color. The notion of truth doesn't fall by the wayside, but instead becomes a means of representing and reasoning when partial knowledge is afforded by aggregating all possible outcomes into a dimensional spectrum.

Both degrees of truth and probability range between 0 and 1; hence, they may seem similar at first. For example, let a 100 ml glass contain 30 ml of water. Then we may consider two concepts: “empty” and “full”. The meaning of each of these can be represented by a certain fuzzy set. One might then define the glass as being 0.7 empty and 0.3 full. Note that the concept of emptiness would be subjective and thus would depend on the observer or designer. Another designer might design a set membership function equally well where the glass would be considered full for all values above 50 ml. It is essential to realize that fuzzy logic uses truth degrees as a mathematical model of the vagueness phenomenon, while probability is a mathematical model of ignorance.

A basic application might characterize sub-ranges of a continuous variable. For instance, a temperature measurement for anti-lock brakes might have several separate membership functions defining particular temperature ranges needed to control the brakes properly. Each function maps the same temperature value to a truth value in the 0 to 1 range. These truth values can then be used to determine how the brakes should be controlled.



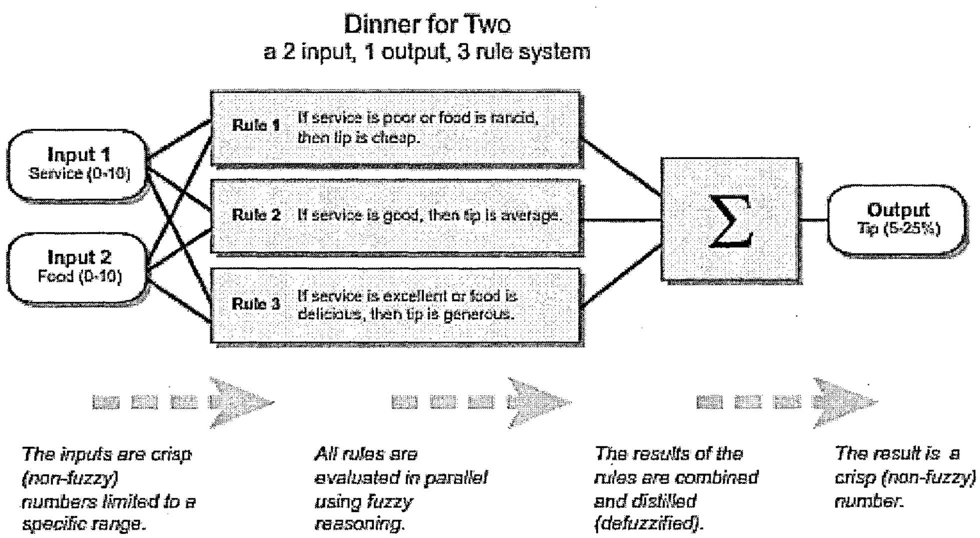
In this image, the meanings of the expressions *cold*, *warm* and *hot* are represented by functions mapping a temperature scale. A point on that scale has three "truth values", one for each of the three functions. The vertical line in the image represents a particular temperature that the three arrows (truth values) gauge. Since the red arrow points to zero, this temperature may be interpreted as "not hot". The orange arrow (pointing at 0.2) may describe it as "slightly warm" and the blue arrow (pointing at 0.8) "fairly cold".

While variables in mathematics usually take numerical values, in fuzzy-logic applications, the non-numeric *linguistic variables* are often used to facilitate the expression of rules and facts. A linguistic variable such as *age* may have a value such as *young* or its antonym *old.* However, the great utility of linguistic variables is that they can be modified via linguistic hedges applied to primary terms. The linguistic hedges can be associated with certain functions.

6.1 Fuzzy Inference Process

Fuzzy inference is the process of formulating a map from a given input to an output by using fuzzy logic. This mapping then provides a basis from which decisions can be made or patterns discerned. The process of fuzzy inference involves all of the pieces that are described in membership functions, logical operations and if-then rules.

This section describes the fuzzy inference process and uses the example of the two-input, one-output, three-rule tipping problem. The basic structure of this example is shown in the following diagram:

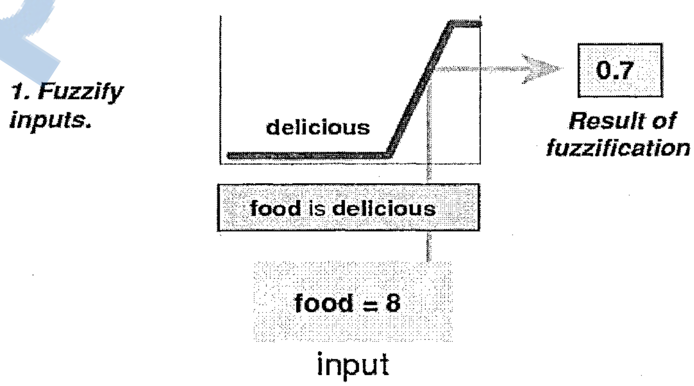


Information flows from left to right, from two inputs to a single output. The parallel nature of the rules is one of the more important aspects of fuzzy-logic systems. Instead of sharp switching between modes based on breakpoints, logic flows smoothly from regions where the system's behavior is dominated by either one rule or another. The fuzzy inference process comprises of five parts: *Fuzzify Inputs, Apply Fuzzy Operator, Apply Implication Method, Aggregate All Outputs, Defuzzify.*

