DISSERTATION

Automatic Scenario Detection for Ambient Assisted Living of Elderly People

Submitted at the Faculty of Electrical Engineering and Information Technology, Vienna University of Technology in partial fulfillment of the requirements for the degree of Doktor der technischen Wissenschaften (equals Ph.D.)

under supervision of

Prof. Dr. Dietmar Dietrich
Institute of Computer Technology
Vienna University of Technology
1040 Wien

and

Prof. Dr. José Fonseca
Electronics and Telecommunications Department
University of Aveiro
Portugal, 3810-193 Aveiro

and

Dr. Dietmar Bruckner
Institute of Computer Technology
Vienna University of Technology
1040 Wien

by

Dipl.-Ing. Guo Qing Yin
Matr.Nr.: 0627120
Pfefferhofgasse, 1030 Wien

30. March 2012

Kurzfassung

Automatische Verhaltenserkennung beruht auf den täglichen Aktivitätsmustern einer Person, wobei das Hauptproblem in der Dynamik dieser Aktivitäten liegt, die sich mit der Zeit verändern. Ein wesentliches Ziel der vorliegenden Arbeit besteht im automatischen Erkennen und Einordnen des Verhaltens von Personen in wechselnden Situationen.

Umgebungsunterstütztes Leben dient der Verbesserung der Lebensqualität älterer Personen und der Verlängerung des selbstbestimmten Lebens in ihrem eigenen Haushalt. Die Arbeit konzentriert sich auf automatische Verhaltenserkennung in bestimmten Situationen des umgebungsunterstützten Lebens. Zur Durchführung dieses Vorhabens werden die Aktivitäten der Personen analysiert und klassifiziert, wobei zwischen regelmäßigen und zufälligen Verhaltensmustern unterschieden wird. Dabei werden zur Analyse regulärer Aktivitäten und zur Entwicklung von Aktivitätsmodellen Gaussian mixture Modelle und Split-Merge Algorithmen eingesetzt. Zur Behandlung zufälliger Aktivitäten und zum Aufbau eines Aktivitätsmodells wird ein angepasstes Hidden Markov Modell verwendet. Darüber hinaus wird Sensorfusion zur Ableitung von Aktivitätsmodellen aus Sensordaten unterschiedlicher Sensortypen benutzt. Auf Basis dieser Aktivitätsmodelle wird dann eine automatische Verhaltenserkennung realisiert.

Die Arbeit beinhaltet drei Kernpunkte: Anpassung von Split-Merge Algorithmen und Hidden Markov Modellen zum Aufbau von Aktivitätsmodellen für ältere Personen, um normale Tätigkeiten mit verschiedenen und typischen Tagesabläufen abzubilden; dies ist in Kapitel 3 dargelegt. Nutzung der Vorteile von Sensorfusion zur Vervollständigung des Aktivitätsmodells in Kapitel 4. Basierend auf dem statistischen Aktivitätsmodell wird in Kapitel 5 ein Verfahren zur automatischen Verhaltenserkennung entworfen und diskutiert. Die Resultate zeigen, dass das typische tägliche Verhalten automatisch erkannt werden kann. Abschließend folgen in Kapitel 6 Zusammenfassung und Ausblick.

Im Rahmen dieser Arbeit wurden über 4 Monate gesammelte reale Daten zum Test des entworfenen Modells verwendet. Es zeigt sich, dass ungewöhnliche Aktivitäten in der Lebensumfeld älterer Personen erkannt werden. Die Erkennung korreliert mit dem von den Testpersonen geführten Testtagebuch. Die in dieser Arbeit entwickelten Ansätze und Technologien können in echten Sensorsystemen zur Unterstützung des täglichen Lebens älterer Menschen genutzt werden.

Abstract

Automatic scenario detection is based on the daily behavior patterns of a person. The main problem is that the person's activities are dynamic, meaning that most scenarios do not recur in exactly the same way. An important objective of current research is therefore to detect and quantify typical scenarios for a person automatically in a dynamic situation.

Ambient assisted living focuses on enhancing the quality of life and prolonging independent living of the elderly in their own home. This thesis concentrates on automatic scenario detection in a range of ambient assisted living scenarios. In order to realize automatic scenario detection, the activities of the elderly must be analyzed and classified. Regular behavior patterns and random behavior must be recognized and treated differently. In the research project covered in this thesis, Gaussian mixture models and the split-merge algorithm were used to analyze regular behavior and learn the behavior model. A hidden Markov model was utilized to recognize and deal with random behavior, thus refining the behavior model. Additionally, sensor fusion was used to create a complete behavior model from sensor state data from different types of sensors. Finally, automatic scenario detection was realized based on these behavior models.

There are three important scientific points in the thesis: utilizing the Gaussian mixture model, split-merge algorithm and hidden Markov model to learn behavior models for elderly persons, i.e. modeling the normal activities of an elderly person with several typical daily routines. This is illustrated in Chapter 3. Utilization of the advantages of sensor fusion makes the behavior model complete in Chapter 4. Based on the statistical behavior models, the realization of automatic scenario detection is introduced and discussed in Chapter 5. The results indicate that the typical daily scenario of the user can be automatically detected. Chapter 6 contains the conclusions and outlook.

In the thesis, a real dataset covering a duration of about 4 months was used to test the learnt model. The detections were compared with a daily journal written by the elderly and the results indicated that unusual activities within the living environment of the elderly person were being detected correctly. Approaches and technologies which could be used in a real sensor system to help the elderly in their daily lives will also be developed and discussed within this thesis.

Vorwort

Das Konzept des Ambient Assisted Living wurde mit dem Ziel entwickelt, die Lebensqualität älterer Menschen zu verbessern und ihre eigenständige Lebensführung in den eigenen vier Wänden zu verlängern. Zum Schutz der Privatsphäre ist dabei die Anwendung einiger Arten von Sensoren nur begrenzt möglich. Darüber hinaus treten aus Altersgründen öfters Gedächtnis- und Bewegungsschwierigkeiten auf, sodass einige Methoden wie das Tragen von Sensoren am Körper oder die Nutzung eines Notknopfs zum Ruf nach Hilfe nicht zuverlässig genug angewendet werden können. Aus den genannten Gründen schlägt die Dissertation einen neuen Ansatz vor: die Beobachtung der Aktivität älterer Menschen mit nicht invasiven Sensoren und die Erstellung eines auf den Beobachtungsergebnissen basierenden Aktivitätsmodells; die Entwicklung einer automatischen Verhaltenserkennung unter Verwendung des erlernten Modells; schließlich die automatische Alarmierung geeigneter Pflegepersonen im Fall der Erkennung ungewöhnlicher Situationen.
In Kapitel 1 werden Motivation der Arbeit, Problemstellung und Argumentation, Ziele und Beitrag der Arbeit dargelegt. Kapitel 2 beschreibt den Stand der Technik zu Ambient Assisted Living, Situations- und Verhaltenserkennung. In Kapitel 3 werden die Aktivitäten älterer Menschen analysiert und in zwei Formen klassifiziert: regelmäßige und zufällige Aktivität. Zur Analyse der regelmäßigen Aktivitäten wird ein Gaussian Mixture Model und Split-Merge Algorithmus verwendet, zum Erlernen zufälliger Aktivitäten dienen ein Hidden Markov Modell und Forward Algorithmus.
Wegen der Nachteile der einzelnen Sensormessung wird in Kapitel 4 Sensorfusion eingeführt. Zur Realisierung werden zwei Ansätze vorgeschlagen: a) Fusionieren der Einzelaktivitätsmodelle jedes separaten Sensors zu einem gemeinsamen Modell, oder b) Erlernen des Aktivitätsmodells mit den bereits fusionierten Sensormessungen. Die Umsetzung der ersten Methode erweist sich wegen der unterschiedlichen Struktur der Aktivitätsmodelle je Sensor als schwierig. Die Nutzung des zweiten Ansatzes erfordert die Lösung des Problems wie aufeinander folgende Zustände zu fusionieren sind. Dazu wird in der Arbeit eine Kompatibilitätsmatrix eingeführt, mit der die Ähnlichkeit aufeinander folgender Zustände beurteilt und sie abhängig von einem vorgegebenen Schwellenwert zusammengeführt werden. Darüber hinaus wird basierend auf den Zustandsdaten und definierten Datenbeispielen diskutiert, wie der optimale Schwellenwert ermittelt werden kann. Unter Ausnutzung der Vorteile der erwähnten Methoden wird ein Aktivitätsmodell mit Sensorfusionsdaten eingeführt und zusätzlich Multisensordatenkorrelation vorgestellt.
Basierend auf den dargelegten Vorarbeiten wird in Kapitel 5 eine automatische Verhaltenserkennung entwickelt. Für regelmäßige Tätigkeiten werden Mittelwerte und Standardabweichungen der einzelnen Cluster berechnet und die Grenzen für diese Cluster angegeben. Eine Tätigkeit wird als falsche Aktivität behandelt, wenn der Zeitpunkt der Aktivität außerhalb der angegebenen Grenzen liegt. Für zufällige Aktivitäten wird der täglichen Routineablauf der Benutzer mit einem Hidden Markov Modell analysiert. Die automatische Verhaltenserkennung ergibt sich damit aus der Erkennung eines am besten passenden oder eines ungewöhnlichen Tagesablaufs unter Nutzung eines Hidden Markov Modells sowie des Forward Algorithmus. Weiters werden die übereinstimmenden Ergebnisse von Einzelsensoren und Multisensoren gegenübergestellt, um die Vergleichbarkeit der verwendeten Methoden zu zeigen. Abschließend werden drei Beispiele als Fallstudie behandelt, in denen verschiedene Arten von ungewöhnlichen Aktivitäten basierend auf dem gelernten Modell entdeckt wurden. An Hand des von der Testperson geführten Tagebuchs zeigt sich, dass die durch das Modell als ungewöhnliche Aktivitäten identifizierten Ereignisse genau mit tatsächlichen Situationen korrelieren, die während des Tests auftraten.
Abschließend bringt Kapitel 6 Fazit und Ausblick und fasst die wichtigsten Schlussfolgerungen und Beiträge zusammen. Darüber hinaus werden mehrere Vorschläge zu möglichen zukünftigen Forschungen gemacht.

Preface

The concept of ambient assisted living was proposed in order to enhance the quality of life and prolong independent living of elderly people in their own home. Due to privacy concerns, the applicability of certain types of sensors is, however, limited. Furthermore, some elderly people have problems with memory and movement, thus some methods, for example the wearing of a sensor on the body or activating a button to call for help, cannot be fully realized by the elderly people themselves. In answer to these issues, a novel approach is proposed in this thesis: to observe the behavior of the elderly person with non-invasive sensors and learn the behavior model of the user, and to realize automatic scenario detection based on the learnt model. Furthermore, to send an automatic alarm signal to an appropriate human caregiver when unusual situations are detected.

In chapter 1, motivation of the work, problem statement and reasoning, aims of the work, and scientific methods used in the thesis are introduced. In chapter 2, the state of the art in ambient assisted living, situations and scenario recognition are introduced and discussed. In chapter 3, the behavior of elderly people is analyzed and classified into 2 types: regular and random behavior. A Gaussian mixture model and the split-merge algorithm were used to analyze regular behavior, and the hidden Markov model and forward algorithm were utilized to learn and analyze random behavior.

Because of the disadvantages of single sensor measurement, sensor fusion is introduced in chapter 4. Two approaches are suggested to realize sensor fusion: a) fusing the behavior models built from each single sensor to get the fused model, or b) fusing the single sensor measurements first, then using these mixed measurements to learn the behavior model. The former method proved difficult to realize because the structure of the behavior models from each single sensor is different. In order to utilize the second approach, the problem of how to fuse the consecutive states had to be dealt with. A compatibility matrix was therefore introduced to determine the similarity of consecutive sensor states, with a predefined threshold value to judge whether or not to fuse consecutive states. Furthermore, examples of how to evaluate the optimal threshold value based on state data and designed data are shown and discussed. Taking advantage of the abovementioned methods, the behavior model with sensor fusion data is then introduced and multi-sensor data correlation presented.

On the basis of the above work, automatic scenario detection is realized in chapter 5. For regular behavior, the mean value and standard deviation of each cluster were calculated, as well as the boundaries for these clusters. A behavior would be treated as unusual if the time of the behavior was outside the calculated boundaries. For random behavior, the user's daily activity routines were analyzed using a hidden Markov model. Automatic scenario detection was realized when the best matching routine or an unusual daily routine were detected using the hidden Markov model and forward algorithm. Furthermore, matching results from single sensors and multi-sensors were compared to show the comparability of the utilized methods. Finally, 3 examples are offered as a case study in which different types of unusual activities were detected based on the learnt model. When a daily diary written by the user was checked, it verified that the events identified by the model as unusual activity correlated accurately with real-life situations which had occurred during the test.

The conclusions and outlook are presented in chapter 6. The main conclusions and procedures are summarized, and several proposals are made for possible future work.

Acknowledgements

I would like to thank my professor and supervisor Dietmar Dietrich who contributed to this work with his guidance, feedback and comments; Prof. José Fonseca for his feedback and comments, especially in the final phase of the work; Dr. Dietmar Bruckner for his guidance at all times. My discussions with him were invaluable for the formulation of the results in this thesis.
I also wish to thank my good friend Trevor Brawn for his advice on English grammar and syntax in my writing, and my best friend Wolfgang for his encouragement and practical assistance with the wording of the thesis.
Lastly but most importantly I would like to express my gratitude to my aging parents in China, who have supported me during the last more than 10 years as a student in Germany and Austria. This dissertation is not only for me but also for them.

Contents

Abbreviations

AAL	Ambient Assisted Living
AI	Artificial Intelligence
ARS	Artificial Recognition System
ASD	Automatic Scenario Detection
BAS	Building automation system
CM	Compatibility Matrix
DM	Data Mining
EEG	Electroencephalography
EM	Expectation Maximization
FA	Forward Algorithm
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HSMM	Hidden Semi Markov Model
ICT	institute of computer technology
MC	Markov Chain
ML	Machine Learning
SF	Sensor Fusion
SMA	Split Merge Algorithm

1 Introduction

The aging problem is very important for society [Bur04, p. 655]. Ambient assisted living (AAL) may provide a way of solving this problem. AAL focuses on enhancing the quality of life and prolonging independent living of elderly persons within their own home with the help of modern technology. However, since elderly people are often faced with problems such as movement or memory disorders, the question arises how they are able to interact with modern technical systems.

The goal of the thesis is automatic scenario detection (ASD) regarding the activities of elderly persons within their living environment. This is achieved by use of an intelligent, adaptive network of sensors, which are installed in the living environment of the user in order to thoroughly observe his activities and behavior, and letting this network learn the daily normal behavior model of the user. Based on the behavior model, automatic scenario detection may be realized. In case of unusual activities or behavior, an alarm plan can be executed (contacting the user, calling a neighbor, calling an external organization, etc.).

This thesis describes the essential scientific challenges within the selected approach. It combines building automation, symbolic computing, statistics, and gerontology and searches for a new solution in ambient assisted living (AAL).

In this first chapter, the motivation for the choice of research topic will be outlined, and a concise problem statement and explanation of reasoning will lead to the step-by-step introduction of a solution plan. The aim of the work will be presented and the scientific methods used in the proposed solution will be detailed.

1.1 Motivation

According to "Ambient Assisted Living - Country Report Austria" [ACR, p. 3], 21.1% of the population in Austria was older than 60 in the year 2001. This percentage is predicted to increase to 32.1% by the year 2030. At the same time the relative number of potential care providers will be lower. From the same source [ACR, p. 5]: in the year 2001, the ratio of social spending to gross domestic product was 28.5%. Of this, 49.5% was devoted to maintaining the health and functional capability of surviving senior citizens. This means that about 14% of the gross domestic product was being used for caring for the elderly.

It is clear that the aging of the population is not only an economic issue but also a social problem. It is also obvious that progressively more elderly people will need help, but at the same time fewer and fewer persons will be available to work as caregivers to the elderly in an aging society. Who can take care of these old people and how can they be given support to better enjoy their twilight years? The limited human resources and high personnel costs pose a dilemma.

Ambient assisted living (AAL) is a concept that aims to allow elderly people to live more independently. It can help to prolong the period of independence, reduce the demand for human resources and decrease personnel expenditure.

There is a joint AAL programme [AAL] composed of 24 organisations from 22 countries, such as Austria, Germany, France, Finland, Switzerland, Luxembourg, Spain and the United Kingdom. The goal of this programme is to improve the autonomy of elderly people using technical methods.

1.2 Problem Statement and Reasoning

The goal of ambient assisted living (AAL) is to help older people achieve independent living. To achieve this goal, there are several problems which must be addressed.

1.2.1 Problems of Ambient Assisted Living

The first problem lurks in the term “ambient assisted”. This actually means that the user’s home or “ambience” (i.e. the monitoring set-up) should collect information from the surrounding environment and from the user himself, and send the information to an appropriate caregiver. In this way elderly people can get help in time when they need it.

Within this issue there are two sub points which must be dealt with: first, how the “ambience” collects information from elderly people. In effect, many sensors must be mounted in the living environment and elderly people may be required to carry these sensors on their bodies all the time. This way, the system can obtain information from the user. However, is it feasible for elderly people to wear sensors all the time? If so, can it be made convenient and comfortable for them? Second, what kind of information should the “ambience” send to the caregiver? This is in part an ethical problem. Should image and voice information be sent to the caregiver or control center? Would such an approach intrude on the privacy of elderly people?

The second problem comes from the object - the people themselves: elderly people often suffer from medical conditions such as movement disorders, memory disorders, etc. How can elderly people use such technical systems, and are there any sensors or controls that need to be activated by the users themselves?

Due to movement disorder, memory disorder or other physiological and psychological problems, many elderly people do not want to learn new skills such as how to use a computer, even to send an e-mail, or do not want to use a new type of mobile phone. In such a situation, or if a system or product is somewhat difficult to use, the affected persons might not be able to use it properly. The effectiveness of this kind of system or product would therefore be reduced in reality. On the other hand, if an elderly person encounters action obstacles or an unknown or dangerous situation, how can he or she activate the system or product and get help in time?

1.2.2 Requirements

Based on the abovementioned issues and questions, a sensor system with the following requirements will be developed in this thesis: no cameras or microphones shall be used; no sensors shall be required to be worn on the body; nothing should need to be activated by the user; nothing should need to be learnt by the user; an alarm should be sent to a caregiver in case of emergency; the user's privacy should be protected and his comfort and safety increased. Because this work is not only theoretical research but also has a real-life practical application, the requirements of the developed system have to be discussed in detail beforehand. For example: What kind of system should it be? What types of sensors will work most effectively? What service or functionality is provided to the user by the system? These questions will be discussed in the following sections.

1.2.2.1 System Works Independently

First and foremost, the sensors must work independently, because the users are elderly people. Many of them have different physiological and psychological problems, such as movement disorder or memory disorder, and some of them do not want to (or would be unable to) learn how to use new technical gadgets. Furthermore, potentially dangerous situations exist, such as a fall, as a result of which the person may either be unconscious or unable to move. In this case the person would not be able to activate the product themselves, even if they knew how. Given these circumstances, the system must be able to detect emergencies and activate itself.

1.2.2.2 Improving the Physical and Psychological Comfort

Secondly, the sensors should preferably work in an invisible fashion. Cameras and microphones should be avoided if possible. Visible sensors mounted in the living environment make it obvious that the user is being permanently observed. This is not comfortable for the user, particularly regarding cameras and microphones. In [TIL04, p. 159], the authors indicate that microphones and cameras are so common and generally used as recording devices that they can be perceived as invasive and threatening by some people. Examples from [TIL04, p. 173–174]: one of the subjects (an elderly person) stated that she would not have agreed to the study if it had involved video observation. A second subject (also elderly) would have agreed but would not have allowed cameras in the bathroom. Furthermore, the user should not be forced to carry different kinds of sensors on his or her body all the time. Firstly, because of the inconvenience; secondly, due to possible physiological and psychological problems of elderly people, it would be impractical and unrealistic to expect them to remember to wear and operate the equipment correctly. They may forget to hang it around their neck or put it on their wrist; or they may misplace it and be unable to find it when it is needed.

The user should not need to wear any sensors as explained above. This should be avoided because of the inconvenience to the user and because certain common physiological and psychological problems of elderly people would render it unrealistic to expect them to always wear and operate the equipment correctly. The use of cameras and/or microphones should also be avoided if possible, as they openly intrude into the individual's privacy. Cameras would initially seem to be an obvious solution. If they were installed in the living environment of the user, images of the user could be sent to a caregiver or other monitoring persons outside the living environment. That way the caregiver could monitor the user directly. If any dangerous situations occurred, the caregiver could recognize them and help the user immediately. The method sounds good in principle, but in reality cameras and microphones can be perceived as invasive and threatening by many people [TIL04, p. 159]. This is especially true if the cameras are installed in the living room or bathroom. Another example: if microphones were installed in the living environment, the user could call the caregiver directly through the microphone and get assistance that way whenever he needed it. However, microphones could also send private conversations or other private sound information to the caregiver. These two examples show clearly that cameras and microphones intrude into the user's private life. Thus in the thesis work, cameras and microphones were avoided. In the thesis only sensors such as motion detectors, door contactors, pressure mats and similar kinds of passive sensors were used. They send no images or sound information to the caregiver. They should not even send any concrete parameters outside of the living environment. All the parameters of the user should be processed within the system and an alarm signal should only be sent to the caregiver if an unusual situation occurs. Thus the privacy of the user would be fully protected.