6.1.1 Fuzzify inputs as first step (step 1)

The first step is to take the input and determine the degree to which it belongs in each of the appropriate fuzzy sets via membership functions. The input is always a crisp numerical value limited to the universe of discourse of the input variable (in this case, the interval between 0 and 10) and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1). Fuzzification of the input amounts to either a table lookup or a function evaluation.

This example is built on three rules, and each of the rules depends on resolving the input into a number of different fuzzy linguistic sets: service is poor, service is good, food is rancid, food is delicious and so on. Before the rules can be evaluated, input must be fuzzified according to each of these linguistic sets. For example, to what extent is the food really delicious? The following figure shows how well the food at the hypothetical restaurant (rated on a scale of 0 to 10) qualifies, via its membership function, as the linguistic variable *delicious*. In this case, we rated the food as an 8, which, given your graphical definition of delicious, corresponds to $\mu = 0.7$ for the delicious membership function. In this manner, each input is fuzzified over all the qualifying membership functions required by the rules.

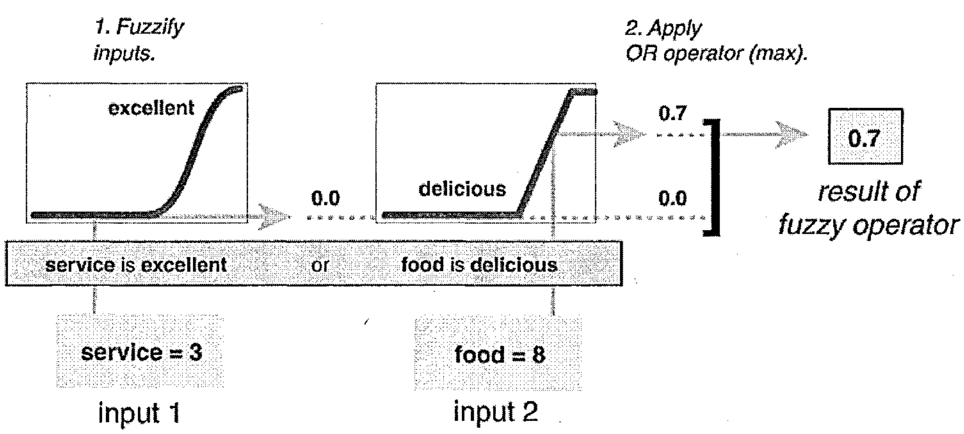


6.1.2 Apply Fuzzy Operator as step 2

After the input is fuzzified, you know the degree to which each part of the antecedent is satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain a number that represents the result of the antecedent for that rule. This number is then applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value.

By using logical operations, any number of well-defined methods can fill in for the AND operation or the OR operations, which are respectively supported by: *min* (minimum) and *prod* (product) and *max* (maximum) and the probabilistic OR method, also known as the algebraic sum that is calculated according to the equation: $probor(a,b) = a + b - ab$

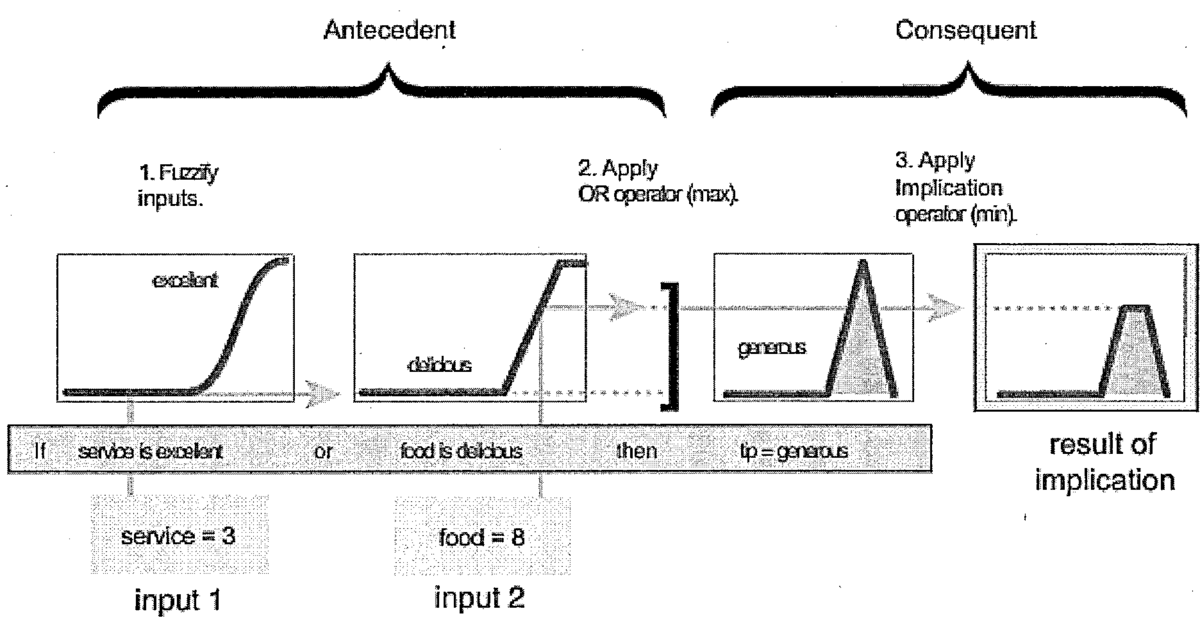
The following figure shows the OR operator *max* at work, evaluating the antecedent of rule 3 for the tipping calculation. The two different parts of the antecedent (service is excellent and food is delicious) yielded the fuzzy membership values of 0.0 and 0.7 respectively. The fuzzy OR operator simply selects the maximum of the two values, 0.7, and the fuzzy operation for rule 3 is complete. The probabilistic OR method would still result in 0.7.



6.1.3 Apply Implication Method as step 3

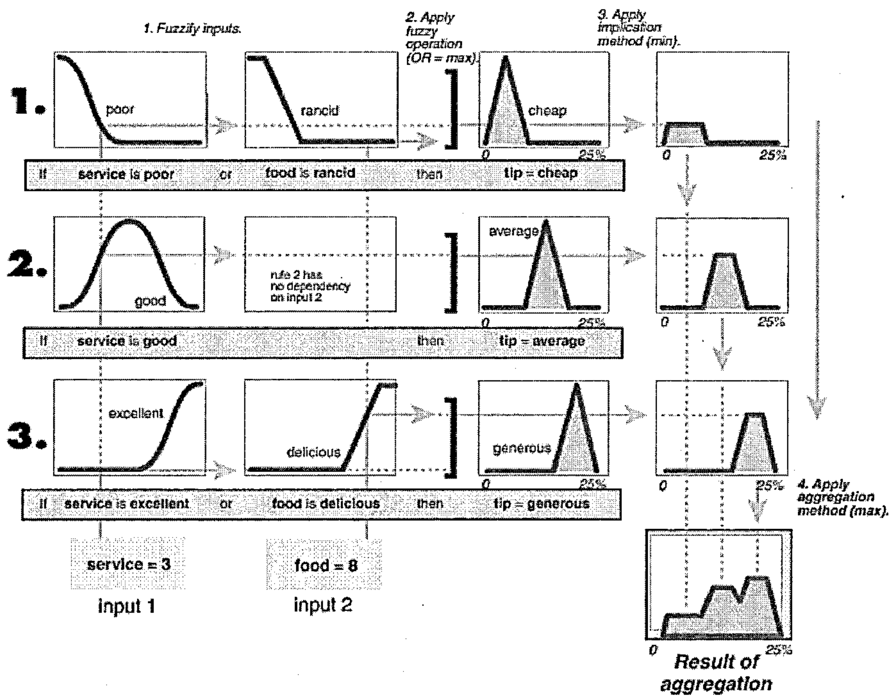
Before applying the implication method, you must determine the rule's weight. Every rule has a *weight* (a number between 0 and 1), which is applied to the number given by the antecedent. Generally, this weight is 1 (as it is for this example) and thus has no effect at all on the implication process. From time to time, you may want to weight one rule relatively to the others by changing its weight value to something other than 1.

After proper weighting has been assigned to each rule, the implication method is implemented. A consequent is a fuzzy set represented by a membership function, which appropriately weights the linguistic characteristics that are attributed to it. The consequent is reshaped by using a function associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent and the output is a fuzzy set. Implication is implemented for each rule. Two methods are supported, and they are the same functions that are used by the AND method: *min* (minimum), which truncates the output fuzzy set, and *prod* (product), which scales the output fuzzy set.



6.1.4 Aggregate All Outputs as step 4

Because decisions are based on the testing of all of the rules in a FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the output of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable. As long as the aggregation method is commutative (which it always should be), then the order in which the rules are executed is unimportant. Three methods are supported: *max*(maximum), *probor*(probabilistic OR) and *sum* (simply the sum of each rule's output set). In the following diagram, all three rules have been placed together to show how the output of each rule is combined, or aggregated, into a single fuzzy set whose membership function assigns a weighting for every output (tip) value.



6.1.5 Defuzzification as step 5

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. Perhaps the most popular defuzzification method is that of centroid calculation, which returns the centre of area under the curve. There are five built-in methods which are supported: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum and smallest of maximum.