1.2.2.3 Increasing Safety

Thirdly, the system should increase the safety of independent living for the elderly. The thesis focuses on automatic scenario detection; meaning if unusual situations and scenarios occur, the system should detect them in time, thus increasing the safety of independent living in the elderly person's own home. For example, say the user falls ill and is unable to get out of their bed. Using the behavior model learnt for this person as a reference, the system should be able to detect this abnormal situation and send an alarm signal to the caregiver. Thus the user would receive help in time before the situation became worse.

According to the description above, the system or product should be smart and able to work independently of the user. In other words, it should be "cognitive", "learning", and capable of "analysis and decision making". Each of these sections is a topic of "Artificial Intelligence", so it is obvious that the problems are difficult and full of challenge.

1.2.3 Reasoning

The picture 1.1 shows the entire reasoning process. Much work has been done with ambient assisted living (AAL), such as [HKWE08, p. 113–119], [HSM07, p. 29–32], [FAP08, p. 219–224] and [YRA98, p. 387–392]. In this thesis, a new approach is developed: to realize scenario recognition based on the daily living habits of elderly people (such as when they get up, take their meals, go to bed, etc.). For example, an elderly person may regularly get up in the morning at about 7 o'clock. One day the person may get up at 6:50 and the next day he may rise at 7:10. This means the average time at which the person gets up is about 7 o'clock, but the exact time may vary. In such situations, statistical methods are used. Over a period of, for example, one week or one month (or longer; it is clear that the sample time interval should be long enough to include common and less common causes for variation, e.g. weekends or the changing seasons etc.), the different points in time when the subject gets up in the morning will be gathered together. Using statistical approaches such as the split-merge algorithm (SMA) and hidden Markov models (HMM) ([YBZ09, p. 6], [YB09b, p. 4155–4158], [YB09a, p. 4], [YB09c, p. 60–63] and [Bru07, p. 26–32]) an expected time frame in which the person normally gets up in the morning can be calculated. For example, a sample might show that in the past month, a given subject had gotten up at about 7 o'clock with a deviation of 20 minutes. This means the subject had gotten up no earlier than 6:40 and no later than 7:20. Therefore the subject would normally be expected to rise within this time frame. In the same way an estimated time for other habits of the subject, like the times when he or she has lunch or dinner or goes to bed, can be determined. These different habits are then gathered together to form a statistical behavior model of the user. This statistical behavior model could be gained from motion detectors installed in the living room and bedroom, for example. In order to get information on the entire living environment, the bathroom and kitchen and perhaps other spaces would have to be included, meaning more sensors would be needed. Sensor fusion (SF) can then be utilized to compile a behavior model of the user in the entire living environment. Furthermore, through fusing with the data of various other sensors, the subject's activities can be monitored in detail. After sensor fusion (SF), a comprehensive statistical behavior model of the user may be built. Based on the statistical behavior model it becomes possible to realize automatic scenario detection (ASD) within the elderly person's living environment. All of the above will be discussed and explained step by step in more detail in the following chapters.

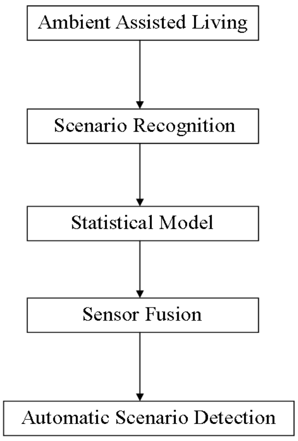


Figure 1.1: From ambient assisted living to automatic scenario detection

1.2.3.1 From Ambient Assisted Living to Scenario Recognition

There are many ideas for AAL such as robotics for the elderly, computers for the elderly, video surveillance and sensors worn by the elderly. In [HKWE08, p. 113–119], a conversational robot was developed in order to increase the enjoyment of the elderly in their daily life. An intelligent, dynamic facility was introduced in [HSM07, p. 29–32] which assists the elderly user in browsing the internet. In [FAP08, p. 219–224], a video surveillance system was proposed. Its aim was to detect when a subject took a fall. A ring sensor was introduced in [YRA98, p. 387–392]; this was a 24 hour tele-nursing system.

All of these ideas and concepts provide help to the elderly, but in this thesis a new method will be presented that aims to be more convenient, comfortable, and not intrude on the privacy of the subject. As stated above, the method should involve no cameras or microphones. No sensors should need to be worn on the body of the subject. There should be nothing the subject needs to activate, and nothing the subject needs to learn. An automatic alarm should be sent to the caregiver in case of an emergency. This would provide privacy protection while increasing comfort and safety. All of these restrictions and requirements make the method very difficult to realize. Where is the breakthrough point? The answer lies with the elderly subjects themselves.

After many years of life, most elderly people have their own living habits and daily routines: getting up in the morning, going to the toilet, showering, eating breakfast, . . . , cooking in the kitchen, eating lunch, napping, . . . , having dinner, watching TV, . . . , going to bed at night. The elderly person repeats their habitual activities nearly every day, and the same behavior tends to occur at around the same time. For example, a certain person might get up at around 7:00 in the morning. Some days maybe a little earlier, for example 6:50, and other days a little later, for example 7:10, but the behavior of "getting up in the morning" for this person occurs just around 7:00. This is a living habit of the person. Other living habits may include when the person has lunch, when he/she watches TV in the evening, when he/she goes to bed at night. These daily routines can be gathered together and compiled into a living habits model. Because the habits are relatively stable, the living model of the elderly person should be stable. Can we use the stable living habits of the elderly to learn a daily scenario model for an individual? Furthermore, can we use the daily scenario model to detect abnormal behavior? These questions lead directly to the breakthrough point and the basic idea of the thesis.

From the above discussion and analysis, the basic idea is extracted. That is, it should be possible to analyze the stable living habits of elderly persons to learn a daily scenario model, and use the completed model to detect abnormal behavior. Just as picture 1.1 indicates, the analyzing and reasoning work may then be translated from ambient assisted living to scenario recognition. Step-by-step in-depth analysis and reasoning follows in the next sections.

1.2.3.2 From Scenario Recognition to Statistical Model

Based on the presumption that most elderly people have relatively stable living habits, it is possible to learn an individual daily scenario model for a given elderly subject. However, the daily living patterns of the elderly are not entirely unchanging - there are always some differences in the daily living pattern. For example, on some days a person might get up a little later, on other days they might go to bed a little earlier. On weekends they may spend a little longer on cleaning work or take a walk outside. So the activities of the elderly will follow a similar pattern but will not be exactly the same each day. In certain situations there may be large differences. This means the activities have deviation, which introduces random variables.

The statistical model can deal with these kinds of activities, for the function of the statistical model is to describe the behavior of objects with random variables and determine the probability distributions. For this reason the statistical model was chosen to solve the scenario recognition problems.

From Statistical Model to Gaussian Mixture Model and Split-Merge Algorithm

If we think of the activities of elderly people as variables, we obtain a relationship between the activities and variables. The picture 1.2 illustrates the reasoning process behind the creation of a special model from analysis of basic activities using an appropriate algorithm. On the top left side of picture 1.2 are the activities of elderly people. This includes all types of daily actions by the subjects such as getting up in the morning, going to the WC, taking a shower in the bathroom, etc. The activities are treated as variables because they are random and dynamic. Within the overall activity pattern there are some activities which happen at nearly the same point of time every day, for example the elderly person may take a tablet of medicine in the morning at about 8 o'clock. Such activities are termed special activities. Just like activities treated as variables, the special activities are treated as cluster variables because they focus on some special value intervals (cluster). Furthermore, the activities will be analyzed in order to form a behavior model of the subject. The analyzing of special activities is the same as cluster analyzing if the special activities are treated as cluster variables. For cluster analyzing, the Gaussian mixture model (GMM) and split-merge algorithm (SMA) are used. More discussion and explanations with picture 1.2 will occur in the following section.

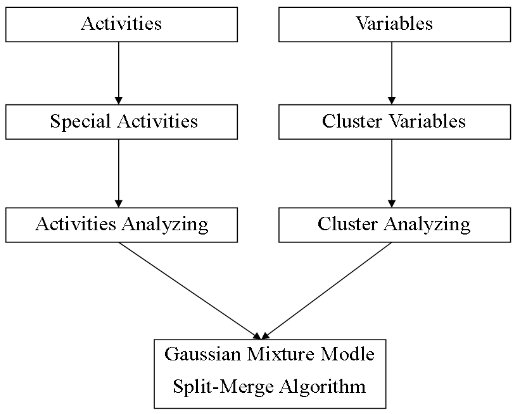


Figure 1.2: From statistical model to Gaussian mixture mode and split-merge algorithm

One aspect of activities of the elderly is that the same activities happen daily and mostly in a particular time domain. Such as: time for breakfast, lunch, dinner or time for taking medicines. These activities happen daily in relatively stable time domains, yet there are temporal deviations within the activities. These kinds of activities are just like cluster variables that focus on some special location but have deviation. Now that a relationship between special activities and cluster variables has been built, the question of activity analyzing is translated to one of cluster analyzing.

The Gaussian mixture model (GMM) is one of the best candidates for cluster analyzing, as it allows different means and deviations within a model function. Furthermore, for analyzing the mean and deviation of these variables, the split-merge algorithm (SMA) can be introduced. Because the variables are dynamic variables, they lead to the changing of the mean and deviation. The SMA handles these kinds of dynamic variables and produces the mean and deviation in a dynamic fashion. This means the mean and deviation parameters derived from a SMA are directly influenced by the dynamic variables. More detail about the GMM and SMA can be found in [YBZ09, p. 6], [YB09b, p. 4155–4158], [YB09a, p. 4] and [YB09c, p. 60–63].

In summary, the Gaussian mixture model and split-merge algorithm are chosen for the analysis of certain activities. In the next step, a different model will be used to analyze other types of activities.

From Statistical Model to Hidden Markov Models

Another kind of activity is the kind which happens daily, but not in any particular time domain. Example: the daily routines of the elderly in the living room. One day a subject might stay in the living room from 9:00 to 12:00 and carry out activities there, then from 12:00 to 14:00 might engage in activities away from the living room. Later, from 14:00 to 23:00 the subject might return to the living room and carry out more activities there. Then from 23:00 to 06:00 the next day the subject might sleep in the bed area of the living room without any activities. But the next day after getting up at 06:00 there might be activity in the living room until 10:00 followed by a phase of rest until 13:00. From 13:00 to 15:00 there might be more activities, then from 15:00 to 21:00 is the subject could be quiet again. Later from 21:00 to 23:00 the subject might engage in more activities in the living room and then sleep again until the next morning at 06:00.

The above example indicates that these kinds of activities are not located in some special time domain, so that the Gaussian mixture model and split-merge algorithm cannot be used for analysis here. But based on the work in [Bru07, p. 137], this kind of activity can be handled using a hidden Markov model. [Bru07, p. 137] uses the hidden Markov model to represent different daily behavior models. This thesis takes advantage of [Bru07, p. 137], however it uses not only a hidden Markov model (HMM) but also the split-merge algorithm (SMA) to suit the ambient assisted living (AAL) domain. On the other hand, [Bru07, p. 137] uses a single motion detector as the original data source. In this thesis, different types of sensors in different locations are used as original data sources. Furthermore, sensor fusion (SF) is used to treat data from different types of sensors. Based on this sensor fusion, the behavior model of the user is learnt, and automatic scenario detection (ASD) is applied to the resulting behavior model.

In summary, due to the points described above, the GMM, SMA and HMM were chosen to analyze different activities of the elderly and to learn statistical behavior models of the user. This represents the first step in the solution approach.

1.2.3.3 From Statistical Model to Sensor Fusion

Within the living environment of the elderly person there are different areas such as the living room, bathroom, WC, and entrance. Different types of sensors (such as motion detectors, door contactors, pressure mats) may be installed in different locations in the living environment. These different types of sensors investigate the activities of the user within the living environment. For a complete behavior model of the subject, the detected activity data from the user need to be fused together. The picture 1.3 shows one example of an application environment with different types of sensors. In picture 1.3 there are 3 different areas: bathroom with WC, entrance, and living room. In the bathroom there is a sensor (number 1), for example a motion detector. It can detect movement if the subject carries out activities in the bathroom and WC. In the entrance there is a motion detector (sensor number 2). If the user engages in activities in the entrance, sensor 2 will detect those activities. In the living room there are 3 sensors numbered 3, 4, and 5. In the example, these are also all motion detectors. Sensor 3 is installed at the top right corner near the bed in the living room. Sensor 4 is installed at the bottom right corner near the table and chairs. Sensor 5 is installed at the bottom left corner near the kitchen in the living room.

Each sensor in the picture 1.3 can only detect activities and learn a statistical model of the elderly person's movements when the user is at the relevant location. For instance, sensor 1 can only investigate the activities of the user when the user is in the bathroom and WC. Thus using the

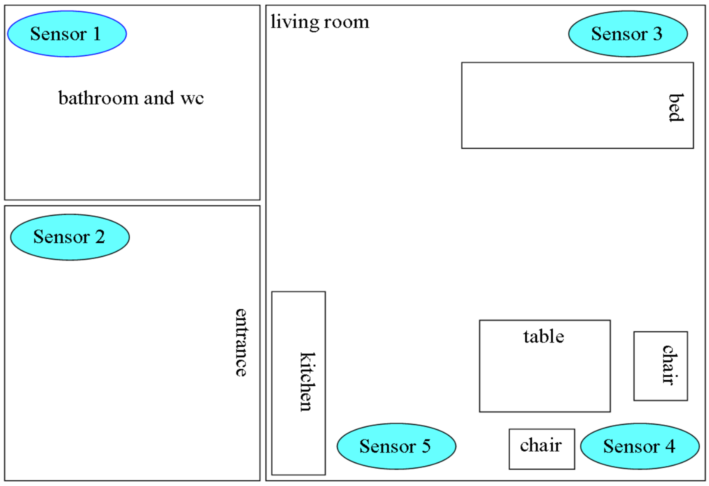


Figure 1.3: An application environment with different types of sensors

data of sensor 1 exclusively, only a behavior model of the user in the bathroom and WC could be learnt. But in the course of a day, the subject frequently moves from one location to another or stays in one location for a longer time. If the subject moves from the bathroom to the entrance, this activity must be analyzed using the output from bathroom/WC sensor 1 and entrance sensor 2 together. Furthermore, if the user moves from the entrance to the living room, the output of sensor 2 in the entrance combined with the outputs from sensors 3, 4 and 5 in the living room must be considered. So for a complete behavior model for the entire living environment (bathroom and WC, entrance, and living room) different kinds of statistical models from different types of sensors must be merged together. Sensor fusion (SF) can be applied to achieve this merging.

If different types of sensors are focused on a single location, SF produces a more accurate learning result. In the example above, assume that sensors 4 and 5 are motion detectors. They can thus analyze movement between the middle and upper areas of the living room, but can only partially interpret activities in the upper left-hand and upper right-hand corners of the room. If sensor 3 is a pressure mat, however, it may be used to analyze activities simultaneously with motion detectors 4 and 5. Sensor fusion of the output of sensors 3, 4 and 5 might then conclusively show, for example, when the user is positioned in the upper right-hand corner of the living room. In this example, SF can be seen to create a focus for activity detection in the upper right-hand corner of the living room. This means that SF allows sensors to detect activities not only in a broad area (from the middle to the upper edge of the room), but also within a smaller concrete area (the upper right-hand corner), with increased accuracy of detection and learning for the defined concrete area.

Generally speaking, by using SF the learning results from the individual sensors will be merged together and combined into a complete model system with higher accuracy (e.g. the above example). The merging of all learnt statistical models through SF in order to create a complete statistical model system is the second step in the solution approach.

1.2.3.4 From Sensor Fusion to Automatic Scenario Detection

Above we discussed that through SF, all the learnt models can be merged together and combined into a complete behavior model of the user, thus representing the entire living behaviour of a particular elderly person. The "whenever, wherever and whatever" behaviors of the user can be observed by different sensors unobtrusively installed in the user's living environment. Based on the learnt model, if there are any unusual situations or scenarios, the system can detect them and send an alarm signal to the user or a predefined caregiver. This represents the third step in the solution approach.

In the above sections, ambient assisted living (AAL), the statistical model, Gaussian mixture model (GMM), split-merge algorithm (SMA), hidden Markov models (HMM), sensor fusion (SF) and lastly automatic scenario detection were discussed step by step. Together, they form a clear path towards the ultimate goal of automatic scenario detection.

1.3 Aims of the Work

This thesis will show how statistical models of a subject's normal living activities can be learnt. Once the model for a particular person is complete, if abnormal activities by the user occur, the model will detect these automatically and an alarm signal will be sent out. This is the aim of the thesis. An example: the user falls to the floor and cannot get up by himself. For a situation such as the above, an alarm plan can be worked out (e.g. contacting the user by phone or other auditory device, calling a neighbor, or calling an external organization). The system must work independently and in a preferably invisible fashion. Its intent is to increase comfort and safety and prolong independent living of the elderly.

1.3.1 Scientific Aims

The thesis is based on the work of [Bru07, p. 137], but also expands and extends the work of [Bru07, p. 137] (with hidden Markov models to analyze single motion detector data). Furthermore, sensor fusion (SF) is used in the thesis in order to analyze different types of sensors and form a more complete behavior model. Other methods including hidden Markov models (HMM), the split-merge algorithm (SMA), and sensor fusion are used to realize automatic scenario detection (ASD).

The aim of the work is automatic scenario detection (ASD), but in order to reach to this aim, several other problems had to be solved first. One was that the statistical model of the elderly had to be learnt. Another was using sensor fusion to learn a complete statistical model of the user. The last problem was that of communicating the results of the statistical model and sensor fusion (SF) in order to realize automation for scenario detection.

1.3.2 Statistical Model

A combination of different methods and adaptations of different approaches and algorithms were used to learn the behavior model. Some may be applied to activities which happen regularly, such as the user eating lunch or dinner at a certain time every day, or taking medicine at a fixed time. Other kinds of activities are random; for example watching TV in the living room, then going to the kitchen to get something, going to the bathroom to wash one's hands and finally going back to the living room. These kinds of activities happen irregularly. Their locations and time durations are likely to change from one time period to the next, and from one day to the next. In this thesis, different algorithms were used to learn models for different activities. Thus "normal" behavior (i.e. the typical daily routines) of a given elderly person may be modeled. The system has the ability to learn a "normal living behavior model" of the user whose daily routines will not always be the same. For instance, the weekend routine may perhaps not be the same as the weekday routine, and holidays may be different from normal week days. Therefore models for several different routines should be learnt. In the designed process, the system has to learn these models itself, which was one of the main challenges of the project.

1.3.3 Sensor Fusion

One of the disadvantages of single sensor measurement in ambient assisted living is limited spatial coverage [Elm01, p. 4–5]. For example, a single motion detector installed in the living room detects the activity of the user only when the user is actually in the living room. If the user is in their bathroom, kitchen or WC, the sensor in the living room will be unable to detect these activities, and if any unusual activities were to happen outside of the living room, the sensor could equally not detect them. Furthermore, the behavior model from this single sensor is only related to the living room, and what might happen at the other locations cannot be included in the model. The model would thus not be complete for the entire living environment.

In order to describe the behavior of elderly people as completely as possible, a few sensors were placed in each area of the living environment. Useful information could be obtained from different sensor resources, thus increasing the reliability of the sensor data. The statistical behavior models from different sensors installed in different locations (rooms) were fused together in order to learn one complete statistical model system for the entire living environment.

1.3.4 Automatic Scenario Detection

The most important situations and scenarios in the daily lives of elderly people should be detected in a robust way. Here the term "robust" means that similar situations and scenarios should be recognized even if the situations and scenarios have changed slightly. For example, one day the user might get up at 7:00 in the morning, go to the bathroom for 20 minutes, then go to the kitchen for 30 minutes. On another day, the user might get up at 7:30, go to the bathroom for 10 minutes, then stay in the kitchen for 1 hour. These two scenarios are very similar, but occur at different points in time with different durations. Such situations should be treated as different instances of the same scenario. Thus the model would have the ability to detect similar scenarios in a different time frame and/or with a different duration. On the other hand, there are always repeated patterns in the daily living of elderly people, such as getting up in the morning, going to the toilet, showering, having breakfast, cooking in the kitchen, eating lunch, napping, having dinner, watching TV or going to bed at night. These scenarios happen nearly every day, but generally not at the exact same point in time and not with the same time duration. The system should recognize these situations and scenarios even when their time frame changes. If something unusual happens, however, the system should detect it and send an alarm to the caregiver.

1.4 Contribution

From the above discussion about scientific aims we know that the statistical behavior model of the user will be learnt first, then the learnt model will be completed using sensor fusion (SF), and lastly any unusual activity of the user will be recognized based on automatic scenario detection (ASD). But which methods and algorithms will be used to reach these aims?

1.4.1 Unsupervised Learning of Learn Statistical Model

As discussed in the scientific aims, the statistical behavior model of the user must first be learnt. But the problem is that each individual user has different activity patterns resulting from different living habits. For example, one user may get up in the morning at around 7:00, have lunch at about 12:30 and go to bed at about 23:00, but another user may get up in the morning at around 8:00, have lunch at about 11:30 and go to bed at about 22:00 every day. As a reflection of the different living habits of different users, the statistical behavior model should be different in each case. To achieve this, unsupervised learning was used in the project. The methods and algorithms used were chosen with the intent of being able to detect and interpret varying user behavior to learn an appropriate behavior model for each user. This means that the same methods and algorithms may be used for every user, but if the living habits of different users differ, then so will the learnt models.

In the thesis, the Gaussian mixture model (GMM), split-merge algorithm (SMA) and hidden Markov model (HMM) were used to learn the daily statistical behavior model of the subjects. Because they are statistical methods and algorithms, they lend themselves well to the unsupervised learning method. These characteristics make GMM, SMA and HMM powerful tools for dealing with different users with different living habits.

1.4.2 Sensor Fusion to Create a Complete Behavior Model

Using sensor fusion (SF), a complete statistical model of the user's behavior patterns may be learnt. Experimentation in the thesis will begin by using data from only one type of sensor, for example a motion detector, to learn a statistical model of the user. But for a statistical model with more precision, more sensors should be used. For example, 3 motion detectors installed at different locations in the living environment of the user can be used to learn a more detailed behavior model of the user. Alternatively, different types of sensors (for example motion detectors and door contactors) can also be used to learn a more precise behavior model of the user.

For sensor fusion there are several approaches. The main problem is that the result of sensor fusion should be in the form of state data. This state data connects sensor fusion and hidden Markov models (HMMs). Using HMMs, it is possible to realize automatic scenario detection. Within the thesis, appropriate methods will therefore be applied to the sensor and other data types to realize sensor fusion.

1.4.3 Automatic Scenario Detection to Find Unusual Activity

As stated above, the learning results from statistical models (such as the Gaussian mixture model, split-merge algorithm, and hidden Markov models) and sensor fusion should be combined together in order to form complete behavior models and realize automatic scenario detection based on these combined data. In the end result, if the user exhibits abnormal activities, the model should recognize them automatically and send an alarm to the caregiver. How the model recognizes unusual activity by the user will be discussed in the following chapter. The picture 1.4 shows the applied scientific methods.

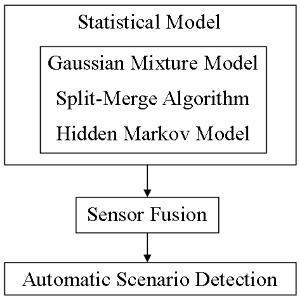


Figure 1.4: Contributions of the work

In the picture 1.4, the rectangles represent the statistical models, sensor fusion (SF), and automatic scenario detection (ASD). It is clear to see that the statistical models of the subject's daily activities will be learnt first, then these statistical models will be combined using sensor fusion (SF) in order to realize automatic scenario detection (ASD). As mentioned previously, different methods and algorithms (GMM, SMA, HMM) will be used to learn statistical models for different types of activities (regular and random).

In this chapter, the aging problem was discussed. Through problem analysis and reasoning, a novel way to realize ambient assisted living was proposed. The concrete methods and algorithms necessary to achieve automatic scenario detection were presented. The state of the art of the statistical model, sensor fusion and automatic scenario detection will be discussed in detail in the next chapter.

2 State of the Art

As this thesis attempts to realize automatic scenario detection (ASD) within ambient assisted living (AAL), the state of the art in technologies and theories concerning AAL as well as current established methods in ASD will be introduced in this chapter.

While various projects from the field of AAL will be introduced in the following, one of the main focal points will be the Artificial Recognition System (ARS) project of which this thesis represents a section. This project, which has its roots in building automation systems (BAS), is currently in progress at the Institute of Computer Technology [ICT] of the Vienna University of Technology and focuses on the development of an intelligent system. It sets great store by the principle of intertwining different scientific disciplines and has achieved good results after several years of work, e.g. the findings of the subprojects ARS-Perception and ARS-Psychoanalysis. Details will be given in the following.

With regard to the field of ASD, appropriate theories and algorithms such as data mining, machine learning and situation and scenario recognition will be described in detail.

2.1 Ambient Assisted Living

All of the ambient assisted living projects discussed in this section relate strongly to the intelligent environment and are rooted in the building automation domain. In [PP05, p. 55–62] the ARS project is regarded as the next step towards an intelligent environment, and in [PLD05, p. 48–55] it is introduced as a set of new concepts for building automation. The ARS project will therefore be highlighted within this thesis as a pioneer project in building automation. Based on the presentation of state of the art AAL principles and projects as well as theories and technologies in the fields of intelligent environment and BAS, the core relationship between BAS and AAL will be extended in the thesis.

2.1.1 Projects about Ambient Assisted Living

The following examples provide an overview of current work in the field of ambient assisted living:

2.1.1.1 In-HAM vzw

The project In-HAM vzw [HAM] is located in Belgium. Test users live in one of 4 home labs for a period of time during which, in cooperation with an advisor and other experts, the living situation of the test users is optimized in a step by step process. The optimizations aim for an "inclusive society", meaning they are focused on removing difficulties for the elderly and disabled in integrating themselves into common society.

Compared to the project in this thesis, the In-HAM vzw test subjects are clearly different. Each of them undergoes a test period between one month and about 4 months, during which time the behavior of the user is observed by different types of sensors. The sensor data is sent to the controller and stored. In this thesis, however, sensor data are analyzed and the behavior model of the user are learnt. Because each test person has an individual behavior pattern and habits, the learnt behavior models for different users should be different. Furthermore, the lives of test persons are not exactly the same from day to day. Every day is a little different compared to other days, and the behavior of the user changes from day to day. So in order to get a general model that includes as many different situations as possible, a long test period is needed.

2.1.1.2 Living Tomorrow

The project Living Tomorrow [LTM] is located Belgium and the Netherlands. This project focuses on building a living environment promoting safety, comfort, and convenience for the user. A flat was developed in which elderly persons can maintain independent living with the help of modern technologies and systems. For example, the use of "smart cooking" in the home can help the inhabitants prepare their meals: sensors are fitted on a smart oven and cook hood which continually measure the temperature and moisture of the food. In case of danger such as burning these appliances are disconnected instantly. Another example is an innovative carpet integrated with an alarm system. When the inhabitant falls on it, thus creating sustained pressure over a large area, the carpet detects the fall and sends an alarm signal to the caregiver.

In contrast to the Living Tomorrow project, this thesis focuses exclusively on the behavior of the elderly. No special appliances were installed in the living environment. Only sensors such as motion detectors, door contactors, accelerometers, light sensors, and temperature sensors were used. Instead of a special carpet, accelerometers were installed in the living environment, and in the case of the inhabitant falling on the floor, these accelerometers detect the situation and alert the caregiver.

2.1.1.3 HomeLab

The Philips HomeLab project is located in the Netherlands [HLB]. Its essential features are human voice recognition and the creation of digital fantasy environments. The system can understand voice commands by the user and respond by recording a voice mail, making it possible for the user to e.g. enjoy music everywhere in the home on multiple home devices. Embedded technology is also built into household appliances. For example, a mirror in the bathroom becomes an interactive user interface that consolidates multiple devices into a single system for managing typical digital activities: recording voice mail, watching a video or listening to music in any room in the home. The HomeLab project is devoted to developing a modern home with many innovative appliances.

Compared to the HomeLab, only simple sensors were used in this thesis. One of the ideas of the thesis was to build a system with low costs and easy installation. The system should be affordable by families with ordinary incomes.

2.1.1.4 SSP

The project SSP (Senior Social Platform) is from Austria [SSP]. The aim of this project is to develop a social networking system in order to reduce the isolation and loneliness of senior citizens. For example, SSP developed a special human-machine interface. After registration, this interface can be used to send emails, log in to one's profile, search for friends on the internet, make contact with other social networks, friends and family, or upload pictures. These kinds of activities can increase social contacts for the elderly and reduce their loneliness.

In contrast to the SSP project, this thesis focuses on the development of a system that acts as an invisible butler. Sensors were installed under the table which is used for meals, or on the walls in the rooms. The user need not interact with the system. Instead, it observes the user's behavior day and night, analyzes it and learns a behavior model for the user. In case of unusual activities, the system detects the abnormality and automatically sends an alarm signal to a caregiver.

2.1.1.5 Inhaus

The inHaus project is located in Germany [HIS]. There are two components included in the project: inHaus1 is concerned with smart homes and inHaus2 with smart buildings. The project focuses on the development of intelligent room and building systems to be established in nursing homes, hospitals, offices and hotels. Different kinds of concepts and solutions have been developed within this project which spans many different engineering disciplines.

While inHaus seeks development of solutions for institutional and business facilities, this thesis is devoted to developing a system used in residential housing. Based on the physiological and psychological characteristics of some elderly people (for example movement and memory disorders), the system was designed to as be easy as possible to apply. Nothing should be activated by the user, and in case of danger the system should send an alarm signal to a caregiver automatically.

2.1.1.6 FZI Living LAB

The FZI Living LAB project is also located in Germany [FZI] and is devoted to the development of assistance functions, such as giving the elderly reminders to take medication and recognizing emergency situations. A hardware level, service level and process level will be introduced in the system.

When compared to the FZI Living LAB, this thesis has some similar goals. For example, the split-merge algorithm (SMA) was used to learn the point in time when the user takes tablets every day. If the user forgets to take the tablets on a particular day, the system sends a reminder signal to the user. Futhermore, hidden Markov models (HMM) were used to learn the behavior model of the user, and if unusual or emergency situations occur, the system recognizes them and sends an alarm signal to a caregiver.

2.1.1.7 Aware Home

The Aware Home project was developed in the USA [AWH] and focuses on the development of home tools for the future, especially new sensing technologies and algorithms for home robotics. For example, tools have been designed for activity characterization, behavior tracking, and understanding of activity context.

In contrast to Aware Home, this thesis focuses on automatic scenario detection based on the learnt behavior model of the user. Here the term "behavior model" refers to a model of the daily activity routines of the user, where different routines have different characteristics. Using hidden Markov models (HMM), each routine is expressed as a sequence of consecutive states.

2.1.2 The Project Artificial Recognition System

The Artificial Recognition System (ARS) project was initiated in the year 1999 by the Institute of Computer Technology (ICT) of the Vienna University of Technology under the direction of Prof. Dr. techn. Dietmar Dietrich and is rooted in building automation systems. In [DS00, p. 343–350], the authors point out that fieldbus systems and the human nervous system have similar characteristics. A biological system was thus introduced into a building automation system for the first time. The essential target is to develop a "controller" using a biological model to deal with massive amounts of sensor data.

The ARS project is not based exclusively on mathematics and computer science. It is a completely new method which also takes advantage of neurology, psychology, pedagogic theory and psychoanalysis.

Using the workings of the human mind as reference and consulting Sigmund Freud's id, super-ego, and ego model, the objective is to construct a "controller" capable of processing massive amounts of sensor data through a correspondingly complex network.

The resulting Artificial Recognition System based on perception and psychoanalysis has developed rapidly in recent years. Related projects are the Smart Kitchen project, the Building Assistance System for Safety and Energy Efficiency project, the Smart Embedded Network of Sensing Entities project, and the PsychoAnalytically Inspired Automation System project.

2.1.2.1 The Artificial Recognition System-Perception Project

"Perception" of real raw data from sensors requires the extraction of essential information from these data, a method which is especially useful when there is a massive amount of data output. The authors in [PLD05, p. 48–55], [Pra06, p. 129] and [PDHP07, p. 21–32] proposed such a perception model featuring sensor values, micro symbol layer, snapshot symbol layer, representation symbol layer, and application.

The first step changes the sensor values to symbols. In this step a data mining algorithm is used to recover patterns and/or symbols from the sensor data. As a result the data volume is reduced significantly.

The second step goes from the micro symbol layer to the snapshot symbol layer. The number of symbols is reduced and each symbol has a function representing the information pertaining to the situation or object(s) of interest.

In the representation symbol layer which the symbols come from, the snapshot symbol layer fuses snapshots of information from different instants of time. The symbols at this level thus contain not only the information from the current moment, but also information from one or more earlier moments. Fusion of several data snapshots over a time period allows deductions to be made about the dynamic processes underlying the data encoded in the symbols. At this point scenario detection can be interpreted at the application level.

The following is an example of the entire process: an elderly person walks into a room and closes a window. The system receives data from the floor and motion detector sensors which are then sent to the micro symbol layer. Through sensor fusion, the system knows that something is moving in the room but does not know what the "something" is, as it cannot distinguish whether an elderly person, a child or a pet is present. The data from the floor sensor and motion detector does not include this kind of information. This happens in the micro symbol layer. From here the data is converted to symbols and the amount of data is reduced. The symbols are sent to the snapshot symbol layer and fused again according to the predefinitions. In this example case, the information about the elderly person is added to the system, allowing it to recognize that there is an elderly person moving in the room. Later these symbols are transferred to the representation symbol layer. Together with the sensor data from the switch and the predefinitions, the system then knows that the person is closing a window. This happens in the representation symbol layer. Using all the processes from micro symbol layer to representation layer, the system works with reduced symbols and predefined rules to correctly interpret a situation.

Thus the task of perception is achieved, but the data processing is not yet finished. As an "intelligent controller", the system is required to have the capacity to control itself, and to have the ability to make decisions to react to changing situations. Psychoanalysis is used to achieve this goal.

2.1.2.2 The Artificial Recognition System-Psychoanalysis Project

The Artificial Recognition System-Psychoanalysis project, based on Sigmund Freud's id, ego and super-ego model, references human emotions as a rating system to construct a decision making process in order to realize the interaction between the environment and an autonomous system.

Compared with the traditional "controller" in the control system, here the "decision making process" not only has the ability to react to outside changes but is also aware of the inside of the "process" itself.

For example, the traditional controller in a building automation system (BAS) uses sensor data and setpoint value differences as input, processes these data and sends the correcting values to the controlled devices. In this way it controls the BAS and its environment. In the developed psychoanalysis system the "decision making process" plays the role of the "controller". It can not only react to sensor data and setpoint values and send control information to control devices, but also has the ability to be aware of itself. For example, if the situation is difficult to control, it can feel negative emotions, while if the whole system is working in harmony, it may perceive contentment.

It should be said that the attempt to develop such a system is a truly courageous one, for up until now there is no tried, tested and widely recognized "correct model" for the human psyche itself. Sigmund Freud's id, super-ego, and ego model does not perfectly reflect a person's inner world, so it is obvious that scientists are faced with two areas to work in: not only the construction of a "decision making process" according to Sigmund Freud's id, ego and super-ego model, but also the process of improving and supplementing the model itself.

2.1.3 Survey of Ambient Assisted Living

In this section the state of the art in research on ambient assisted living (AAL) will be introduced. In [CAJ09, p. 277–298], the authors provide a very good survey of the technologies that comprise ambient intelligence. They focus on technologies that assist in sensing, reasoning, and acting. The picture 2.1 shows the relationships between these components. At the top of the picture 2.1 is the topic - ambient assisted living. Underneath are the three components comprising AAL: sensing, reasoning, and acting. The component reasoning includes various sections: modeling, behavior prediction and recognition, decision making, and spatial and temporal reasoning. In the following, explanations of different components in the picture 2.1 will be presented and several papers dealing with AAL will be introduced.

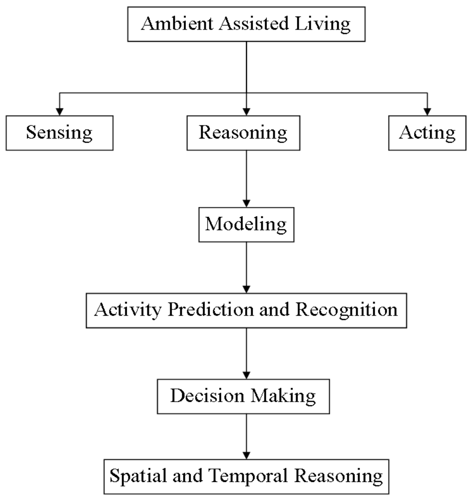


Figure 2.1: Survey of ambient assisted living abstracted in ([CAJ09, p. 277–298])

2.1.3.1 Sensing

The authors in [CAJ09, p. 277–298] state that collecting sensory data from the real world is the first element of sensing. Motion sensors may be used to detect people or persons can wear sensors in order to help a system in tracking them. Structuring this sensor data is another task of sensing. This task is complex and difficult due to the generally large volumes of multidimensional data which include data noise and missing values. So the sensor data must be filtered, disambiguated, and interpreted before it can be used.

2.1.3.2 Reasoning

In the picture 2.1 the interface between sensing and acting is reasoning. User modeling, behavior prediction and recognition, decision making and spatial-temporal reasoning are included in the reasoning process. In the following, these four components of reasoning will be introduced.

As the term indicates, user modeling intends to model the user's behavior. The created model is treated as normal behavior, and interpreting behavior data based on the model allows detection of unusual behavior by the user. If any behavior by the user is outside the range of the model, this behavior will be treated as unusual. Another major function of reasoning is behavior prediction and recognition, which may also be realized based on the user's behavior model. Decision making is another topic of reasoning, and neural networks and fuzzy rules are applied in this process. Spatial and temporal reasoning means interpreting where and when the behavior of the user takes place.

2.1.3.3 Acting

In a smart home, different types of sensors are used. Light, sound, temperature, contact, and motion are measured. In order to receive reliable information from the different types of sensors, sensor fusion was used in [CAJ09, p. 277–298]. Based on such reliable information, an ambient intelligent system can carry out executive actions.

2.1.3.4 References about Ambient Assisted Living

A health integrated smart home information system was introduced in [DVD+02, p. 673–682]. When a new patient moves into the living environment featuring the developed home information system, a new automatic learning period is required due to the individuality of each patient's behavior and activities. A correlation between the rigorous information produced by the system and clinical reality is then performed by a physician who certifies the events occurring in the home information system and decides whether the patient may return to his or her own home. The system features three levels of automatic measuring: the circadian activity, the vegetative state, and some state variables specific to certain organs involved in particular diseases. Location sensors are placed in each room of the home information system, allowing monitoring of the patient's successive daily activity phases within the home environment. These define different confidence zones in which the behavior of the patient can be considered as "normal" or "abnormal".

The authors in [MVWG07, p. 74–94] determined hazards within a smart house environment using an emotive computing framework. Representing a hazardous situation as an abnormal activity, they modeled normality based on the concept of anxiety using an agent based probabilistic approach. Interactions between the user and the environment were detected using multi-modal sensor data. The authors explored a more comprehensive integration of this multi-modal sensor data, in the form of simple sensor (such as pressure pads and reed switches) and audio data, into the anxiety framework. The novelty of this approach is threefold. Firstly, an interaction-based probabilistic model for hazard detection within a smart house environment was introduced and formalized. Anxiety represents an emotive model that is scalable and independent of activity sequences. Secondly, a robust detection of foreground audio events was used for activity detection in assisted living. This enabled the integration of event detection by audio into the anxiety framework demonstrating (a) the extended functionality of the model and (b) the fusion of multi-modal sensor information within the framework. Thirdly, a personal digital assistant as a third form of sensor was incorporated, providing interactive feedback from the user.

In terms of actual applications in AAL, [WSB+09, p. 710–713] discussed two cases: one was the topic of robust fire detection and the other was fall detection. For the former, a combined approach using optical detectors and gas sensors was used. Conventional fire detection as it is currently applied uses an optical detection system in almost all cases, which sets off an alarm when a certain amount of light is scattered out of the optical pathway. The approach in [WSB+09, p. 710–713] was to combine conventional optical detectors with gas sensors of various types in order to achieve greater robustness and a drastically reduced false alarm rate.

For the latter case, three-dimensional detectors based on laser interferometers were tested together with accelerometer-based detectors which had to be attached to the test subject. This, however, was only necessary for testing and training procedures and could ideally be removed when the fall detection system with the three-dimensional detectors mounted in the rooms of the flat/home had been installed successfully.

Time series sensor data from a motion detector were used in [YB10a, p. 6] and [YB10b, p. 3–7]. First the data amount was reduced using a predefined threshold value and translated to “states” in predefined time intervals. Then a hidden Markov model, the forward algorithm and the Viterbi Algorithm were used to learn the person’s daily behavior model. The results indicated that the best matching routine in the model would be detected based on observation sequences, meaning that the learnt behavior model had the ability to distinguish different observation sequences. However, in both abovementioned papers the sensor data came from a single sensor. Furthermore, how to judge whether an observation sequence is normal or not was not discussed in these papers.

In [YB10c, p. 4], time series sensor data from a medical box contactor and a meal entrance contactor were used. Using a Gaussian mixture model and the split-merge algorithm, the sensor data gathered over a period of about one and a half months was analyzed and the statistical behavior model of the user constructed. This paper only introduced how to learn a behavior model utilizing the split-merge algorithm but did not discuss possibilities of using the model to detect unusual behavior.

The paper [YB11c, p. 197–204] introduced a method which reduced the work needed to research the observation symbol probability distribution by according the limited output of distinct observation symbols per state. Furthermore, the forward algorithm was used to calculate the probability of an observed sequence based on the learnt model. Just like in [YB10a, p. 6] and [YB10b, p. 3–7], this paper only used data from a single sensor, and also did not discuss possible methods for judging whether an observation sequence is normal or not.

In order to increase the accuracy of the behavior model (by increasing the amount of state data), the author in [YB11b, p. 5] altered the parameters of a data translator, thereby producing two behavior models with higher accuracy. This change in the amount of state data also caused changes to the structure of the learnt behavior model. States from repeated routines of the behavior models were merged together, thus comparing the different models and discovering behavior trends of the user. Furthermore, hidden behavior could be detected using these more accurate models. The paper began to research how changes to the parameters of the hidden Markov model influenced the structure of the learnt behavior model. However, the employed data still came from a single sensor.

In [YB11a, p. 529–534], data correlation was used to deal with the data and detect the relationships between sensors. In a predefined time interval, the data from different sensors were correlated, and the huge amounts of data were thus translated to comparatively few correlation parameters between sensors. Recorded changes in the values of these correlation parameters between sensors in different time intervals meant that different activities of the user were being detected by the system. Unlike the previously mentioned publications which only used data from a single sensor, this paper began to use data from different kinds of sensors.

Most of the work in the abovementioned papers was limited in two respects: only data from a single sensor was used (not including the work in [YB11a, p. 529–534], where data from different kinds of sensors were correlated), and no method was introduced to detect unusual behavior based on a hidden Markov model.