7 Appendix B: some samples of applied MATLAB

In the current appendix, we present some MATLAB sample codes applied in the implementation of the proposed activity recognition model:

Fuzzy-inference system

```
[num,txt,raw] = xlsread('SmartHomeData.xls','mytable')
[num2,txt2,raw2] = xlsread('SmartHomeData.xls','mytable2')
[num3,txt3,raw3] = xlsread('SmartHomeData.xls','mytable3')
num(:,1)=[]
num2(:,1)=[]
num3(:,1)=[]
data=[num, num2, num3]
quantity =size(data,1)

for i=1:quantity
output(i,1)=1;
end

fismat = genfis2(data,output,0.5)
```

Subtractive clustering

```
path('C:/folder',path)
[num,txt,raw] = xlsread('SmartHomeData.xls','mytable');
[num2,txt2,raw2] = xlsread('SmartHomeData.xls','mytable2');
[num3,txt3,raw3] = xlsread('SmartHomeData.xls','mytable3');
num(:,1)=[];
num2(:,1)=[];
num3(:,1)=[];
data=[num, num2, num3];
d1=data(:,109);
d2=data(:,101);
d3=data(:,93);
data2=[d1, d2, d3];
[ClusterCentres,S] = subclust(data2,0.5);
```

Regression

```
Y = regress(FuzzyRank,RolePlaying)
```

Smoothing

```
yy = smooth(FuzzyRank,RolePlaying)
```

method	Description
'moving'	Moving average (default). A lowpass filter with filter coefficients equal to the reciprocal of the span.
'lowess'	Local regression using weighted linear least squares and a 1st degree polynomial model
'loess'	Local regression using weighted linear least squares and a 2nd degree polynomial model
'sgolay'	Savitzky-Golay filter. A generalized moving average with filter coefficients determined by an unweighted linear least-squares regression and a polynomial model of specified degree (default is 2). The method can accept nonuniform predictor data.
'rlowess'	A robust version of 'lowess' that assigns lower weight to outliers in the regression. The method assigns zero weight to data outside six mean absolute deviations.
'rloess'	A robust version of 'loess' that assigns lower weight to outliers in the regression. The method assigns zero weight to data outside six mean absolute deviations.

8 References

1. Alzheimer's, A., *2012 Alzheimer's disease facts and figures.* Alzheimers Dement, 2012. **8**(2): p. 131-168.
2. Liara. *Information available from: http://liara.uqac.ca.*
3. DOMUS, *Information available from: http://domus.usherbrooke.ca/.*
4. MavHome, *Information avilable from: http://ailab.wsu.edu/mavhome/.*
5. Allen, J.F., *Maintaining knowledge about temporal intervals*, in *Communications of the ACM*1983, ACM Press. p. 11.
6. Zadeh, L.A., *Probability Measures of Fuzzy Events.* Journal of Mathematical Analysis and Applications, 1968. **23**(2): p. 421-&.
7. Amirjavid, F., A. Bouzouane and B. Bouchard, *Trace of Objects to Retrieve Prediction Patterns of Activities in Smart Homes.* International Conference on Conceptual Structures for Discovering Knowledge (ICCS 2011), 2011: p. 353-357. URL: http://link.springer.com/chapter/310.1007%1002F1978-1003-1642-22688-22685_22630.
8. Amirjavid, F., A. Bouzouane and B. Bouchard, *Modeling of activities as fuzzy temporal multivariable problems*, in *Workshop on Space, Time and Ambient Intelligence 2013 (STAMI)*, D.J. Cook, Editor 2013, AAAI: Bellevue, Washington USA. p. 1-7. Current status: Accepted and under publication.
9. Amirjavid, F., A. Bouzouane and B. Bouchard, *Spatiotemporal Knowledge Representation and Reasoning under Uncertainty for Action Recognition in Smart Homes*, in *AAAI Spring Symposium on AI and Health communication* 2011, AAAI: Stanford. p. poster.
10. Amirjavid, F., A. Bouzouane and B. Bouchard, *Spatiotemporal knowledge representation and reasoning under uncertainty for action recognition in Smart Homes.* Proceedings of the 22nd Midwest Artificial Intelligence and Cognitive Science Conference (MAICS'11), 2011: p. 188-194. URL: http://ceur-ws.org/Vol-710/paper183.pdf.
11. Amirjavid, F. et al., *Recognition of fuzzy contexts from temporal data under uncertainty case study: activity recognition in Smart Homes* in *International Workshop on Data Integration and Mining (DIM-2012)*2012, IEEE: Las Vegas, NV. p. 231 - 238. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6303015&url=http%6303013A%6303012F%6303012Fieeexplore.ieee.org%6303012Fiel6303015%6303012F6294629%6303012F6302564%6303012F06303015.pdf%06303013Farnumber%06303013D06303015.
12. Amirjavid, F., A. Bouzouane and B. Bouchard, *Activity modeling under uncertainty by trace of objects in Smart Homes.* Journal of Ambient Intelligence and Humanized Computing, 2012: p. 1-9. URL: http://link.springer.com/article/10.1007/s12652-12012-10156-12655.
13. Amirjavid, F., A. Bouzouane and B. Bouchard, *Data-driven modeling of the simultaneous activities in ambient environments case study: automatic data-driven anomaly recognition for interruptive activities in Smart Home.* Journal of Ambient Intelligence and Humanized Computing, 2013: p. 1-24. Current status: Accepted and under publication.

14. Nations, U., *World Population Ageing: 1950-2050*, 2002, United Nations: United Nations.
15. Association, A.P., *Diagnostic and statistical manual of mental disorders*. fourth ed. 2000, Virginia: American Psychiatric Association. 992.
16. Wherton, J.P. and A.F. Monk, *Technological opportunities for supporting people with dementia who are living at home.* International Journal of Human-Computer Studies, 2008. **66**(8): p. 571-586.
17. Potkin, S.G., *The ABC of Alzheimer's disease: ADL and improving day-to-day functioning of patients.* Int Psychogeriatr, 2002. **14 Suppl 1**: p. 7-26.
18. Gustavsson, A., et al., *Predictors of costs of care in Alzheimer's disease: a multinational sample of 1222 patients.* Alzheimers Dement, 2011. **7**(3): p. 318-327.
19. Bouchard K., BouchardB. and Bouzouane A, *Guideline to Efficient Smart Home Design for Rapid AI Prototyping: A Case Study*, in *5th ACM International Conference on Pervasive Technologies Related to Assistive Environments (PETRA'12)*2012, ACM: Crete Island. p. 1-8.
20. Abdulrazak, B.G., S.; Bouchard, B.; Pigot, H.; Mokhtari, M. *Toward Useful Services for Elderly and People with Disabilities.* in *th International Conference on Smart Homes and Health Telematics*. 2011. Montreal, Canada: Springer.
21. Bouchard, B., S. Giroux and A. Bouzouane, *A keyhole-plan recognition model for Alzheimer's patients: First results.* Applied Artificial Intelligence, 2007. **21**(7): p. 623-658.
22. Ramos, C. and J.C.S. Augusto, Daniel *Ambient Intelligence: the Next Step for Artificial Intelligence*, in *IEEE Intelligent Systems* 2008, IEEE. p. 15-18.
23. Jakkula, V., D.J. Cook and A.S. Crandall. *Temporal pattern discovery for anomaly detection in a smart home* in *3rd IET International Conference on Intelligent Environments (IE'07)*. 2007. Ulm: IEEE.
24. Witten, I.H., Eibe, F. and Hall, M. A, *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems)*. 3rd ed. 2011: Morgan Kaufmann.
25. Mitsa, T., *Temporal data mining*. 2010: Chapman & Hall.
26. Corporation, T.C., *Introduction to Data Mining and Knowledge Discovery.* 3 ed. 1999: Two Crows. 36.
27. Frawley, W.J., G. Piatetskyshapiro and C.J. Matheus, *Knowledge Discovery in Databases - an Overview.* Ai Magazine, 1992. **13**(3): p. 57-70.
28. Michael. Steinbach, V., Kumar, *Introduction to Data Mining*. Pearson international Edition ed. 2006: Addison Wesley. 769
29. Nazerfard, E., P. Rashidi and D.J. Cook. *Discovering temporal features and relations of activity patterns*. in *International conference on data mining workshops*. 2010.
30. Rashidi, P. and D.J. Cook, *Mining and monitoring patterns of daily routines for assisted living in real world settings*, in *1st ACM International Health Informatics Symposium* 2010, ACM. p. 336-345.
31. Biswas, J. *Perspectives in elder care with technology.* in *9th Int. Conference on Smart Homes and Health Telematics (ICOST'11)*. 2011. Montreal: Springer.
32. Mohamed Tarik Moutacalli, Bouchard B. and Bouzouane A.: , *Unsupervised Activity Recognition Using Temporal Data Mining*, in *International Conference on Smart*