2.1.4 Building Automation Systems for Ambient Assisted Living

Due to demographic changes and aging in Europe, the proportion of senior citizens within the population is continually increasing. According to "Ambient Assisted Living - Country Report Austria", 21.1% of the Austrian population was older than 60 in the year 2001, and it was estimated that in the year 2030 this value will have increased to 32.1% [ACR, p. 3].

In other European countries the situation is similar. This is obviously a challenge for the citizens and the social and health-care systems, but at the same time also offers opportunities for various industries and the European market as a whole. The Ambient Assisted Living Joint Programme [AAL] was launched to deal with these challenges and opportunities. The programme focuses on enhancing the quality of life of elderly people while at the same time strengthening the affected industries through the use of information and communication technologies.

The Ambient Assisted Living Joint Programme was founded in 2007 as an international association with research and development funding. The background of the programme is the ongoing problem of the aging of the population, and the key focus is to help affected people with new technologies. As Viviane Reding, then EU Commissioner for Information Society and Media, said: "There is no reason for older people in Europe to miss out on the benefits of new technologies. The solutions and services resulting from this programme will help them to remain active in society as well as staying socially connected and independent for a longer time" [AEU].

The following elements of AAL are found in [AEU]: extending the length of time people can live in their preferred environment by increasing their autonomy, self-confidence and mobility; supporting health maintenance and functional capability of elderly individuals; promoting a better and healthier lifestyle for individuals at risk; enhancing security to prevent social isolation and maintaining the multifunctional network around the individual; supporting carergivers, families and care organisations; increasing the efficiency and productivity of resources used in aging societies.

All of these contents can be classified and transferred to the components of automation control systems. They are the fundamental demands for AAL, but these demands need to be translated into the language of technology and implemented in real, practicable solutions.

The picture 2.2 indicates the relationship between building automation systems and AAL. In the center of picture 2.2 are the sensor, controlled device, controller, and network. These are the basic components of building automation systems. In picture 2.2, the components are connected with each other by thick black lines with arrows. Each of the components corresponds to certain features of AAL, which are represented by the contents of the ovals. The features are connected to the basic components of building automation systems by dotted lines with arrows. Explanations of these features follow.

Sensors used in AAL systems are ideally mounted in non-invasive or invisible ways, and should be distributed within the respective environment (such as the home, outdoors, vehicles, public spaces, ...) or directly integrated into appliances or furniture. These sensors are intelligent sensors capable of collecting, processing and analyzing data.

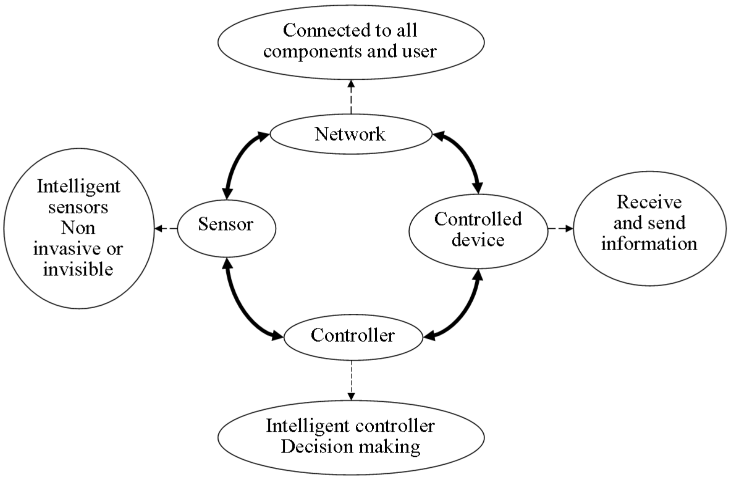


Figure 2.2: Components of building automation systems for ambient assisted living

Controllers used in AAL are intelligent controllers. They use information from sensors, actuators and other sources, can function locally or remotely, and have decision-making abilities catered to the situation. For example, they can send alarm information to a caregiver or call center directly and instantaneously.

The intelligent network system in AAL operates in a dynamic situation. Sensors and actuators are therefore connected to each other, and this connection should be maintained even when the user leaves his or her home and goes outside.

The controlled devices in AAL systems play important roles: in an automation system, the basic components (sensor, controller, network, and controlled device) interact with each other and the system as a whole interacts with the environment. The controlled device receives information from the sensor or controller through a dynamic network. Furthermore, a controlled device may also send information back to the sensors or controller.

In summary, all the components of an automation system work together. Based on user information, environmental information and control center information, the system recognizes situations, makes decisions and implements actions. The ultimate goal is to sustain a comfortable and secure lifestyle for the user. The system should be able to cope with sudden unexpected events, for example if the user faints and falls to the floor; or trips, hits their head and becomes unconscious. In such cases, the system must respond rapidly and send alarm information to a caregiver or emergency center, thereby getting assistance for the user with the minimum of delay.

Several examples which relate to AAL will be given in the following.

Barrett hand: The Company Barrett Technology Inc., located in Cambridge, USA has developed human-like products with a special focus on human-like hand and arm prostheses. Details can be found in [BAH].

Robotic neuro-rehabilitation: In [FBJ06, p. 650–659], the authors developed a new method of robotic manipulation which for example allowed a robot to grasp, lift and move objects by interpreting sensory stimuli and using them to interact intelligently in real situations.

Neural-machine interface: In [GNV+08, p. 315–320], the authors used human electroencephalography signals to control an intelligent wheelchair.

Real-time control technology: Recently another real-time control technology was developed which also uses human electroencephalography (EEG) signals to control a wheelchair. The most interesting point is that the time delay between the EEG signal sending a message and the wheelchair reacting to the message is as short as 125 milliseconds. More information can be found in [RTC].

Smart home: In [HMEZ+05, p. 64–74], the authors introduced a house equipped with different intelligent sensors, such as smart laundry, smart floor, smart display, and home safety monitor. They used new technologies to achieve a high level of home automation.

2.2 Situations and Scenario Recognition

In the following, the state of the art in situation and scenario recognition will be presented and discussed. Scenario recognition is becoming more and more important in detecting sequences of events and reacting to them properly [LBVD09, p. 8]. In order to detect sequences of events, there must be patterns within these events. There are at least two concepts of scenario recognition: predefined templates and the application of an unsupervised learning algorithm using statistical methods [LBVD09, p. 8]. The topic of this thesis are the elderly, and as their activities and behaviors within their living environment are largely random, it is difficult to predefine templates. This thesis therefore focuses on using statistical methods and unsupervised learning algorithms to determine activity patterns and learn an appropriate behavior model.

The state of the art in machine learning will also be introduced and discussed in order to determine appropriate algorithms for pattern detection and model building. But because the activities and behaviors of the elderly in their living environment are mostly random, it must be assumed that the activity variables of the user will be dynamic, and that large amounts of data will be created. Therefore a data mining process will be required prior to the machine learning process in order to reduce the anticipated data amounts.

In the following, data mining will be introduced first, then machine learning and the algorithms for user behavior model building will be discussed. Finally, the state of the art in situation and scenario recognition will be presented in detail.

2.2.1 Data Mining

Definitions for data mining can be found in relevant literature, such as "data mining (DM) is the application of specific algorithms for extracting patterns from data" in [FPSS96, p. 39], or "knowledge discovery and data mining (DM) are techniques to discover strategic information hidden in very large databases" in [GG99, p. 20]. The state of the art in data mining, a technology used in various domains, is abstracted from the following papers.

The picture 2.3 shows different methods and algorithms for data mining: non-derivable frequent itemsets [LC09, p. 481–498]; the individually adaptable perturbation model [LKT08, p. 5–21]; the apriori-like group pattern mining algorithm and valid group-growth algorithm [WLH06, p. 240 282]; top-K frequent closed itemsets of data streams [Li09, p. 10779–10788]; temporal high utility itemsets-mine [CTL08, p. 1105–1117]; and spatio-temporal data mining [HGFR06, p. 192–214]. Each of these approaches will be discussed briefly below.

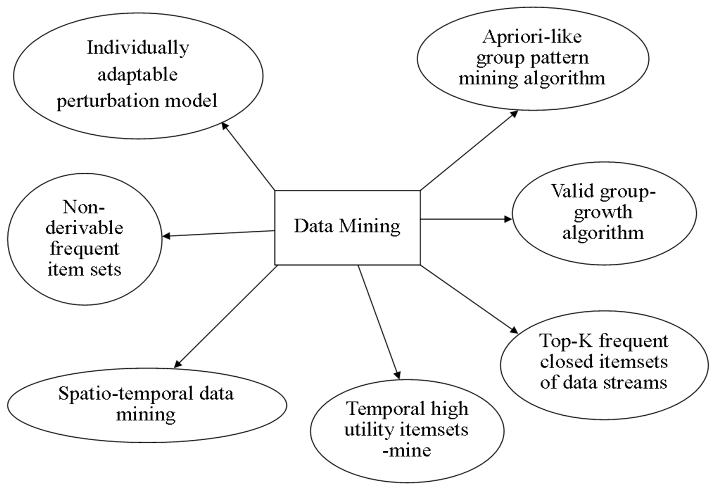


Figure 2.3: Methods and algorithms of data mining

The authors in [LC09, p. 481–498] designed a compact data structure named non-derivable frequent itemsets to efficiently maintain a dynamically selected set of itemsets. In non-derivable frequent itemsets, the nodes are divided into four categories to reduce the redundant computational cost based on their properties.

The individually adaptable perturbation model, which enables individuals to choose their own privacy levels, was proposed by the authors in [LKT08, p. 5–21]. They focused primarily on the perturbation techniques which are usually used in scenarios where individuals can perturb their private data with some known random noise. The authors presented a novel two-phase perturbation method for numerical data that allows individually adaptable privacy protection.

The authors in [WLH06, p. 240–282] presented a new approach to deriving groupings of mobile users based on their movement data. They utilized user movement data collected by logging location data emitted from mobile devices tracking users, and formally defined "group pattern" as a group of users who were within a distance threshold from one another for at least a minimum duration. They then proposed a framework that summarized user movement data prior to group pattern mining. Two algorithms, the apriori-like group pattern mining algorithm and the valid group-growth algorithm, were developed to mine valid group patterns. While the apriori-like group pattern mining algorithm was derived from the apriori algorithm for classical association rule mining, the valid group-growth algorithm adopted a mining strategy similar to the frequent pattern-growth algorithm and was based on a novel data structure known as valid group graph.

The authors in [MMCG09, p. 1224–1236] discussed the design and safety requirements for large-scale privacy-preserving data mining systems in a fully distributed setting where each client possesses its own records of private data.

An efficient single-pass algorithm, entitled top-K frequent closed itemsets of data streams, was proposed in [Li09, p. 10779–10788] for mining a set of top-K closed frequent itemsets from data streams within a transaction-sensitive sliding window. An effective data structure, called closed itemset lattice, was developed to maintain the essential information about the current set of closed itemsets from data streams. The experimental results showed that the proposed algorithm achieved high accuracy, lower memory usage and fast execution time.

A novel method, namely temporal high utility itemsets-mine, was proposed by the authors in [CTL08, p. 1105–1117] for mining temporal high utility itemsets from data streams efficiently and effectively. It is the first work on mining temporal high utility itemsets from data streams. The novel contribution of temporal high utility itemsets-mine was that it could effectively identify the temporal high utility itemsets by generating fewer candidate itemsets, thus reducing execution time substantially when mining high utility itemsets in data streams. In this way, the process of discovering all temporal high utility itemsets under all time windows of data streams could be achieved effectively and faster than before while using less memory. This meets the critical requirements on time and space efficiency for mining data streams.

A method for spatio-temporal data mining based on GenSpace graphs was described in [HGFR06, p. 192–214]. Using familiar calendar-based and geographical concepts, such as workdays, weeks, climatic regions and countries, spatio-temporal data could be aggregated into summaries in many ways. The developed method automatically searched for a summary with a distribution that was anomalous (far from user expectations). According to current expectations the possible summaries were repeatedly ranked and the user was allowed to adjust his or her expectations. Furthermore, a propagation path in the GenSpace subgraph was chosen which reduced the storage and time investment for the mining process.

The above methods and algorithms show different research ideas about data mining based on different data types. The used data in this thesis was collected by various sensors installed in a smart environment, and each datum could assume only the values 0 or 1. If one sensor sends one data unit each minute, it will send 1440 data units (60×24) in one day. If the same sensor sends one data unit each second, there will be 86400 data units (1440×60) per day, and if a total of 100 sensors are installed in the system, there will be 8.64 million data units (86400×100) sent daily. If a particular building is very large, even more sensors would need to be installed, and the data amount would increase yet more. This is a primary problem in a smart environment with many sensors.

The second problem with such large amounts of data is that not all the collected values are equally important and useful. This means that while data from certain types of sensors may be rare but deliver important information, other types of sensors may send far more data which, however, may contain comparatively little useful information. Thus how to interpret the various collected data is another important challenge.

In view of the different types of sensors and the differing data amounts occurring in this work, appropriate data mining methods had to be selected. Based on the state of the art in data mining, this thesis therefore integrates several ideas and utilizes various data mining methods and algorithms.

2.2.2 Machine Learning

The author in [Alp04, p. 3] defines machine learning thusly: "machine learning is programming computers to optimize a performance criterion using example data or past experience". The author in [Nil98, p. 1] states that "machine learning usually refers to the changes in systems that perform tasks associated with artificial intelligence".

The picture 2.4 shows the different algorithm types used for machine learning, such as supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, transduction, multi-task learning, and inductive reasoning.

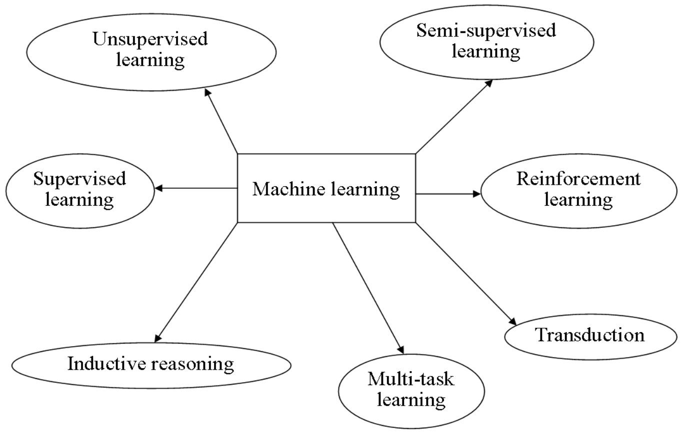


Figure 2.4: Types of machine learning

As mentioned in Chapter 1, unsupervised learning was used to learn a behavior model in this thesis. Major parameters of unsupervised learning are that the input data is unlabeled, and clustering and independent component analysis are important forms. Clustering (cluster analysis) in particular is a typical method of unsupervised learning. This method assigns the input data to different subsets (called clusters), so that the packets of data which make up a given cluster are similar. Here similar means that the features of the data are similar; for example, the distance between the data in the same cluster is shorter than the distance to the data in a different cluster. In a dynamic system the input data can be either merged together or split into different clusters whose numbers may vary. The Gaussian mixture model is commonly used in the clustering algorithm. A detailed description of the clustering algorithm can be found in [YBZ09, p. 6].

Many different methods and algorithms for machine learning have been introduced in various publications. The following papers represent the state of the art in machine learning.

In [Mar09, p. 260–266], the author examined how habituation can be mathematically modeled and discussed how well such models fit the revised characteristics of habituation. He then demonstrated how the models could be combined with neural networks in order to realize various applications. Numerical models of habituation were considered as well as how and why they should be used in artificial intelligence and machine learning. The author coupled standard neural networks with simple models which demonstrate some of the characteristics of habituation and provided experimental evidence to prove that the resulting system did in fact exhibit the expected behavior.

A clustering-based machine learning algorithm (named the clustering algorithm system) was introduced in [AOJ06, p. 248–258]. The first approach in this paper was learning from examples, and it was found that the clustering algorithm system supported single and multiple inheritance and exceptions. It also avoided probability assumptions which are well understood in concept formation. The second approach was learning by observation; here the clustering algorithm system applied a set of operators proven to be effective in conceptual clustering. The authors showed how the clustering algorithm system built and searched through a cluster's hierarchy to incorporate or characterize an object.

The authors in [MZB07, p. 922–937] presented a novel approach to machine learning, which they called argumentation based machine learning. This approach combined machine learning from examples with concepts from the field of argumentation. The idea was to provide expert's arguments or reasons for some of the learning examples. These arguments could constrain the combinatorial search among possible hypotheses, and also direct the search towards hypotheses which were more comprehensible in light of expert's background knowledge. The authors realized the idea of argumentation based machine learning in the form of rule learning. Usually the problem of learning from examples is stated thus: examine given examples and find a theory that is consistent with the examples. Each example is specified by an attribute-value vector and the class to which the example belongs. In the setting in [MZB07, p. 922–937], arguments were used to explain (or argue) why a certain learning example should be in the class as given (arguments for the class are called positive arguments) or why it should not be (arguments against are called negative arguments). Examples that were accompanied with arguments were termed argumented examples.

In [MA98, p. 99–123], the authors discussed important approaches to inductive learning methods such as propositional and relational learners, with an emphasis on methods based in inductive logic programming, as well as approaches to lazy methods such as instance-based and case-based reasoning. Inductive learning methods are typically used to acquire general knowledge from examples. Lazy methods are those in which the experiences are accessed, selected and used in a problem-oriented way.

Using a supply chain network, the authors in [CEMW07, p. 174–193] demonstrated the feasibility and robustness of employing machine learning and genetic algorithms to appropriately model, understand, and optimize data intensive environments. Employment of these algorithms, which learn from and optimize data, can obviate the need to perform more complex, expensive and time consuming experiments which often disrupt system operations. The structure of the proposed algorithms consisted of two major components: (a) training a machine learning algorithm (polynomial support vector machine) to model a supply chain, and (b) applying a genetic algorithm to obtain input settings that yield optimum system performance (the optimization process).

The authors in [ROTZ08, p. 359–366] addressed the architectural design of the extreme learning machine classifier network, where the employment of hidden nodes which are too small or too large leads to underfitting/overfitting issues in pattern classification. In particular, they presented a pruned extreme learning machine algorithm as a systematic and automated approach for designing an extreme learning machine classifier network. The pruned extreme learning machine provided a systematic approach for designing the network architecture of the pruned extreme learning machine classifier. Using statistical methods to measure the relevance of each hidden node in contributing to the prediction accuracy of the classifier, the appropriate architecture of the classifier network could thus be defined.

In this thesis the Gaussian mixture model (GMM), split-merge algorithm (SMA), and hidden Markov models (HMM) were used. These methods belong to the realm of unsupervised learning algorithms. Unsupervised learning algorithms are well suited to the dynamic variables created by the random activity of the user for two main reasons. Firstly, these unsupervised learning algorithms have a deviation parameter, meaning a tolerance for the encountered random variables. Secondly, the GMM, SMA and HMM are statistical methods which allow a relatively long time interval (for example some days) to learn and analyze the random variables.

The publications cited in the following represent the state of the art in the use of the Gaussian mixture model, split-merge algorithm and hidden Markov models. This section deals with the mathematical background of these algorithms and models.

2.2.2.1 Gaussian Mixture Model

The authors in [CWF04, p. 2626–2639] proposed a self-splitting Gaussian mixture learning algorithm for Gaussian mixture modeling to be applied to speaker identification, while the author in [UNGH98, p. 274–283] presented a split and merge expectation-maximization algorithm to overcome the local maximum problem in Gaussian mixture density estimation. In [RG97, p. 731–792], the authors developed a new methodology for fully Bayesian mixture analysis, using reversible jump Markov chain Monte Carlo methods to jump the parameter subspaces and the different numbers of components in the mixture. A standard Gaussian function is defined as

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.1}$$

Here μ is the expected value (mean value), σ is the standard deviation, x is the value in Gaussian function. If there are different mean and standard deviation values in the Gaussian function, so the Gaussian function is multivariate Gaussians and could be called Gaussian mixture model (GMM). Just as in chapter 1 introduced that sprit-merge algorithm (SMA) could be used to analyze the mean values, standard deviation values, and the mixture weight of each component (cluster) in Gaussian mixture model.

2.2.2.2 Split-Merge Algorithm

The paper [MH05, p. 203–210] proposed a new kind of dynamic merge-or-split learning algorithm to deal with the selection of number of Gaussians in the mixture, and the authors in [ZCSC03, p. 1973–1983] introduced a split-and-merge operation in order to alleviate the problem of local convergence of the usual expectation-maximization algorithm. In [SHP08, p. 1–2] and [XFJ06, p. 838–842], the authors used cluster analysis methods for video image analysis; in particular, in [SHP08, p. 1–2] hidden Markov model-based clustering for learning motion patterns over time and detecting abnormal activity in a video surveillance scene was described. The authors in [XFJ06, p. 838–842] presented a novel dynamical Gaussian mixture model for tracking elliptical living objects in video frames, where the parameters describing each object's position and shape were analyzed using a Gaussian mixture model. Finally, in [NM06, p. 761–774] the authors proposed novel methods for evaluating the performance of object detection algorithms in video sequences, proposing region splitting or merging.

Before using split-merge algorithm (SMA) to cluster a set, the ranges of the parameters must be defined. If the predefine number of initial components $S = 3$, that means there are sets $\{\mu_1, \mu_2, \mu_3; \sigma_1, \sigma_2, \sigma_3; P_1, P_2, P_3\}$. Here P is the percent value of each component (mixture weight), $\Sigma \mathrm{P}_{(1,2,3)} = 1$, and the prior parameters are random variables. The threshold value for split, merge and delete components are $\mu_{threshold}$, $\sigma_{threshold}$, $\sigma_{threshold2}$, $P_{threshold}$. The maximum number of values for adjusting the learning rate is M and the current value count is M'. Each new observation value accepted into the value set T is $T_r (r \geq 1)$. The index s is the component index within the mixture model. With these parameters and definitions, the value set can be clustered using the split-merge algorithm. The following is the brief introduction of the split-merge algorithms (SMA).