*Systems, Devices and Technologies (SMART 2012)*2012, IARIA: Stuttgart, Germany. p. 1-8.
33. Inomata, T. et al., *Activity recognition from interactions with objects using dynamic Bayesian network*, in *3rd ACM International Workshop on Context-Awareness for Self-Managing Systems* 2009, ACM. p. 39-42.
34. Wadley, V.G. et al., *Mild cognitive impairment and everyday function: evidence of reduced speed in performing instrumental activities of daily living.* Am J Geriatr Psychiatry, 2008. **16**(5): p. 416-424.
35. Roddick, J.F. and M. Spiliopoulou, *A bibliography of temporal, spatial and spatio-temporal data-mining research*, in *ACM SIGKDD*1999, ACM: New York, NY, USA p. 34-38
36. Philipose, M. and K.P.P. Fishkin, M.; Patterson, D.J.; Fox, D.; Kautz, H.; Hahnel, D., *Inferring activities from interactions with objects*, in *Pervasive Computing, IEEE*2004, IEEE: Seattle, WA, USA p. 50-57.
37. Maurer, U. and A.S. Smailagic, D.P.; Deisher, M. , *Activity recognition and monitoring using multiple sensors on diferent body positions*, in *International Workshop on Wearable and Implantable Body Sensor Networks*2006, IEEE: Cambridge, MA.
38. Liao, L., D. Fox and H. Kautz, *Location-based activity recognition using relational markov networks*, in *International Joint Conference on Artificial Intelligence* 2005, IJCAI: Edinburgh, Scotland, UK. p. 773-778.
39. Wray, R. and J. Laird, *Variability in human modeling of military simulations*, in *Behaviour Representation in Modeling and Simulation* 2003, Curran Associates, Inc: Scottsdale, Arizona. p. 160-169.
40. Gu, T. and H.K.J.L. Zhanqing Wu; Xianping Tao; Pung, *An emerging patterns based approach to sequential, interleaved and concurrent activity recognition*, in *International Conference on Pervasive Computing and Communication* 2009, IEEE: Galveston, TX.
41. Pei, J., J. Han and W. Wang, *Constraint-based sequential pattern mining: the pattern-growth methods.* Journal of Intelligent Information Systems, 2007. **28**(2): p. 133-160.
42. Rashidi, P. and D.J. Cook, *Adopting the smart home to the user.* IEEE transactions on systems, man and cybernetics, 2009.
43. Hayes, T.L. et al., *Distributed healthcare: simultaneous assessment of multiple individuals.* Ieee Pervasive Computing, 2007. **6**(1): p. 36-43.
44. Jakkula, V. and D.J. Cook, *Mining Sensor Data in Smart Environment for Temporal Activity Prediction*, in *International Conference on Knowledge Discovery and Data Mining Workshop on Knowledge Discovery from Sensor Data (sensor-KDD 2007)*2007: San Jose.
45. Iram Fatima, M.F., Young-Koo Lee and S. Lee, *Computer Science and its Applications*. Lecture Notes in Electrical Engineering Vol. 203. 2012, Netherlands: Springer
46. Amirjavid, F., A. Bouzouane and B. Bouchard, *Intelligent Temporal Data Driven World Actuation in Ambient Environments Case Study: Anomaly Recognition and Assistance Provision in Smart Home*, in *Proceedings of 12th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2013)*2013, IEEE: Niigata, Japan. p. 1-7. Current status: Accepted and under publication.
47. Mathworks. *Information available from: www.mathworks.com*. Available from: www.mathworks.com.

48. Augusto, J.C. and C.D. Nugent, *Smart homes can be smarter.* Designing Smart Homes: Role of Artificial Intelligence, 2006. **4008**: p. 1-15.
49. Singla, G., D.J. Cook and M. Schmitter-Edgecomb. *Incorporating temporal reasoning into activity recognition for Smart Home residents.* in *Proceedings of the AAAI Workshop on Spatial and Temporal Reasoning*. 2008. Chicago, Illinois.
50. Cook, D.J. et al. *MavHome an agent-based Smart Home.* in *PERCOM'03*. 2003. Washington DC: IEEE.
51. Roy, P.C. et al., *A Possibilistic Approach for Activity Recognition in Smart Homes for Cognitive Assistance to Alzheimer's Patients*, in *Activity Recognition in Pervasive Intelligent Environments Atlantis Ambient and Pervasive Intelligence*. 2011, Springer. p. 31-56.
52. Stankovski, V., *Application of decision trees to Smart Homes*, in *Designing Smart Homes*. 2006, Springer.
53. G. Acampora, M.G., V. Loia and A. V. Vasilakos, *Interoperable and adaptive fuzzy services for ambient intelligence applications.* ACM Transactions on Autonomous and Adaptive Systems, 2010. **5(2):**: p. 1-26.
54. Gaggliolі, A. *Optimal experience in ambient intelligence.* in *Ambient Intelligence:The Evolution Of Technology, Communication and Cognition Towards The Future of Human-Computer Interaction*. 2005. Amsterdam: IOS
55. Sadri, F., *Ambient intelligence: A survey.* ACM Computing Surveys (CSUR) 2011. **43** (4): p. 66.
56. Aarts, E., *Ambient intelligence: A multimedia perspective.* Ieee Multimedia, 2004. **11**(1): p. 12-19.
57. Coroama V., R.F. *The chatty environment—providing everyday independence to the visually impaired.* in *2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications (UbiHealth)*. 2003.
58. Vandewynckel J., O.M., Bouchard B., Ménélas B., Bouzouane A. *Towards a real-time error detection within a smart home by using activity recognition with a shoe-mounted accelerometre*. in *International Conference on Ambient Systems, Networks and Technologies (ANT-2013),* . 2013. Halifax, Nova Scotia, Canada, : Elsevier publisher, .
59. BJHC&IM, *The British Journal of Healthcare Computing & Information Management.* 2006. **23**(1): p. 39.
60. José L. Encarnação and T. Kirste, *Ambient intelligence: Towards smart appliance ensembles*, 2005, Springer Berlin Heidelberg. p. 10.
61. Zimmermann, H.J., *set theory and its applications*. 4th ed. 2001: Springer. 544.
62. Liu, H. and L. Yu, *Toward integrating feature selection algorithms for classification and clustering.* Ieee Transactions on Knowledge and Data Engineering, 2005. **17**(4): p. 491-502.
63. Miner, G., Elder, J., Hill. T, Nisbet, R., Delen, D. and Fast, A., *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. 2012: Elsevier
64. Draper, N.R.S., H. , *Applied Regression Analysis Wiley Series in Probability and Statistics*. 1998.
65. Barwise, J., *An Introduction to First-Order Logic*, in *Handbook of Mathematical Logic. Studies in Logic and the Foundations of Mathematics.* 1977 Amsterdam.
66. Ana Colubi, D.D., *Fuzzy sets in statistics.* Computational Statistics and Data Analysis, 2012. **56**

67. Dubois, D., *Special issue: Handling incomplete and fuzzy information in data analysis and decision processes.* International Journal of Approximate Reasoning, 2011. **52**(9): p. 1229-1231.
68. Dubois, D. and H. Prade, *Fuzzy set and posibility theory-based methods in artificial intelligence.* Artificial Intelligence, Elsevier, 2003. **148**.
69. wikipedia, *IRIS data: Information from:* http://en.wikipedia.org/wiki/Iris_flower_data_set.
70. Simonoff, J., *Smoothing Methods in Statistics, 2nd edition.* 1998: Springer.
71. Kullback, S.L., R.A. , *On Information and Sufficiency.* Annals of Mathematical Statistics, 1951. **22**: p. 8.
72. MacQueen, J. *Some methods for classification and analysis of multivariate observations.* in *Fifth Berkeley Symp. on Math. Statist. and Prob.* 1967. University of California Press.
73. Nock, R.a.N., F., *On Weighting Clustering.* IEEE Trans. on Pattern Analysis and Machine Intelligence, 2006.
74. Kabini Karanja, I.W., Nyakoe Nyauma *Application of Adaptive Neural Fuzzy-inference systems in Machining*. 2011: LAP Lambert Academic Publishing.
75. Priyono, A. and M. Ridwan, *Generation of fuzzy rules with subtractive clustering, Jurnal Teknologi, 43:143-153, 2005.* Teknologi, 2005. **43**: p. 143-153.
76. Quinlan, J.R. and J. Ghosh. *Top 10 Algorithms in Data Mining*. in *International Conference on Data Mining (ICDM'06)*. 2006. Hong Kong: IEEE.
77. Cortes, C. and V. Vapnik, *Support-Vector Networks.* Machine Learning, 1995. **20**(3): p. 273-297.
78. Rokach, L.M., O., *Data mining with decision trees: theory and applications.* World Scientific Pub Co Inc, 2008.
79. Qian, H.M., et al., *Recognition of human activities using SVM multi-class classifier.* Pattern Recognition Letters, 2010. **31**(2): p. 100-111.
80. Shannon, C.E., *A Mathematical Theory of Communication.* Bell System Technical Journal, 1948. **27** (3): p. 44.
81. Quaresma, T.G.a.P. *A preliminary approach to the multilabel classifcation problem of Portuguese juridical documents.* in *11th Protuguese Conference on Artificial Intelligence, EPIA 2003*. 2003 Beja, Portugal: Springer.
82. Nielsen, T.D., VERNER JENSEN, FINN, *Bayesian Networks and Decision Graphs.* Information Science and Statistics. 2007: Springer.
83. Schölkopf, B. and A.J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* 2002: MIT Press.
84. Fleury, A., M. Vacher, and N. Noury, *SVM-Based Multi-Modal Classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms and First Experimental Results.* IEEE Transactions on Information Technology in Biomedicine, 2010. **14**(2): p. 10.
85. Rosenblatt, F., *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychol Rev, 1958. **65**(6): p. 386-408.
86. Vapnik, V. and A. Chervonenkis, *Theory of Pattern Recognition.* 1974, Moscow: Nauka.
87. B. E. Boser, I.M.G., and V. N. Vapnik,, *A training algorithm for optimal margin classifiers*, in *Fifth Annual Workshop on Computational Learning Theory*1992, ACM: Pittsburgh.