Compute and then normalize posteriors $P_s(T_r)$, posterior here means how the components are assigned

$$P_s(T_r) = P_s \times \varphi_{\mu,\sigma^2}(T_r); P_s(T_r) := \frac{P_s(T_r)}{\Sigma P_s(T_r)} \tag{2.2}$$

Compute new means

$$\mu_s := (1 - P_s(T_r)) \times \mu_s + P_s(T_r) \times \frac{M' \times \mu_s + T_r}{M' + 1} \tag{2.3}$$

Compute new variances

$$\sigma_s := (1 - P_s(T_r)) \times \sigma_s + P_s(T_r) \times \frac{M' \times \sigma_s + |\mu_s - T_r|}{M' + 1} \tag{2.4}$$

Compute new priors, prior here means which probability value the component can be obtained

$$P_s := \frac{M' \times P_s + P_s(T_r)}{M' + 1} \tag{2.5}$$

Keep the learning rate and adaptability if

$$M' \geq M, M' = M \tag{2.6}$$

After some initial iterations, start checking if it is necessary to split components: If $\sigma_s > \sigma_{threshold}$, then create new component (index S) from old component (index s). The mathematical form will be further introduced in the next chapter.

If necessary, merge components s' and s''

If $|\mu_{s'} - \mu_{s''}| < \mu_{threshold}$ and $|\sigma_{s'} - \sigma_{s''}| < \sigma_{threshold2}$ then merge component s'' into s' and delete component s''. Here $\sigma_{threshold2} \leq \sigma_{threshold}$. The mathematical form will be further introduced in the next chapter.

If any component's prior decreased too much so that $P(s') < P_{threshold}$ then delete the component and adjust the other priors $P(s)$:

$$P(s) := \frac{P(s)}{\Sigma P(s)} \tag{2.7}$$

From the first step repeats the algorithm with all new values.

The Gaussian mixture model (GMM) and the split-merge algorithm (SMA) with mathematical form [YB10c, p. 4] will be further introduced and discussed in the chapters dealing with this thesis' actual application.

2.2.2.3 Hidden Markov Model

In this work, a hidden Markov model was used for situation and scenario recognition. This is because in this thesis no cameras or microphones were used as sensors, so the behavior of the user could not be directly observed. The behavior of the user could only be indirectly observed with the sensors (such as motion detector, door contactor) and the observation (activity routine) was a probabilistic function of the behavior (state), which is in fact the definition of a hidden Markov model. The state of art in this field will therefore be introduced next.

The picture 2.5 shows the different methods and algorithms of hidden Markov models (HMM) such as: the sequence maximum likelihood estimator introduced in [Ler92, p. 127–143]; the Kalman filter model described in [GCL06, p. 222–243]; an exact likelihood of HMMs in an open interval introduced in [Ler92, p. 127–143]; nonstationary HMMs as utilized in [SK95, p. 31–46]; a hidden semi-Markov model described in [BB06, p. 2192–2209]; finally, the authors in [Li05, p. 977–984] discussed probability optimal, state sequence, and parameter estimation. These methods and algorithms will be discussed briefly in the following.

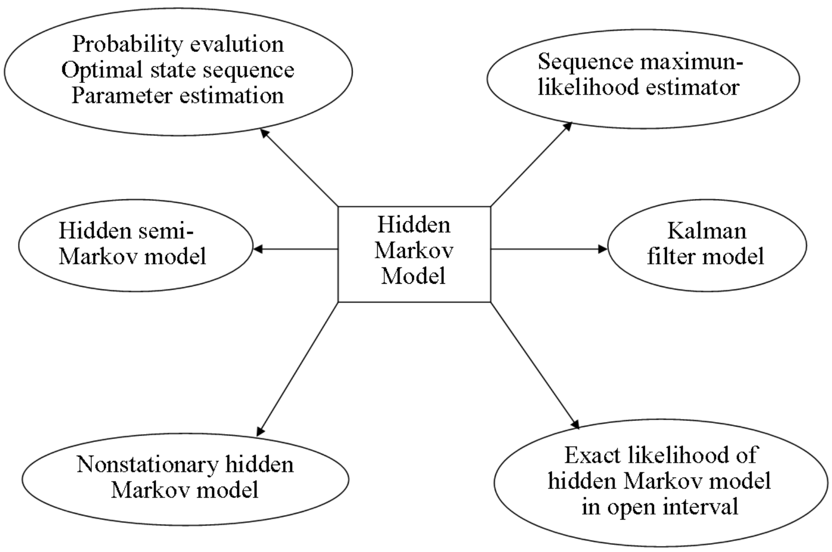


Figure 2.5: Methods and algorithms of hidden Markov model

The publication [Li05, p. 977–984] presented a new type of hidden Markov models in which the current state depends both on the immediately preceding state and the immediately preceding observation, and the state sequence is still a Markov chain. Several new algorithms were proposed and simulated for the three basic problems of interest: probability evaluation, optimal state sequence and parameter estimation.

Hidden Markov models assume a sequence of random variables to be conditionally independent given a sequence of state variables which forms a Markov chain. Maximum-likelihood estimation for these models can be performed using an estimation-maximization-algorithm. In [Ler92, p. 127–143], the consistency of a sequence of maximum-likelihood estimators was proven. The method introduced in [Ler92, p. 127–143] to study the exact likelihood of HMMs was extended to the case where the state variables evolve in an open interval of the real line. Under rather minimal assumptions, the authors in [GCL06, p. 222–243] obtained convergence of the normalized log-likelihood function to a limit that they identified as the true value of the parameter. The method was illustrated in full detail on the Kalman filter model.

The inappropriateness of the standard HMM for capturing state duration behavior has often been pointed out. While explicit state duration modeling in the HMM has been developed, it is not sufficient for modeling the intrinsically dynamic, or nonstationary, transition process. In [SK95, p. 31–46], the authors explored the nonstationarity of Markov chains and proposed a nonstationary HMM defined with a set of dynamic transition probability parameters $A(\tau) = a_{ij}(\tau)$ and a function of time duration τ. Compared to the traditional models, this model was defined as a generalization of the standard HMM and the state duration HMM, with the description being given for discrete observation distributions. Through a set of experiments, it was shown that the proposed model was more capable of capturing the dynamic nature of signals with higher discrimination power in on-line character recognition.

Hidden Markov models reproduce most of the stylized facts about daily series of returns. A notable exception is the inability of these models to reproduce one ubiquitous feature of such time series, namely the slow decay in the autocorrelation function of the squared returns. It has been shown that this stylized fact can be described much better by means of hidden semi-Markov models. These were illustrated in [BB06, p. 2192–2209].

In [TCC09, p. 608–619], the authors suggested employing a HMM to detect machine failure in process control. They proposed models for cases of indistinguishable production units as well as distinguishable production units. Numerical examples were given to illustrate the effectiveness of the proposed models.

A hidden semi-Markov model (HSMM) is an extension of the HMM designed to remove the constant or geometric distributions of the state durations assumed therein. A larger class of practical problems can be appropriately modeled in the setting of HSMMs. A major restriction is encountered, however, in both conventional HMM and HSMM. It is generally assumed that there exists at least one observation associated with every state that the hidden Markov chain takes on. The authors in [YK03, p. 235–250] removed this assumption and considered the following situations: (i) observation data may be missing for some intervals; and (ii) there are multiple observation streams that are not necessarily synchronous to each other and may have different emission distributions for the same state. They therefore proposed a new and computationally efficient forward-backward algorithm for HSMMs with missing observations and multiple observation sequences (O). The required computational effort for the forward and backward variables was reduced to O(D), where D is the maximum allowed duration in a state. Finally, they applied this extended HSMM to the estimation of the mobility model parameters for the internet service provisioning in wireless networks.

Several publications, for example [MT93, p. 6], [Bil02, p. 1] and [Rab89, p. 258] offer definitions of the Markov chain: $P(Q_{t+1} = q_{t+1}|Q_t = q_t, Q_{t-1} = q_{t-1} \dots Q_0 = q_0) = P(Q_{t+1} = q_{t+1}|Q_t = q_t)$. Here Q_t is a random variable from a countable state space at time t, and q_t is the taken variable in a countable set at time t.

In a Markov model, each state corresponds to an observable (physical) event. In this thesis, however, the behavior of the user could not be directly observed (no cameras or microphones were used, so the caregiver could not directly observe the user). Only sensors like motion detectors and door contactors were used. The behavior of the user could be indirectly observed with the sensors (such as motion detector, door contactor) and the observation (activity routine) was a probabilistic function of the state, which is in fact the definition of a hidden Markov model: a doubly embedded stochastic process with an underlying stochastic process that is not itself observable (it is hidden), but can be observed through another set of stochastic processes that produce a sequence of observations [Rab89, p. 259].

There are several parameters which characterize hidden Markov models. For a better understanding, these parameters will be explained briefly in the following.

The number of states N.

The number of output distinct observation symbols each state M.

The state transition probability distribution matrix $A = \{a_{ij}\}$.

$$a_{ij} = p(Q_{t+1} = j | Q_t = i) \tag{2.8}$$

$0 \leq a_{ij} \leq 1$ and $\sum_{j=1}^{N} a_{ij} = 1$, $1 \leq i, j \leq N$. Here Q_t is the current state at time t.

The state emission probability distribution matrix $B = \{b_{ik}\}$.

$$b_{ik} = p(O_t = k | Q_t = i) \tag{2.9}$$

$0 \leq b_{ik} \leq 1$ and $\sum_{i=1}^{M} b_{ik} = 1$, $1 \leq i \leq N$, $0 \leq k \leq M$. Here O_t is the output symbol at time t.

The initial state distribution $\pi = \{\pi_i\}$.

$$\pi_i = p(Q_o = i) \tag{2.10}$$

In Chapters 3 and 4, hidden Markov models and their parameters will be introduced with more detail step by step within the actual application.

2.2.3 Survey of Situations and Scenario Recognition

The activities of a person in his living environment are a central theme of this thesis, and its goal is to discern the activities of a person using a recognition system. To facilitate this goal, situation and scenario recognition were used and will therefore be introduced in the following.

Many publications offer definitions of the term"scenario". In [DSVS99, p. 697], a scenario is defined as a possible interaction sequence between a system and its environment. In [HRD10, p. 326], scenarios are described as a well-established approach to describing functional requirements, uncovering hidden requirements and trade-offs, and validating and verifying requirements. A scenario is specified as a sequence of event predicates that can be enriched with object-based low-level features and directional predicates in [SGU09, p. 1]. It models an expected evolution of the process in [DRF98, p. 144]. Scenarios are defined by a domain expert using a high level language. A scenario models a class of behaviors, a real, previously encountered situation. Finally, in [BFJZ93, p. 117] scenarios are partial descriptions of system and environment behavior arising in restricted situations. Generally speaking, the term "scenario" describes an event or series of actions and events.

The picture 2.6 illustrates the state of the art approaches and algorithms used for automatic scenario detection (ASD) such as: timed automata [DSVS99, p. 697–713]; eleven criteria [HRD10, p. 326 350]; scenario-based query-processing system [SGU09, p. 15]; coarse-to-fine algorithm [ZL09, p. 5976–5986]; decision tree [CW09, p. 4101–4105]; process evolution [DRF98, p. 139–155]; and predefined perception pattern [LBVD09, p. 8]. In the following these approaches and algorithms will be discussed briefly.

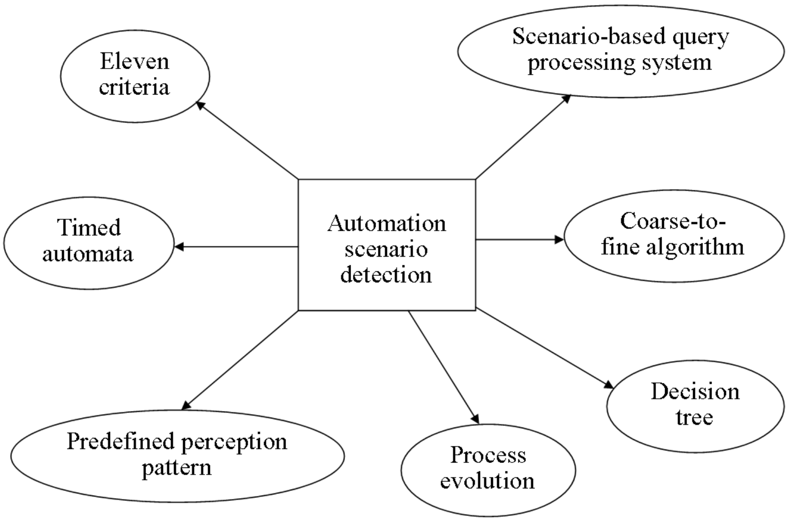


Figure 2.6: Approaches and algorithms of automatic scenario detection

In [DSVS99, p. 697–713], timed automata were used as a target formal method due to the direct relationship between formal interpretation of scenarios and partial runs of the described system. The authors distinguished two composition methods: the declarative composition and the inductive composition. The former was created according to the manner explicitly indicated by the user, and the latter used inductive rules to insert a scenario into an existing specification which may be initially empty. Because time is an important concept in the emerging networks and applications, the authors indicated that the representation of scenarios needed to be able to express time constraints that reflect real-time system situations.

The authors in [HRD10, p. 326–350] proposed a collection of eleven criteria to help categorize and compare many timed scenario notations: timed-action/event enabling; instantaneous (atomic) versus durational actions; absolute versus relative time; system clocks, local versus global; physical versus logical; urgency; time domain; time representation/measurement; time expressiveness (timed constructs/constraints); informal, semi-formal and formal semantics; time analysis and verification; specification executability and tool support. Furthermore, the authors presented a survey of forty-seven time-based construction approaches (corresponding to fifteen timed scenario notations) based on the eleven evaluation criteria.

A scenario-based query-processing system for video surveillance archives was proposed in [SGU09, p. 15]. The authors introduced an inverted tracking scheme which effectively tracked moving objects and enabled view-based addressing of the scene. There are specific situations which can be considered sequences of events, and the existence of such sequences was of interest. Scenario-based query processing was proposed to detect these sequences in video-surveillance archives, thereby reducing the gap between low-level features and high-level semantic content.

In [ZL09, p. 5976–5986], the authors presented a novel scene detection scheme for various video types. They analyzed video shots using a coarse-to-fine algorithm. The key frames containing no useful information were detected and removed using template matching. Spatio-temporal coherent shots were then grouped based on the temporal constraint of video content and visual similarity of activities in the shots.

Decision tree learning was used for freeway automatic incident detection (AID) in [CW09, p. 4101–4105]. In this case incident detection was viewed as a classification problem with two desired output classes: incident and non-incident. California algorithms compared the values of traffic flow with predetermined threshold values using decision tree logic. If the threshold values were exceeded, the existence of an incident was declared. In this research, the decision tree was constructed by learning from the data, and the structure of the decision tree and its node values (threshold values) was automatically generated using data mining techniques. The learnt tree served for generating rules to detect incidents, thus making it entirely different from California algorithms.

A technique for recognizing a session (the clinical process evolution) by comparison against a predetermined set of scenarios (the possible behaviors for this process) was proposed by the authors in [DRF98, p. 139–155]. They used temporal constraint networks to represent both scenario and session. An index of temporal proximity was introduced to quantify the degree of matching between two temporal networks in order to select the scenario best fitting a session.

Furthermore, the authors proposed scenario recognition as a technique for reasoning about time in dynamic systems whose behavior cannot be determined completely by any mathematical model, and applied it to temporal reasoning in medical domains: the time-course of a clinical process was compared to a predetermined set of possible behaviors for this process. These predetermined behaviors were named scenarios. Recognition of a scenario S (or a part of S) implied that the observed time-course of the process, called a session (S), corresponded to S. This recognition allowed the authors to anticipate forthcoming events from the partial instantiation of the recognized scenario, and to intervene in the process, for instance in order to avoid specific expected but undesirable situations.

The authors in [LBVD09, p. 8] presented two concepts of scenario recognition and their implementation: one based on predefined templates and the other applying an unsupervised learning algorithm using statistical methods.

Their first scenario recognition model was based on predefined perception patterns, called image templates. It combined different sensor outputs and gave them semantic meaning. Furthermore, recognized image templates were used as transition conditions between the states of a scenario recognition process based on predefined patterns of possible scenarios. Based on the level of symbolization, two types of templates were necessary to guarantee scenario recognition: the image template, representing a typical set of perceived data within a single moment, and the scenario template, representing a perceived sequence in time. In the operational phase, the system's output was one or more recognized scenarios. Sensory values alone are not sufficient to realize modern scenario recognition that needs a broad overview of information, in particular over longer periods of time.

The second model for scenario recognition followed an approach based on unsupervised learning of behavior patterns. During a learning phase, the system detected and learned all new scenarios that took place so that it could differentiate "normal" scenarios from exceptional scenarios during the operational phase. It learned a set of prototypes for scenarios from the actual data and weighted them according to their frequency. In the operational phase, it classified new data with respect to the prototypes and computed an overall probability for these data.

Automatic scenario recognition has a wide application in many domains, as exemplified in the following paragraphs.

In [SGU09, p. 15], scenario-based query processing was used for video surveillance archives. Automatic scene detection was used for story retrieval in [ZL09, p. 5976–5986]. In [DRF98, p. 139–155], scenario recognition was employed for temporal reasoning in medical domains. Scenario recognition also plays an important role in model building automation systems - an example for this can be found in [LBVD09, p. 8].

One of the questions in automatic scenario recognition is that of "delays". In [DSVS99, p. 697 713], the authors defined two kinds of delays: triggering delays and completion delays. Another issue in automatic scenario recognition is "segmentation", which is discussed in [ZL09, p. 5976–5986].

The picture 2.7 shows the different types of situation and scenario recognition related to ambient assisted living (AAL) and intelligent environments, such as: intertransaction association rule mining as introduced in [LWV07, p. 95–116]; a scenario-based query-processing system as described by the authors in [SGU09, p. 15]; situation and scenario recognition used for security, privacy and the individual's domain in [FVPW07, p. 15–29]; and multi-agent systems as introduced in [BFPLC10, p. 3986–3999]. These approaches will be briefly introduced in the following.

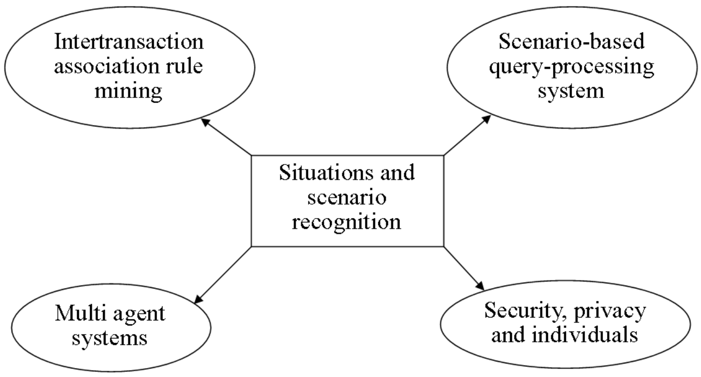


Figure 2.7: Types of situation and scenario recognition

Motivated by a growing need for intelligent housing to accommodate aging populations, the authors in [LWV07, p. 95–116] proposed a novel application of intertransaction association rule mining to detect anomalous behavior by smart home occupants. An efficient mining algorithm that avoided the candidate generation bottleneck limiting the application of current intertransaction association rule mining algorithms on smart home data sets was introduced, and an original visual interface for the exploration of new and changing behaviors distilled from discovered patterns using a new process for finding emergent rules was presented. The authors discussed the observations of emergent behaviors detected in the homes of two real world subjects.

Today, the need for architectures and computational models for large scale open multi-agent systems is considered a key issue for the success of agent technology in real world scenarios. The main goal of [BFPLC10, p. 3986–3999] was to describe a case study in home care scenarios, where an abstract architecture and a computational model for large scale open multi-agent systems based on a service-oriented approach were applied. The authors presented service examples for the management of a dependent home environment and demonstrated the new features of their proposal.

The success of ambient intelligence will depend on how secure it can be made, how the privacy and other rights of individuals can be protected, and how individuals can come to trust the intelligent world that surrounds them and in which they move. These issues were adressed in [FVPW07, p. 15–29] by analyzing scenarios for ambient intelligence applications that had been developed in the preceding years. The authors elaborated on the assumptions that promotors made about the likely use of these technologies and possible unwanted side effects, and concluded with a number of threats to personal privacy that had become evident.

A scenario-based query-processing system for video surveillance archives was proposed in [SGU09, p. 15]. In this system, a scenario was specified as a sequence of event predicates that can be enriched with object-based low-level features and directional predicates. The authors introduced an inverted tracking scheme, which effectively tracked moving objects and enabled view-based addressing of the scene. Their query-processing system also supported inverse querying and view-based querying, for after-the-fact activity analysis. A specific surveillance query language to express the supported query types in a scenario-based manner was proposed and a visual query-specification interface devised to facilitate the query-specification process.

This concludes the overview of the state of the art in scenario recognition. The scenario recognition methods used in this thesis based on the employed machine learning algorithms will be discussed in detail in the following chapters.

3 Behavior Model

Ambient assisted living (AAL) is a relatively new research field. It focuses on enhancing the quality of life of the elderly and prolonging their ability to live independently in their own homes with the help of modern technology. But with many elderly people suffering from conditions such as movement disorder, memory disorder, ... how can they use these modern technological systems? As discussed in Chapter 1, because of the abovementioned and other physiological and psychological problems, it is difficult for elderly people to learn and acquire new skills, such as how to use a computer, send an e-mail, and use new types of mobile phones. Therefore, products designed for the elderly should be as easy to use as possible, such as alarm buttons installed on the wall or wearable on the wrist, mobile phones with large keys and easy-to-read displays, or computers with touch screen technology. These products have appeared on the market and impart certain advantages to the elderly. But in some cases it is impossible for elderly persons to press an alarm button or use a phone to get help from outside, such as when they have taken a fall and cannot get up, or if they are ill in bed and lapsing into unconsciousness. In these sorts of situations a person may be in grave danger and should receive help as quickly as possible. But it is just in such dangerous situations that the user cannot do anything to activate the products designed to get needed help, because all the products mentioned above need to be activated by the user and a user who is unconscious or unable to move cannot activate anything. Only if products could be developed that do not require activation by the user, but instead are capable of autonomously determining when help is needed in abnormal or dangerous situations, this problem would be solved.

The idea behind this thesis is that "normal" behavior models of the user may be learnt through statistical methods, so that when any activity outside of the model occurs, that activity will be treated as a potentially abnormal or dangerous situation. In such cases an alarm signal will automatically be sent to a caregiver, and the user need do nothing in the entire process.

In this chapter, learning the behavior models of the user will be discussed. On the one hand, the activities of the user are dynamic and random. On the other hand, the living environment of the user influences his or her activities as well, for example: when the temperature in the living room becomes too high, the user will perhaps open the window. Here the user activity "opening the window" is influenced by the temperature within the living environment. Another example: the user is asleep in bed at night and suddenly the phone rings, so the user has to get up to answer the phone. Here the user activity "getting up" is motivated by the phone ringing. These two examples show that changes in the environment have an influence on the user. Such changes are difficult to model (a model for changing temperature is possible but it is difficult to include in the model if and when the user would open the window in correlation with rising temperatures in the living environment. The other example, the ringing telephone, is even more difficult to model). Since the changing environment directly influences user activity, and changes in the environment are difficult to model, it follows that those user activities caused by changes in the environment will also be difficult to model - the changing environment makes the activities of the user more complicated. Thus learning a useful behavior model of the user is an extremely challenging task. Furthermore, there is generally more than one room in the user's living environment, such as the living room, bath room, kitchen .... So the living environment and activity types change according to the changing location of the user. To learn the behavior model of the user under these circumstances will be even more complicated and challenging. In the end, the model should possess the ability to determine whether any given user activity is normal or abnormal, and should make this distinction by itself. Automatic decision making is therefore another challenging task in the thesis.

During the research, activities by the user were categorized into regular behavior (such as the user taking medicine tablets every day at nearly the same time, or the user receiving meals from a caregiver at nearly the same time daily) and random behavior (such as the user's daily routines).

For regular behavior the Gaussian mixture model (GMM) and the split-merge algorithm (SMA) were used (because the time distribution of the behaviors corresponds to the Gaussian mixture model, and the split-merge algorithm was used to determine the mean value, standard deviation value and mixture weight of the Gaussian mixture model) to learn the behavior model.

The hidden Markov model (HMM) was used to learn the behavior model for random behavior (such as the user's daily routines). The various activities constituting random behavior change from one day to another and the next activity only depends on the current activity, which corresponds to the definition of a Markov chain. Furthermore, these types of activity were not directly observable in the presented research project (without the use of cameras and microphones), but the observations (activity routine) from sensors (such as motion detectors and door contactors) were probability functions of these activities (states), which is characteristic of the HMM. For these reasons, the HMM was used to analyze the daily routine behavior. This chapter will detail the learning of the random behavior model of the user within a single room, while the behavior model of the user in the entire living environment (with different rooms) will be discussed in the following chapters. Furthermore, because user activities are dynamic and may appear completely random in a short timescale, there is no sense in trying to learn a behavior model in a short time interval, for example a few hours or one or two days. In such a short time interval the different types of activities of the user would likely not all occur and the model would thus be incomplete. For example, a behavior model of the user for the period of Monday to Friday could be learnt, but the user might engage in entirely different activities on weekends, so the model learnt from Monday to Friday would be incomplete for the user. Only over a longer time interval, for example at least one or more weeks, is it possible to learn a general and complete behavior model of the user.

3.1 Gaussian Mixture Model and Split-Merge Algorithm

Just as chapter 1 introduced that split-merge algorithm (SMA) was used to analyze the parameters of Gaussian mixture model (GMM). The Gaussian mixture model (GMM) allows different mean and deviation values within a Gaussian function. The split-merge algorithm (SMA) can be used to analyze the mean and deviation values. With the dynamic variables the mean and deviation values are changing. Split-merge algorithm (SMA) handles these kinds of dynamic variables and produces the mean and deviation values. That means the mean and deviation values dived from a split-merge algorithm (SMA) directly influenced by the dynamic variables.

There are many papers about Gaussian mixture model (GMM) and split-merge algorithm (SMA): for speaker identification [CWF04, p. 2626–2639] proposes a self-splitting Gaussian mixture learning algorithm for Gaussian mixture modeling, [UNGH00, p. 2109–2128] presents a split and merge expectation-maximization (EM) algorithm to overcome the local maximum problem in Gaussian mixture density estimation. In [RG97, p. 731–792] the authors have developed a new methodology for fully Bayesian mixture analysis, using reversible jump Markov chain Monte Carlo methods for jumping the parameter subspaces and the different numbers of components in the mixture, while [MH05, p. 203–210] propose a new kind of dynamic merge-or-split learning algorithm to deal with the selection of number of Gaussians in the mixture, [Ait99, p. 117-128] describe an expectation-maximization (EM) algorithm for nonparametric maximum likelihood estimation with variance component structure, [VL02, p. 77–87] introduces a greedy algorithm for learning Gaussian mixture model (GMM), using combination of global and local search, [ZCSC03, p. 1973–1983] introduces a split-merge operation in order to alleviate the problem of local convergence of the usual expectation-maximization (EM) algorithm, in paper [YB09b, p. 4155–4158] the authors use split-merge algorithm (SMA) to analyze the data from the tracked video data.

In this thesis Gaussian mixture model (GMM) and split-merge algorithm (SMA) are used to analyze the sensor data, for example the data from medicine box and the data from meal entry contactor. The general daily models will be learnt around the sensor data. Based on the model, if some unusual behaviors happen, the system will send a warning signal to the user themselves or an alarm signal to a neighbor or caregiver.

For example if a user takes tablets from a medicine box at similar time points every day. A contact sensor is installed at the door of the medical box. If the door is opened or closed a signal will be sent to the controller. Based on the gathered data for some days, for example one month, the system learns the model that when the medical box is opened and closed, that is when the user is taking their tablets from the medical box. If one day the user forgets to take tablets at some time points, the system will send a warning signal to the user. The same situation applies for the contact sensor installed at the meal entry. Every day the meal will be sent to the user through a meal entry. A contact sensor gets data every time when the meal is sent into the room or the tableware is sent out of the entry. A model will be learnt based on the gathered data for a time interval, for example one month. If one day no meal is sent to the user at the expected time points, the system will send a warning to the user, neighbor or caregiver.

A contact sensor is installed in the medicine box, so that if the door of the medicine box is opened or closed then different sensor values would be sent to the controller. Here the important point was when the medicine box was opened (or closed) by the user, but not how long the medicine box was open (generally a few seconds to a minute). All the time points t were gathered over one month and were analyzed. There should be set time points $T = (t_1, t_2, t_3 \dots t_n)$. Because the user takes tablets every day several times at some time points, for example in the morning, before lunch or after lunch, and in the evening just before going to bed, there should be some time points distribution. The distribution is corresponded to Gaussian mixture model (GMM). The same situation happened with the sensor data from the meal entrance.

Through analyzing the time points gathered about one month (just as above discussed, too short time interval will no great sense to learn a model, on the other side if the activity of the user is stable, so the learnt model will be stable. A few weeks is the shortest time interval for learning a useful model and with longer time interval the model will not changed much. This point will be discussed with example in the following), a generally model of the sensor data will be learnt, which means, for the medical box sensor is when the user will take tablets, and for the meal entrance contact sensor is when the meal will be send to the user or the tableware send out of the entry. Here the gathered time points composed clusters. In fact this is cluster analysis problem. Gaussian mixture models and split-merge algorithm (SMA) will be used to deal with the analysis of the clustered data.

3.1.1 Gaussian Mixture Model

A standard Gaussian function is defined as

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.1}$$

Here μ is the expected value (mean value of the clustering time points), σ is the standard deviation of each time point cluster, x is the time point value (t_n). If there are different mean and standard deviation values in the Gaussian function, so the Gaussian function is multivariate Gaussians and could be called Gaussian mixture model (GMM). The sprit-merge algorithm (SMA) could be used to analyze the GMM and find out the mean values, standard deviation values, and the mixture weight of each component (cluster) in Gaussian mixture model.

3.1.2 Split-Merge Algorithm

Before the split-merge algorithm (SMA) was used for clustering the time points set, the range of the parameters have to be defined: $0 < \mu \leq 24$ (because there are only 24 hours one day, so the time point interval is between 0 and 24), $0.5 \leq \sigma \leq 2$ (the initial value can be changed according the real situation) , $0 \leq t_n \leq 24$. The number of initial components $s = 3$, that means there are sets $\{\mu_1, \mu_2, \mu_3; \sigma_1, \sigma_2, \sigma_3; P_1, P_2, P_3\}$. Here P is the percent value of each time point component and $\Sigma P_{(1,2,3)} = 1$ and these prior parameters are random variables. Threshold value for split, merge and delete components: $\mu_{threshold}$, $\sigma_{threshod}$, $\sigma_{threshod2}$, $P_{threshold}$. The maximum number of time points for adjusting the learning rate is M and current count is M'. Each new time point that gets into the set T is T_r $(r \geq 1)$. The index s is the component index within the mixture model. For the same parameters of the equations on the right side of the equations are prior symbols (prior here means which probability value the component can be obtained) and on the left side are the posterior symbols (posterior here means how the components are assigned). With these parameters and definition the data set can be analyzed with split-merge algorithm [YB10c, p. 4].

Compute and then normalize posteriors

$$P_s(T_r) = P_s \times \varphi_{\mu,\sigma^2}(T_r); P_s(T_r) := \frac{P_s(T_r)}{\Sigma P_s(T_r)} \tag{3.2}$$

Compute new means

$$\mu_s := (1 - P_s(T_r)) \times \mu_s + P_s(T_r) \times \frac{M' \times \mu_s + T_r}{M' + 1} \tag{3.3}$$

Compute new variances

$$\sigma_s := (1 - P_s(T_r)) \times \sigma_s + P_s(T_r) \times \frac{M' \times \sigma_s + |\mu_s - T_r|}{M' + 1} \tag{3.4}$$

Compute new priors

$$P_s := \frac{M' \times P_s + P_s(T_r)}{M' + 1} \tag{3.5}$$

Keep the learning rate and adaptability if

$$M' \geq M, M' = M \tag{3.6}$$

After some initial iterations, start checking if it is necessary to split components: If $\sigma_s > \sigma_{threshold}$, then create new component (index S) from old component (index s)

$$\mu_S = \mu_s + \frac{\sigma_s}{2}; \mu_s := \mu_s - \frac{\sigma_s}{2} \tag{3.7}$$

$$\sigma_S = \frac{\sigma_s}{2}; \sigma_s := \frac{\sigma_s}{2} \tag{3.8}$$

$$P_S = P_s; P_s := \frac{P_s}{2} \tag{3.9}$$

If necessary, merge components s' and s''

If $|\mu_{s'} - \mu_{s''}| < \mu_{threshold}$ and $|\sigma_{s'} - \sigma_{s''}| < \sigma_{threshold2}$ then merge component s'' into s' and delete component s''. Here $\sigma_{threshold2} \leq \sigma_{threshold}$.

$$\mu_{s'} := \frac{\mu_{s'} \times P_{s'} + \mu_{s''} \times P_{s''}}{P_{s'} + P_{s''}} \tag{3.10}$$

$$\sigma_{s'} = max(\sigma_{s'}, \sigma_{s''}) \tag{3.11}$$

$$P_{s'} := P_{s'} + P_{s''} \tag{3.12}$$

If any component's prior decreased too much so that $P(s') < P_{threshold}$ then delete the component and adjust the other priors $P(s)$:

$$P(s) := \frac{P(s)}{\Sigma P(s)} \tag{3.13}$$

Repeat with all new values.

3.1.3 Learning Behavior Model with Split-Merge Algorithm

Based on the above introduction in the following the data from the regular behavior (for example the user takes medicine tablets several times every day or the user gets meal from caregiver) of the user should be analyzed with split-merge algorithm and furthermore the behavior model would be learnt.

The first example is about the original sensor data from medicine box about one month (just as above introduced that there must be a relatively longer time period in order to get enough data to learn the user behavior model, such as one month or more). Every time when the user opens or closes the door of the medicine box, the door contactor installed at the medicine box will send signal to controller, the signal indicates when the user opens or closes the door of the medicine box. We can treat these time points as the time points that the user takes tablets.

Figure 3.1 is the histogram of data (time points) from the medical box. X-axis is the daily time from 0 to 24:00 (30 minutes as time interval in each hours, that means each blue bar has time duration 30 minutes, in the 30 minutes if the user opens or closes the medicine box 6 times, the count value of blue bar will be showed in Y-axis with value 6), Y-axis is the count about the activities of the user (how many times the user opens or closes the door of the medicine box in each time interval). From the figure we can see there are different clusters (blue bars gathered together) at different time, about at 7, 10, 12, 15, and 21 o'clock. That means the user takes tablet daily about the time points. But just from the figure the parameter about each cluster, for example mean value and deviation value, cannot be indicated directly. With the split-merge algorithm (SMA) the mean value and deviation value of each cluster can be found out automatically.

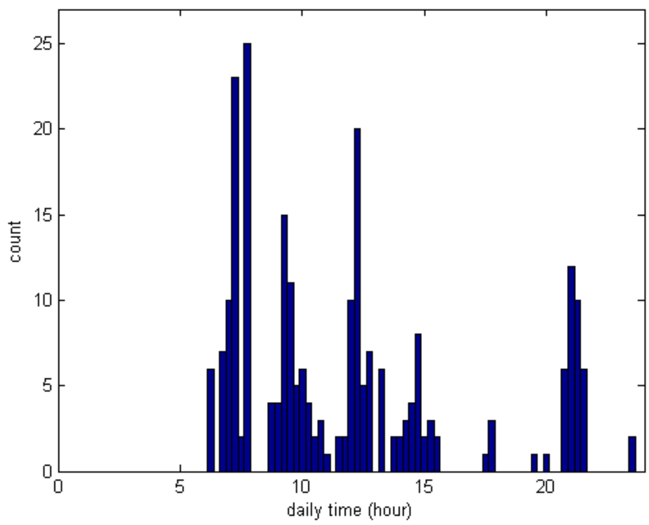


Figure 3.1: The histogram of data from the medicine box in one month

Figure 3.2 showed the learnt result (the green bell curve) with the histogram (the same as figure 3.1) together. The green bell curves here are the learning result, each green bell indicated a Gaussian function with different mean and deviation value. From the green bell curves in figure 3.2 we can see that all the important clusters (different groups of cylinders) were detected through learning. The mean values, standard deviation values, and mixture weight values of these cylinders were found out.

There are mainly 5 clusters (indicated with 5 green bell curves, these curves are the learnt regular behavior model of the user. The model indicates when the user takes medicine tablets with what kind of time deviation) in figure 3.2. X-axis is the daily time from 0 to 24:00; Y-axis is the count about the activities of the use and at the same time Y-axis indicates the probability density of the green bell curves. Each cluster means the time that the user takes tablets: in the morning about 7.3, 10.4, at noon 12.4, 14.5, and in the evening 21.2. The mixture weights are about 30%, 20%, 20%, 20%, and 10%. That means, for example the cluster in the morning about 7.3 has the mixture weight 30% of the whole clusters. Because the data gathered from one month, the user has not taken tablets every day at the exactly same time points, so there are standard deviations for each cluster: 0.4, 0.6, 0.4, 0.4, and 0.5. The standard deviation values came from the learning result of split-merge algorithm, they indicated the user taken medicine tablets did not always at exactly the same time point but a little earlier or a little later about the time mean values.

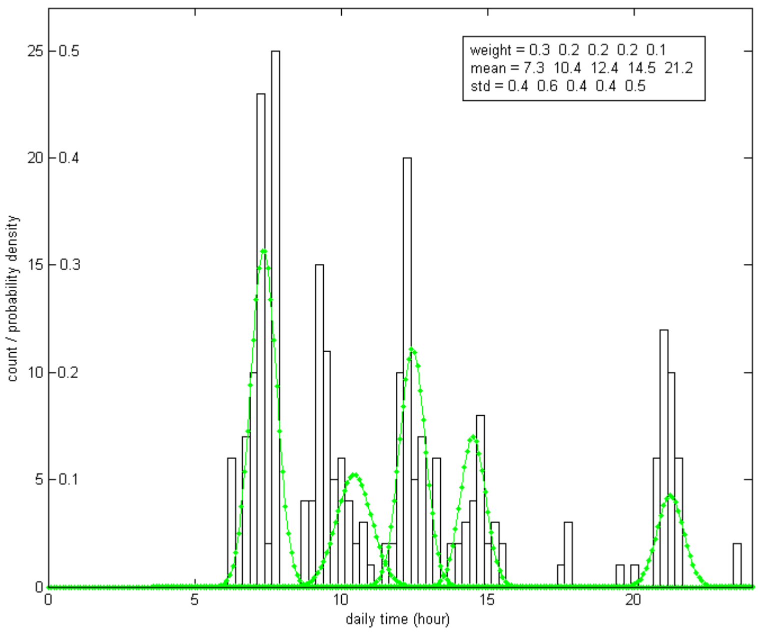


Figure 3.2: To compare the learn result with histogram in one month

Figure 3.3 is the histogram of data with a time period about 2 months from the medical box. From the figure we can see that the data gathered about to 7, 10, 12, 15, and 21 o'clock. With the learning algorithm the mean time point of each cluster can be found out automatically.

Figure 3.4 is the learning result with histogram (data gathered from about 2 months). X-axis is the daily time from 0 to 24:00; Y-axis is the count about the activities of the use and at the same time Y-axis indicates the probability density of the green bell curves. From the Figure 3.4 we can see that there are mainly 5 time clusters that the user takes tablets: in the morning about 7.5, 9.3, at noon 12.2, 14.1, and at evening 20.1. The mixture weights are about 20%, 20%, 30%, 10%, and 20%. The standard deviations for each time points are: 0.5, 0.4, 0.3, 0.5, and 0.5. The standard deviation values came from the learning result of split-merge algorithm. The data

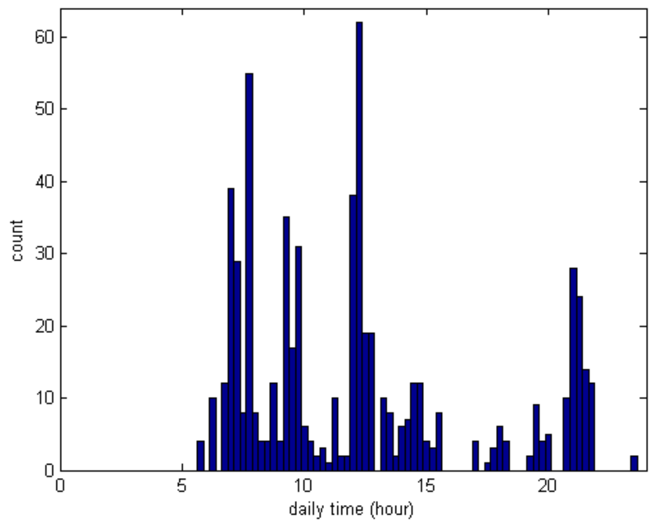


Figure 3.3: The histogram of data from the medicine box about two months

gathered about 2 months, the user has not taken tablets every day at the exactly same time points (a little earlier or a little later about the mean values). The standard deviation values indicated the time deviation that the user taken medicine tablets. The big deviation around 20.1 in the figure is because that the learning result (indicated with green bell curve) was influenced by the data (the time points that earlier than 20.1), so the mean value (in the middle of the green bell curve) looks like deviated to left side. In fact the split-merge algorithm treated these time points (around 20.1) as one cluster, the mean value came from all the data near 20.1. Furthermore because these time points (around 20.1) did not focus on together, so the learning result of the cluster has a big deviation value 0.5.

Another important point from figure 3.4 is that, if the user shows a particular behavior regularly at some time points the standard deviations will be smaller. For example at the third cluster the time mean value is 12.2 and the standard deviation is 0.3. The value is smaller than other standard deviation values. This is because the user takes medicine tablets at noon very regularly, did not have much time deviation. This indicates that regular behavior related with a more precise learning result. On the other hand, in the evening the user takes tablets with less focused time points, so there the standard deviation value is 0.5.