88. Wu, C., A. Khalili , and H. Aghajan. *Multiview Activity Recognition in Smart Homes with Spatio-Temporal Features*. in *International Conference on Distributed Smart Cameras (ICDSC)*. 2010. Atlanta, GA, USA: IEEE.
89. Fahim, M., et al., *Activity Recognition Based on SVM Kernel Fusion in Smart Home*, in *Computer Science and its Applications, Lecture Notes in Electrical Engineering*, S.-S.Y.e. al., Editor 2012.
90. Jakkula, V. and D.J. Cook, *Detecting Anomalous Sensor Events in Smart Home Data for Enhancing the Living Experience*, in *AAAI Workshops; Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*2011, AAAI: Hyatt Regency San Francisco.
91. Agrawal, R. and R. Srikant, *Fast algorithms for mining association rules in large databases.*, in *20th International Conference on Very Large Data Bases, VLDB,* 1994: Santiago, Chile. p. 487-499.
92. Amirjavid, F., A. Bouzouane, and B. Bouchard, *Optimization of activities in smart homes.* International Workshop on Data Mining in the Life Sciences (DMLS'12), 2012: p. 196-207. URL: http://www.ibai-publishing.org/html/proceeding2012.php.
93. Sowa, J.F., *Conceptual Structures: Information Processing in Mind and Machine*. 1984 Addison-Wesley. 481.
94. Zadeh, L.A., *Fuzzy Sets as a basis for a theory of possibility.* Fuzzy Sets and Systems 1978. **1**: p. 3-28.
95. Murthy, K.K., *Systems Philosophy and Management*. 1999. 486
96. Baader, F., et al., *The Description Logic Handbook Theory, Implementation and Applications*. 2003: Cambridge University Press. 624
97. Wilson, B., *Soft Systems Methodology: Conceptual Model Building and Its Contribution.* 2008 Wiley 278.
98. Wang L., J.L. *fuzzy systems and knowledge discovery*. in *Third International Conference, FSKD 2006,*. 2006. China: Springer.
99. Chiu, S.L., *An efficient method for extracting fuzzy classification rules from high dimensional data.* Journal of Advanced Computational Intelligence, 1997: p. 31-36.
100. Reiter, R., *On Closed World Data Bases.* Spinger journal of Logic and Data Bases, 1978: p. 55-76.
101. Gray, R.M., *Entropy and Information Theory (second edition)*. 2011: Springer.
102. Joseph D. Novak, D.B.G., *Learning How to Learn*. 1984: Cambridge University Press. 199
103. Erickson, M.A., *The Big Book of Concepts.* Trends Cogn Sci, 2003. **7**(8): p. 338-340.
104. Bobillo, F., M. Delgado, and J. Gomez-Romero, *Representation of context-dependant knowledge in ontologies: A model and an application.* Expert Systems with Applications, 2008. **35**(4): p. 1899-1908.
105. Mohinder S. Grewal, A.P.A., *Kalman Filtering: Theory and Practice Using MATLAB.* 2011: Wiley 592.
106. Fortin-Simard D., B.K., Gaboury S., Bouchard B., Bouzouane A., *Accurate Passive RFID Localization System for Smart Homes* in *3th IEEE International Conference on Networked Embedded Systems for Every Application (NESEA 2012)*2012, IEEE: Liverpool, UK. p. 8.
107. Lapointe J., B.B., Bouchard J., Bouzouane A., *Smart Homes for People with Alzheimer's Disease : Adapting Prompting Strategies to the Patient's Cognitive Profile*, in *5th ACM International Conference on Pervasive Technologies Related to Assistive Environments*

(PETRA'12), 2012, Association for Computing Machinery (ACM) publisher: Crete Island, Greece. p. 8.

108. Laprise H., B.J., Bouchard B., Bouzouane A., *Creating tools and trial data sets for smart home researchers: experimenting activities of daily living with normal subjects to compare with Alzheimer's patients*, in *International Conference IADIS e-Health 2010 (EH 2010)*2010: Freiburg, Germany.
109. Amirjavid, F., A. Bouzouane, and B. Bouchard, *Data Driven Conceptual Structures for Simultaneous Activities*, in *12th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2013)*2013, IEEE: Niigata, Japan. p. 1-7. Current status: Accepted and under publication.
110. Cook, D.J., *CASAS Smart Home.* Information available from: http://ailab.wsu.edu/casas/, 2013.
111. Amirjavid, F., A. Bouzouane, and B. Bouchard, *Modeling of activities as temporal multivariable problems*, in *second international conference on smart systems*2013: Rome, Italy. p. 1-7. Current status: Accepted and under publication.
112. Baudrit, C. and D. Dubois, *Practical representations of incomplete probabilistic knowledge.* Computational Statistics & Data Analysis, 2006. **51**(1): p. 86-108.
113. Baudrit, C., D. Dubois, and N. Perrot, *Representing parametric probabilistic models tainted with imprecision.* Fuzzy Sets and Systems, 2008. **159**(15): p. 1913-1928.
114. Dubois, D., *Representation, propagation, and decision issues in risk analysis under incomplete probabilistic information.* Risk Anal, 2010. **30**(3): p. 361-368.
115. Amirjavid, F., A. Bouzouane, and B. Bouchard, *Behavior modeling of coded networks in dynamic networks*, in *World Congress on Engineering (WCE 2012)*2012, Springer: London, UK. p. 1-6. URL: www.iaeng.org/publication/WCE2012/WCE2012_pp2678-2683.pdf.