Furthermore, on analyzing the learning result, the conclusion should be: if the user has regular daily behaviors, the learning result will be more precise, on the other hand if the user has a less regular more lifestyle, the learning result will be more inaccurate. This will lead in extreme situations to an incorrect learning result. For example a user takes tablets on Monday morning about 7:30, on Tuesday morning 7:31, on Wednesday Morning 7:29, and nearly every morning about 7:30 the user takes tablets. So the learning result will be like this: the mean value is 7:30 and the standard deviation is 1 or 2 minutes. That means the user takes tablets on the morning about 7:30, perhaps 1 or 2 minutes earlier or later. On the other side if a user takes tablets on Monday morning about 7:30 but on Tuesday morning 9:30 and on Wednesday morning 4:00, and every morning at different time. So the learning result should be like this: the mean value is 7:00 and the standard deviation is 2 or 3 hours. From the different standard deviation value of the two different users we can see that the learning result of the first user is very precise, it has only

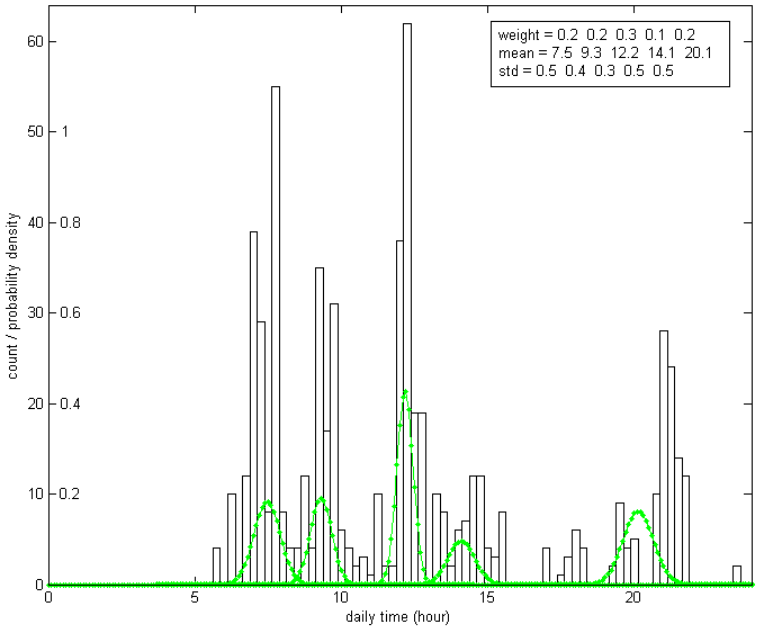


Figure 3.4: To compare the learn result with histogram about two months

deviation 1 or 2 minutes. We can say that the user always takes their tablet(s) in the morning about 7:30, perhaps 1 or 2 minutes earlier or later. But the second user has a deviation of 2 or 3 hours, so we can only say that the user takes tablets in the morning about 7:00, perhaps 2 or 3 hours earlier or later, very inaccurate. In fact the user himself is the "trainer". If the user has stable lifestyle the learning result will be precise but if the user has very irregularly lifestyle the learning result will be very inaccurate.

Fortunately some of the elderly have a relatively stable lifestyles, for example: when they get up, when they take a shower, when they take breakfast, when they have lunch, when they have a rest. . . when they go to bed. A regularity of lifestyle is a basic necessity for a useful learning result. If the stable lifestyle changes, it is probable that something unusual has happened or is about to happen. Using this idea the hidden health problems of the user can be predicted earlier. If we compare the learning result of figure 3.2 and figure 3.4, it is clearly that the user take tablets in a relatively stable time points.

The second example is from the meal entrance contactor. Every time when the caregiver sends meal to the user, the door contactor installed at the meal entrance will send a signal to the controller. The signal indicates when the user gets a meal every day. We can treat these time points as the time points that the user gets their meals.

Figure 3.5 is the learning result from the data set about one month. There are mainly 3 clusters (indicated with green bell curves, these curves are the learnt regular behavior model of the user. The model indicates when the user gets meal with what kind of time deviation). X-axis is the daily time from 0 to 24:00; Y-axis is the count about the activities that the caregiver sends meal to the user (the user gets meal) and at the same time Y-axis indicates the probability density of the green bell curves. The mean values of these clusters are: in the morning 6.4, 7.7, and at noon

12.2. The mixture weights of these clusters are 20%, 20%, and 60%. The standard deviations for each cluster are 0.6, 0.4, and 0.5.

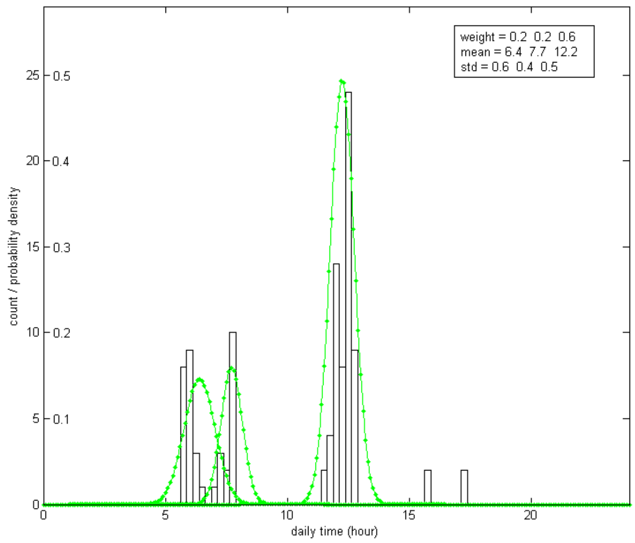


Figure 3.5: To compare the learn result with histogram (data from entrance contactor) in one month

For comparing the learning result to the figure 3.5 the learning algorithm with more date should be run again. Figure 3.6 is the learning result from a data set taken over nearly 2 months. There are 3 predominant clusters (indicated with green bell curves, these curves are the learnt regular behavior model of the user. The model indicates when the user gets meal with what kind of time deviation). X-axis is the daily time from 0 to 24:00; Y-axis is the count about the activities that the caregiver sends meal to the user (the user gets meal) and at the same time Y-axis indicates the probability density of the green bell curves. The mean values of these clusters are: in the morning 6.4, 8, and near noon at 12.3. The mixture weights of these clusters are 20%, 10%, and 70%. The standard deviations for each cluster are 0.5, 0.4, and 0.5.

It has the similarly learning result to the figure 3.5. But if we analyze the learning result clearly, the learning time points changed a little. For example the breakfast times from 6.4 and 7.7 change to 6.4 and 8. The lunch time from 12.2 changes to 12.3. The changing comes from the normal lifestyle of the user and the learning algorithm. The results are in the standard deviation scope. For each learning result shows in figure 3.6: $6.4-0.6 \leq 6.4 \leq 6.4+0.6$, $7.7-0.4 \leq 8 \leq 7.7+0.4$, $12.2-0.5 \leq 12.3 \leq 12.2+0.5$.

That means the learning result in 2 months is in the learning result scope from one month. It indicates that the user has a relatively stable lifestyle and the algorithm is reliable.

On the contrary if we use the learning result (mean and standard deviation value) from about 2 months to test the learning result from one month, the mean values (6.4, 7.7 and 12.2) are in the scope too: $6.4-0.5 \leq 6.4 \leq 6.4+0.5$, $8-0.4 \leq 7.7 \leq 8+0.4$, $12.3-0.5 \leq 12.2 \leq 12.3+0.5$.

The above learning results indicated that the elderly person has stable lifestyle. The user has a meal 2 times (or one time the caregiver takes the tableware from the day before and another time the caregiver sends the meal to the elderly person) in the morning and once at noon, but does not take dinner in the evening. If the lifestyle changed or no meal is sent to the user at the time

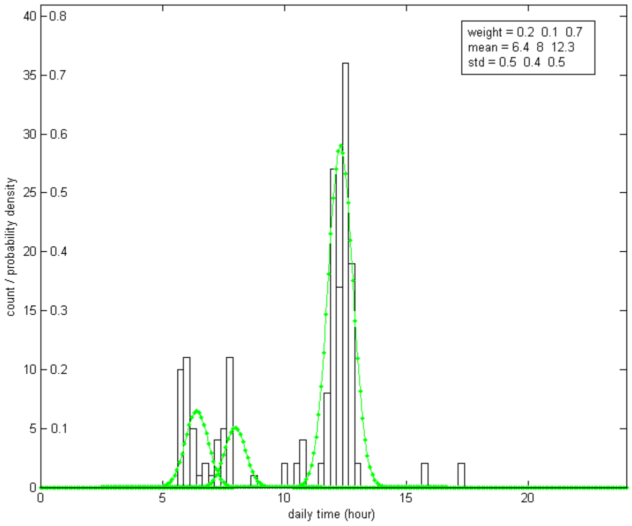


Figure 3.6: To compare the learn result with histogram (data from entrance contactor) about two months

points (within the standard deviation interval), the system should send a signal to neighbor or caregiver.

The above examples indicated that as unsupervised learning approaches the GMM and the SMA could be utilized in practical terms. The Gaussian mixture model (GMM) and the split-merge algorithm (SMA) have advantages in comparison to the other cluster learning approaches, such as by k-means clustering and fuzzy clustering . By k-means clustering the learning result is strongly related the initial definitions (the initial number of clusters) and by fuzzy clustering the central point of a cluster is the mean of all points, that means all the prior data points have to be computed each time. This is a computational problem with a huge amount of data.

Furthermore the above examples indicated that the Gaussian mixture models (GMM) and the split-merge algorithm (SMA) were suitable for analyzing the regular behavior (takes medicine tablets or gets meal from care giver at nearly same time points daily and these behaviors could be recognized by data from single sensor) of the user. But for the random behavior (such as daily activity routine) the above algorithms were difficult to utilize. This was because the behaviors of the user in the daily activity routine were difficult to recognize based on single sensor. It was difficult to judge if same behavior happened. For the daily random behavior it had to be found another way to deal with.

3.2 Hidden Markov Model

Above the Gaussian mixture model and the split-merge algorithm were used to learn the behavior model with regular behavior of the user. In the following the random behavior of the user would be analyzed and the behavior model of random behavior of the user would be learnt. Just as above introduced that some of the elderly people have stable lifestyle, such as sleep in bed, get up in the morning, take a shower, have breakfast, do some homework, have lunch, have a rest, cook for dinner, have dinner, watch TV, and go to bed in the evening. Each of the different behavior could be treated as different state. The different behavior changed from one to other and the state changed from one to another. The behavior for example taking a shower was only related the behavior getting up in the morning, did not related the earlier behavior for example sleeping in the bed. That means the next state (taking a shower) only depends on the current state (getting up in the morning) but did not related with the earlier state (sleeping in the bed). Markov chain ([MT93, p. 558], [Rab89, p. 257–286]) described the kind of characters between different states. That means the daily routine of the user (state changed from one to another and the next state only depends on the current state but did not related with the earlier state) is a Markov chain. Furthermore because of in the thesis no camera and microphone were used as sensors (privacy reasons), so the behavior (such as get up in the morning, take a shower) of the user can not be observed directly. That means the state cannot be observed directly (state is hidden). But the observation (activity routine which came from sensor data, such as from motion detector, door contactor) is a probability function of the state. In such situation (state is hidden and the observation is a probability function of the state) the Markov chain is called hidden Markov model. In this section a hidden Markov model (HMM) will be used to analyze the motion detector sensor data.

There are many papers about Markov chain (MC) and hidden Markov models (HMM): [MT93, p. 558] introduced Markov chain (MC) and stochastic stability in detail. In paper [SO93, p. 8] the authors describe a technique to facilitate the learning of the number of states and the topology of a hidden Markov model (HMM) from examples. Here the Bayesian posterior probability was used for choosing the states to merge and for the stopping criterion. The author of paper [Bil02, p. 32] describes the expectation-maximization (EM) algorithm and parameter estimation for hidden Markov models (HMM). The hidden Markov model (HMM) and the related structures of probabilistic independence networks searched in [SHJ96, p. 37]. In paper [GJH01, p. 398-413] variable-length Markov models were used to achieve efficient representation of behaviors. The Markov models was used as Bayesian networks for the tasks of transmembrane protein topology prediction and signal peptide prediction described in paper [RKR+08, p. 14]. The papers [BSR06, p. 132–137] and [BSL07, p. 8] adopt the hidden Markov model (HMM) to analyze the motion detector data, and thereby to learn the behavior of the user. In paper [BSR06, p. 132–137] the authors take advantage of semantic symbols, to learn probability models in building automation systems (BAS).

3.2.1 Daily Activity Learning from Motion Detector Data

In this thesis a hidden Markov model (HMM) and forward algorithm were used to analyze the sensor data, for example the data from a motion detector installed at a corner of the living room. The general daily activities models about the sensors will be learnt and, based on the model, if some unusual activities and behaviors happen, the system will send a warning signal to the user or an alarm signal to a neighbor or caregiver.

For example, say a user has activities in the morning, at noon and in the evening at the living room. A motion detector is installed at a corner of the living room, and it records the daily activities of the user. Using a hidden Markov model (HMM) and a forward algorithm the activities model of the user may be learnt, because the elderly person has relatively stable lifestyle. But on the other hand there are always differences in daily living patterns. For example at the weekend the user may get up a little later, or spend a little longer on cleaning work. They may also have visitors in the living room. So the activities of the user should be generally keeping a similar style but may not be exactly the same. That means the learnt activities model should keep the similar style but can accommodate different routines. Furthermore if any abnormal activities happen for example: in the morning in the living room there is no user activity, and this kind of situation never happened before, based on the learnt model the system should send a warning or alarm signal to the user, a neighbor, or a caregiver.

3.2.1.1 Basic Parameters

In the living room where a motion detector is installed, if there are activities from the person, the motion detector sends the value "1" to the controller. If there is no activity, the motion detector sends the value "0" to the controller. If there are continuous activities the values will keep sending "1" till the activities halt and the motion detector sends values "0" to the controller. If there is continuous stillness the values will stay at "0" till activity starts again and the motion detector sends values "1" to the controller. This is the basic function of the motion detector.

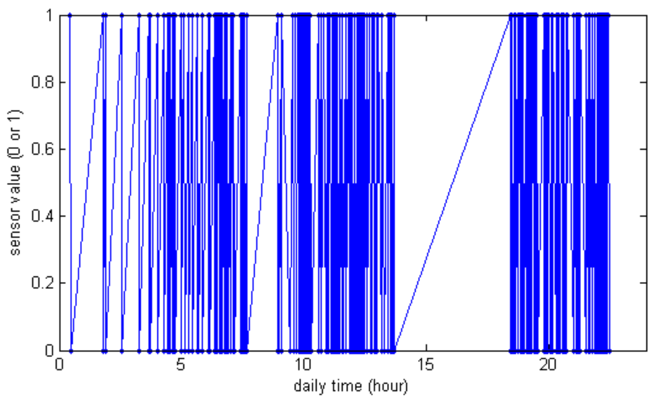


Figure 3.7: The data from one motion sensor in one day

Figure 3.7 shows the gathered sensor data values from one day, there are 407 values all. The x-axis is daily time from 0 to 24 hours. The y-axis is the sensor value. There are only 2 different values 0 or 1. Figure 3.7 indicated that generally in the morning (between 5 and 8), at noon (between 9 and 14) and at night (around 19 and 23) there are many activities, but between 0 to 5 hours, between 8 to 9, between 14 to 19, and after 23 hours it is really quiet.

The questions about these kind of parameters are: first, there are too many parameters in each day; second, each value has the same importance but on the other hand each individual value doesn't have great meaning. For example there is little sensor data (activities) between 0 and 5 in figure 3.7. It perhaps comes from the user rolling their body while asleep at night. If we think a step deeper: a few activities (for example rolling the body in bed or getting up to go to the WC) in the sleeping period is normal but if there are too many activities - perhaps it means the elderly person has sleep disorders or other hidden physical health problems.

How to decide "a few activities" and "many activities" with this kind of sensor data? An easy and useful idea is to gather the activities in a time interval. If the sum value of activities is bigger than a predefined threshold value, so this time interval will be treated as activities value "1", otherwise value "0" is assigned to the time interval. Here the predefined threshold value T_{th} and the time interval $T_{interval}$ play an important role.

The time point of the gathered sensor value (for example in 30 minutes)

$$T = \{t_{1(1)}, t_{2(0)}, t_{3(1)}, t_{4(0)}, t_{5(1)}, t_{6(0)} \ldots t_{n(v)}\} \tag{3.14}$$

Here t_n is the time point that the motion detector sends a value to the controller ($n \geq 1$). Small v is the sensor value itself, it has value "0" or "1".

The time duration between sensor values ΔT

$$\Delta T = (t_{n(0)} - t_{n-1(1)}) \tag{3.15}$$

The sum of the time duration in the time interval T'

$$T' = \Sigma(\Delta T) \tag{3.16}$$

Decide if the time interval gets value "1" or "0"

if

$$T' \geq T_{th} \times T_{interval}, Q_{interval,ix} = 1 \tag{3.17}$$

if

$$T' < T_{th} \times T_{interval}, Q_{interval,ix} = 0 \tag{3.18}$$

Here T_{th} is the threshold value ($0 < T_{th} < 1$, because $T' \geq 0$, so $T_{th} > 0$, because T' cannot bigger than $T_{interval}$, so $T_{th} < 1$). $T_{interval}$ is the predefined time interval, for example 30 minutes. $Q_{interval,ix}$ is the result value that the interval should take. Here "ix" is the interval count (index).

Predefining the $T_{interval}$ as 30 minutes, so there should be 48 intervals (that means $1 \leq n \leq 48$) one day. If the $T_{interval}$ chosen has a smaller value the accuracy will be increased but the computational load will be increased too. For example if the $T_{interval}$ was chosen as 1 minute there will be 1440 states ($60 \times 24 = 1440$ states) each day. On the other hand if the $T_{interval}$ was chosen with a bigger value, for example 120 minutes, there should be just 12 intervals each day, but the observation accuracy of the user activity will be reduced significant. For example the user took a shower in bathroom for 10 minutes (with high activity), had breakfast for 20 minutes in living room (with high activity), and then he read newspaper for 30 minutes (with low activity), at last he had a rest for 40 minutes (with low activity). If the $T_{interval}$ was chosen as 120 minutes it will be difficult to distinguish different activity in the long time interval. It is difficult to decide which $T_{interval}$ value is better. The point will be discussed in the following chapter.

The same situation happens with the threshold value T_{th}. If the T_{th} was chosen too small, some "noise" sensor values (activities like for example the user rolling their body while sleeping and the activities were detected by motion detector which installed in bedroom) will be translated to interval value $Q_{interval,ix} = 1$; if the T_{th} be chosen too big, some activities will be depressed. For example the user went to WC at night. The activity of the user was perhaps only 2 or 3 minutes. If the T_{th} was chosen bigger than 0.1, so the activity would be ignored. Furthermore if the user had dangerous situation during the short time interval, for example fell down in the WC, the dangerous situation would be not detected. How to find an optimal threshold value T_{th} will be introduced and discussed in the following chapter.

In figure 3.8 30 minutes be chosen as $T_{interval}$ and 0.2 as T_{th}. It means at least 6 minutes in the time interval the user has activities that Q can take value 1. If T_{th} was chosen bigger there must be more activities then Q can take value 1. Here 6 minute is just an example.

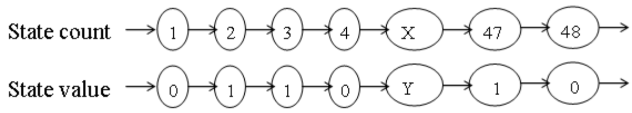


Figure 3.8: Filtered sensor value

The top of figure 3.8 is the state count and the bottom is the filtered sensor state value. The state value "1" means in this state there are activities from the user, others value "0" means in this state there are no activities from the user. Symbol X means state count and Y means state value. From figure 3.8 we can see that there are just 48 state values after the transfer algorithm.

Till now the data account be reduced, the activities of each time interval was translated to state value, but this is just a generally intuition of the activities of the elderly person. Using a hidden Markov model (HMM) and forward algorithm we can get an activities model of the elderly person. If 30 minutes be chosen as $T_{interval}$, so there should be 48 activities value one day, for example $Q_{interval,ix} = \{Q_{interval,1} = 0, Q_{interval,2} = 0, Q_{interval,3} = 1, \ldots, Q_{interval,m} = 0, \ldots, Q_{interval,46} = 1, Q_{interval,47} = 0, Q_{interval,48} = 0\}$. Here $1 \leq ix \leq 48$. If each interval activity value (0 or 1) was treated as "state", so there should be 48 states each day.

3.2.1.2 Markov Chain

The papers [MT93, p. 558], [Rab89, p. 257–286] give an introduction about Markov chain (MC): sequences ϕ_n evolving randomly in time which remember their past trajectory only through its most recent value. Here the sequences were composed with different states. The states are observable (physical) event. But if the states are cannot be observed directly (they are hidden) and the observations are probability functions of the states, so the Markov chain changed to hidden Markov model. In this thesis, because of no cameras and microphones were used, so the behavior of the user could not be directly observed. With the sensors (such as motion detector, door contactor) the behavior of the user could be indirectly observed. Furthermore the observation (activity routine) was a probabilistic function of the behavior (the hidden state), which is in fact the definition of a hidden Markov model.

3.2.1.3 Hidden Markov Model

In Markov model each state corresponded to an observable (physical) event. But when the observation is a probabilistic function of the state, this is called a hidden Markov model (HMM): a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observation [Rab89, p. 257–286].

Connecting the theory and practice (hidden Markov models definition and the real application), at first a hidden Markov model should be learnt based upon the daily activities of the user and then using the model to explain the observed sequence of daily activities. The first question is how to learn a hidden Markov model. It needs some basic definitions and algorithm.

3.2.1.4 Basic Parameters of Hidden Markov Models

There are some parameters characterize hidden Markov models (HMM): the number of states N, the number of output distinct observation symbols each state M, the state transition probability distribution matrix $A = \{a_{ij}\}$, the state emission probability distribution matrix $B = \{b_{ik}\}$, and the initial state distribution $\pi = \{\pi_i\}$. For a better understanding, in the following the real application will be consulted to explain these parameters.

The number of states N

Above introduced that if $T_{interval}$ chosen as 30 minutes, so there should be 48 activity values per day. If each activity is treated as a state there should be 48 states, but because of merging of different states the states count will be reduced. There are two different situations in which the states can be merged together.

The first situation is the merging of identical states. For example in some time intervals in one day the activity value of the user stays at “1”, $Q = \{\ldots 1, 1, 1, 1, 1, \ldots\}$, because the state transition probability value between state t and state t+1 remains at 100% and the state emission probability value keep the same, so these states can be merged into one state. In the merged state there are 2 parameters: $P(Q_{ii})$ and $P(Q_{ij})$.

$$P(Q_{ii}) = \frac{N}{(N+1)} \tag{3.19}$$

$$P(Q_{ij}) = \frac{1}{(N+1)} \tag{3.20}$$

Here $P(Q_{ii})$ is the “self-transition” probability, $P(Q_{ij})$ is the transition probability, and N is the number that is merged states. $P(Q_{ii}) + P(Q_{ij}) = 1$. In the situation $b_{ik} = 1$.

The second situation is the merging of consecutive states when the state values alternate in a regular and predictable fashion. For example, during some time intervals on one day the activity value of the user is $Q = \{1, 0, 1, 0, 1, 0 \ldots\}$, the values “1” and “0” appear alternately. It has the 100% state transition probability value between state t and state t+1 and the emission transition probability with increased states count closer to 0.5. All these states could merge into one state. The parameters $P(Q_{ii})$ and $P(Q_{ij})$ computed as above but the emission transition probability is different.

$$b_{i0} = \frac{N_0}{N} \tag{3.21}$$

$$b_{i1} = \frac{N_1}{N} \tag{3.22}$$

Here N_0 is the count that all the states have values “0” and N_1 is the count that all the states have values “1”. It is clearly $N_0 + N_1 = N$.

The number of output distinct observation symbols each state M

Here are only 2 distinct observation symbols “0” and “1”. So $M = 2$.

The state transition probability distribution matrix $A = \{a_{ij}\}$

$$a_{ij} = p(Q_{t+1} = j|Q_t = i) \tag{3.23}$$

$0 \leq a_{ij} \leq 1$ and $\sum_{j=1}^{N} a_{ij} = 1$, $1 \leq i$, $j \leq N$. Here Q_t is the current state at time t.

The transition probability distribution in the merging situation has been discussed above. Another situation is the split situation: from time interval t to the next time interval $t+1$ there is more than one state connected with the same state Q_t. For example in state Q_t there are 10 values, all these values are "1". In the next time interval $t+1$ there has a state that has 4 values and all the 4 values are "1" and another state has 6 values and all the 6 values are "0". So the state transition probabilities are 0.4 and 0.6 separately.

The state emission probability distribution matrix $B = \{b_{ik}\}$

$$b_{ik} = p(O_t = k|Q_t = i) \tag{3.24}$$

$0 \leq b_{ik} \leq 1$ and $\sum_{i=1}^{M} b_{ik} = 1$, $1 \leq i \leq N$, $0 \leq k \leq M$. Here O_t is the output symbol at time t.

The initial state distribution $\pi = \{\pi_i\}$

$$\pi_i = p(Q_o = i) \tag{3.25}$$

$\sum \pi_i = 1$, for example there are 2 initial states π_1 andπ_2, π_1 include 3 values, all are "0" and π_2 include 7 values, all are "1", so $\pi = \{\pi_1, \pi_2\} = \{0.3, 0.7\}$.

3.2.1.5 The Forward Algorithm

Given a hidden Markov model that means the parameter (π, A, B) are known, how we can find the probability of an observed sequence $O^{(t)} = \{q_1, q_2, \ldots, q_t\}$? Here each of the q is one of the observable set. The forward algorithm is used.

Get the first transition probability α_1.

$$\alpha_1(j) = \pi(j) \times b_j(O_1) \tag{3.26}$$

Here j is the observation count of each observation set.

For $1 < t < T$ (the length of the observation seauence O) get the transition probability $\alpha_t(j)$

$$\alpha_t(j) = b_j(O_t) \times \sum_{i=1}^{N} (\alpha_{t-1}(i) \times a_{ij}) \tag{3.27}$$

For $t = T$ get the termination.

$$P(O/\lambda) = \sum_{i=1}^{N} (\alpha_T(i)) \tag{3.28}$$

Here λ is the learnt hidden Markov model.

3.2.2 Learning User Daily Behavior Model

Some examples about behavior model and structure learning: The authors of paper [KD05, p. 441–448] develop an algorithm for learning the structure of Markov logic networks from relational databases, combining ideas from inductive logic programming and feature induction in Markov networks. In paper [BMW10, p. 157–165] the authors review the evidence for structure learning as a "learning to learn" mechanism and utilize it to sensorimotor control. The authors of paper [BMGR10, p. 5470–5475] proposed implicit score to learn Bayesian network structure from database. The authors of paper [Hsu04, p. 103–122] design a generic fitness function for validation of input specification and use it to develop genetic algorithm wrappers. One of the wrappers used to deal with the variable ordering problem for Bayesian network structure learning. In paper [BSR06, p. 132–137] and [BSL07, p. 8] the authors introduced states merging method that may be used in security, care system, and building automation system (BAS).

3.2.2.1 Split and Merge of States Routines

To learn states routine from different days into one behavior model, the states split and merge algorithms are needed. In the following the rule will be explained with an accompanying graphic representation.

States Split

For example say there are 2 state routines from different days (on top of figure 3.9). At start the states have same values (0) then have different values after 4 states (the top state routine has continuous values 0 but the bottom state routine change state values from 0 to 1). In such a situation the states with the same values will be merged together and from the states with different values the states routine will be split. The bottom of figure 1 shows the state routine split into 2 different routines.

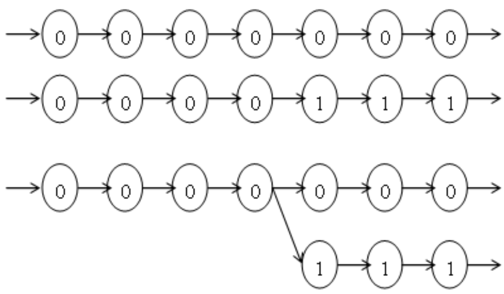


Figure 3.9: The state routine split from the states with different values

States Merge

States merge (figure 3.10) is the converse of states split. At start the states have different values (the top state routine has continuous values 0 but the bottom state routine changes state values from 1 to 0) then have same values (0) after 4 states. In such situation the states with different values were at different routines and from the states with same values the states routine will be merged together. The bottom of figure 3.10 shows the state routines merge to 1 state routine.

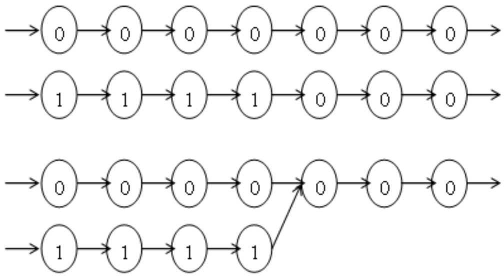


Figure 3.10: The state routine merged from the states with same values

The Complete Behavior Model

Through states split and merge the complete behavior model of the user will be learnt. In figure 3.11 the x-axis is the state value the y-axis is the count of the day. Figure 3.11 shows that all 7 days activity state routines composed to one model. Each small "." represents one state, which has state values "0" or "1". The first state (on the left side of day 7) is the initial state, it has no state value. The last state (on the right side of day 7) is the end state, it has no state value. The green "circle" means states with values "0" and the black "square" means states with values "1". The activities of the user in one week will be totally expressed in the activity state model. But in figure 3.11 there are some states which have the same state values and are consecutive. Through state merge these states can merge again, so the behavior model will be simplified further.

From figure 3.11 the general behavior model indicates that the user has lower activity from 0 to 10 hours, from 11 to 16 there is more activity, then from 17 to 24 hours some days show low activity and other days show high activity.

Finally, merging of states with same values will be introduced. In figure 3.11 there are some consecutive states with the same state values (or different values alternately), if these states can be merged together, so the states in the model will be reduced. It helps us to get a simpler and more general state model. The merge method will be introduced in figure 3.12 and 3.13.

In top of figure 3.12 there is a state routine with consecutive state values "0" and "1". These consecutive states will be merged to one state with the same state value. In bottom of figure 3.12 indicated the merged result. 4 states with values "0" merged to one state with same value and 3 states with values "1" merged to one state with the same value "1".

In top of figure 3.13 there is a state routine with consecutive state values "0" and "1" alternately. These consecutive states will be merged into one state. The merged result is indicated at the bottom of figure 3.13. The 7 states with alternately state values "0" and "1" will be merged into one state and the state has mixture value "0/1". After merging the states of the model will be reduced significantly. Furthermore if there has state with value "0" (or "1") different as neighbor

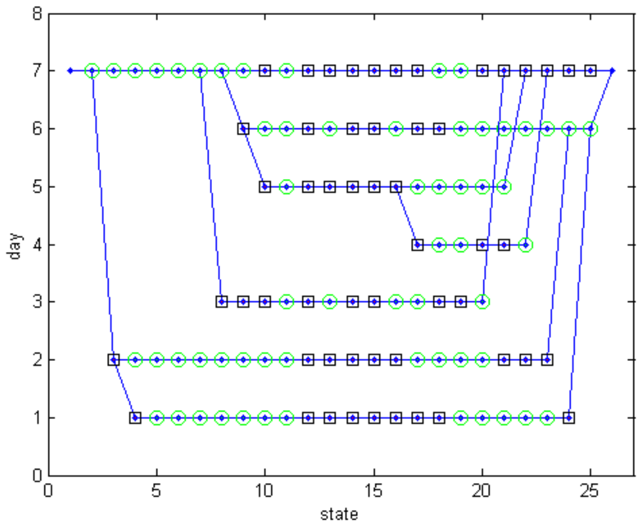


Figure 3.11: The state values in the structure of behavior model

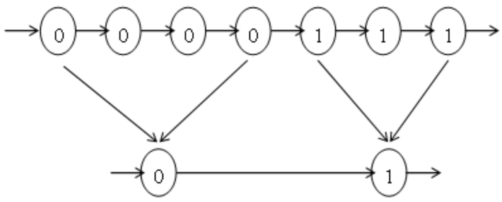


Figure 3.12: Merge the consecutive states with the same values

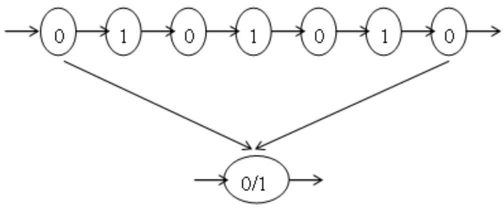


Figure 3.13: Merge the consecutive states with the alternately values

states that the state would be merged together with neighbor states. For example a state sets: 0 0 0 1 0 0, here the 4th state with value “1” would be merged together with neighbor states. The result was a mixture state. The advantage to make this kind of merging is that if the activity of the user changed, a little earlier or a little later, the merged state can include the changing. For example the user went to WC at night about 4:00, so there should be a state had state value “1” and the neighbor states had value “0”. Another day the user went to WC at night about 2:00, so there should be a state had state value “1” and the neighbor states had value “0”. The merged state had the same transition and self transition parameters. That means the activity the user went to WC at night kept the same. The time deviation of the activity ignored. The model is not influenced by the activity which happened a little earlier or a little later. Figure 3.14 shows the result.

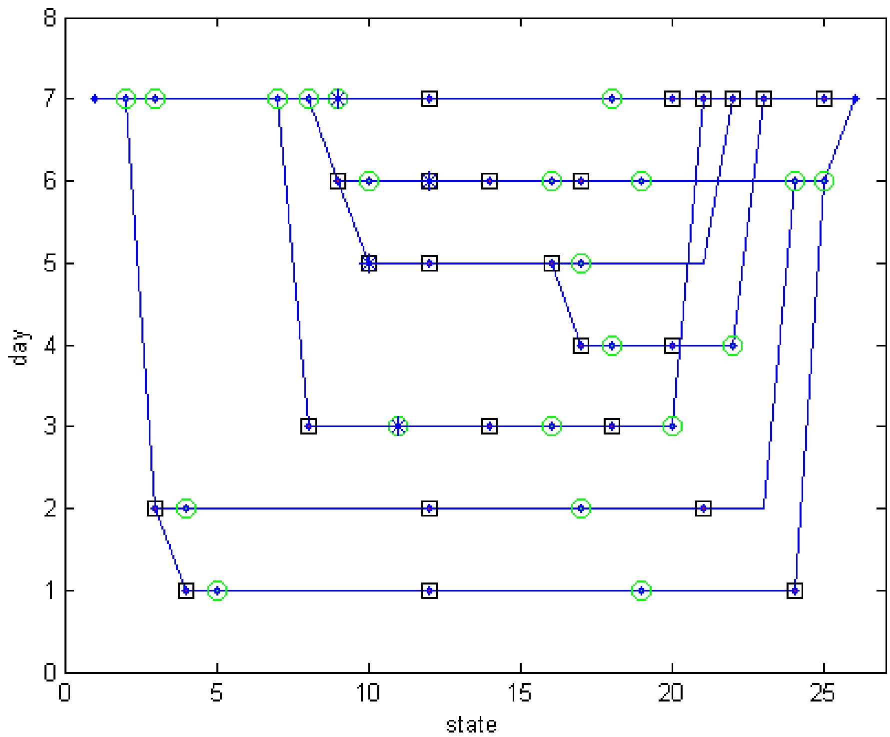


Figure 3.14: The behavior model after state merging. The states with green circles mean that the state has value “0”. The states with black squares mean that the state has value “1” and states with blue stars mean that the state includes mixture value “0/1”.

In figure 3.14 the x-axis is the state count and y-axis is the count of days. The states with green circles mean that the state has value “0”. The states with black squares mean that the state has value “1” and states with blue stars mean that the state includes value “0/1”. If there is a blue star in a green circle that means the first state is “0” and a blue star in the black square means that the first state has value “1”. Comparing figure 3.11 and 3.14 it is clear that the states count is reduced significantly.

Figure 3.14 shows the activities of the user on every day in one week. There is however a problem in that the chosen time interval ($T_{interval}$) of each state is one hour in this figure: for various shorter-lasting activities of the user, this $T_{interval}$ is too long. For example, the user might take a shower for 20 minutes and then go to the kitchen for breakfast for 30 minutes. If the $T_{interval}$ was chosen as one hour, these two different activities would lie within one interval, causing one state to include different behaviors. The resulting model would not be accurate enough to distinguish different behaviors. In order to create a behavior model in greater detail the parameter $T_{interval}$ must therefore be changed to a smaller value, for example 30 minutes (in the following chapter a discourse will deal with finding the optimal $T_{interval}$ value), which of course causes the state count to increase.

Up until now, the discussion has been about how to learn the behavior models of the user based on different activities. For regular behavior, the data from a single door contactor was enough to recognize an activity (when the user opens the door of his or her medicine box, the door contactor installed therein sends a signal to the controller indicating that the user took his or her medicine). But for random behavior (the user's daily routines), a single sensor is not enough because there are different rooms in the living environment and the user engages in activities in all the different rooms. A single motion detector could not cover all of these rooms. In order to create a behavior model of the user covering the entire living environment, many different types of sensors have to be used. The behavior model of the user will be learnt by merging the data from these different types of sensors.

4 Sensor Fusion

In the last chapter, a hidden Markov model (HMM) was used to learn the behavior model for random behavior. But because the used data came from only a single sensor (installed in one location), the model could only include the behavior of the user at one location. Within the user's living environment, however, there are different rooms such as the foyer, bathroom, living room, and kitchen. As it is impossible under these circumstances to learn a complete model using only a single sensor, sensor fusion will be introduced. This means installing different types of sensors in various locations and learning a behavior model relevant to the area of each sensor, then merging all of these individual models together. The result will be a behavior model for the entire living environment.

In this chapter, the disadvantages of using only measurements from a single sensor and the advantages of sensor fusion will be introduced. The various methods of sensor fusion will be discussed. Two ideas to sensor fusion were already presented (fusing the behavior models from each single sensor, or fusing the sensor state data from each sensor and then learning the behavior model). Based on these two ideas a new method of sensor fusion will be introduced. The application of sensor fusion with different types of sensors in different areas within the living environment of the user and the relationship between different kinds of sensors will be discussed. Finally, the results obtained using the chosen approach will be presented and reviewed.

4.1 Introduction to Sensor Fusion

The disadvantages of single sensor measurement generally are [Elm01, p. 4–5]: sensor defection causes information losing of observed objects, limited spatial coverage, limited temporal coverage, imprecision, and uncertainty. In the thesis the significant shortcoming of single sensor measurement is the limited spatial coverage. Because there are different rooms in the living environment of the user, a single sensor can only cover a limited area. The other locations will be not be covered, so the learnt behavior model is just presented for one location. In order to get a complete behavior model of the whole living environment of the user, many different types of sensors should be installed in different locations. Then the data from these sensors will be fused together (or the models fused together) to form a complete behavior model.

The definition of sensor fusion can be found in [Elm01, p. 3]: sensor fusion is the combination of sensory data or data derived from sensory data such that the resulting information is in some sense better than would be possible when these sources were used individually. Here the term better means for example more accurate, more complete, and/or more dependable. In the thesis sensor fusion was used in order to make the behavior model more complete.

Sensor fusion was introduced in the paper [Elm01, p. 5]. The author of the paper discussed the advantages of sensor fusion: robustness and reliability (multiple sensor enables the system to provide information even in case of partial failure), extended spatial and temporal coverage (one sensor can look where others cannot respectively), increased confidence (a measurement of one sensor is confirmed by measurements of other sensors covering the same domain), reduced ambiguity and uncertainty (joint information reduces the set of ambiguous interpretations of the measured value), robustness against interference (by increasing the dimensionality of the measurement space the system becomes less vulnerable against interference), improved resolution (when multiple independent measurements of the same property are fused, the resolution of the resulting value is better than a single sensor's measurement). In the thesis the advantage that extended spatial and temporal coverage of sensor fusion will be used. Through sensor fusion the location-related single sensor data will be merged together to form a behavior model of the whole living environment.

There are different types of sensor fusion [Elm01, p. 7–10]: C3I (command, control, communications, and intelligence) versus embedded real-time applications, three-level categorization (low-level fusion, intermediate-level fusion, high-level fusion), categorization based on input/output (data in data out fusion, data in feature out fusion, feature in feature out fusion, feature in decision out fusion, decision in decision out fusion), categorization based on sensor configuration(complementary, competitive, cooperative).

Above are the different types of sensor fusion but in the thesis what is really needed is state data from sensor fusion. This is because the topic of the thesis is automatic scenario detection; a hidden Markov model will be used to learn the daily activity routine of the user. What a hidden Markov model needs is state data. So here the state data is the connection point between sensor fusion and the hidden Markov model. Different types of sensors were installed in the living environment of the user and each sensor sent raw data to control. Translating and fusion of the raw data from single sensor to state data is the main challenge in the chapter. In chapter 3 a method was introduced to translate the sensor raw data into state data. The difficulty lies in how to fuse the state data from different sensors.

At least two approaches were considered in the thesis. In the first approach, the raw data from single sensor will be translated into state data. The state data will then be used to learn a behavior model. The output from each single sensor was used to learn one behavior model. Then sensor fusion was used to combine the models from the different sensors together to learn a complete model. In the second approach, the raw data from each single sensor will be translated into state data, then the state data from each single sensor will be fused together to form a state data set. Finally the state data set will be fused together to form a complete behavior model. In the following, the two approaches will be introduced and discussed in detail.

4.2 Sensor Fusion with Behavior Model

Just as figure 4.1 showed in the living room there are three sensors (with number 1, 2 and 3) installed in different positions. At first we presume that the three sensors are the same type of sensors, for example motion detector. Because they installed in different positions so each of them has individual view range in the living room. Because of the different view range the learnt models of each sensor are different. Figure 4.2 and figure 4.3 show the different models from each sensor.

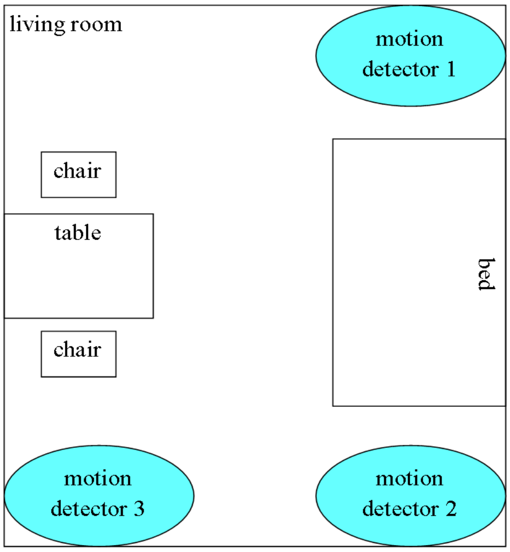


Figure 4.1: Living room with different sensors

Figure 4.2 shows the behavior model of the user from motion detector 1. X-axis from figure 4.2 is state sequence value for each day. Each circle in the model means a state, each state has a time duration 30 minutes. Y-axis from figure 4.2 is the day count. The first state on the day 7 is the initial state, it is just a symbol state, has not state value. The last state on the day 7 is a symbol state too. Between the initial state and the last state there are states with state value 1 or 0. Here in the figure the different state value did not show because in this section only the model structure was focused on. In this chapter the sensor fusion (SF) will be discussed, and the states with different value from different sensors will be fused together. That means that given state values may be changed after sensor fusion (SF), so it is not useful to show the state value here. On the other hand the model here is just used for structure comparing, in such a way that it is not important to show the state value. On day 7 the second state means the time interval from 0 to 0:30 at night. Then the third state covers the time interval from 0:30 to 01:00. The one before the last circle on day 7 covers the time interval from 23:30 to 00:00 the next day. In this way each state covers an activity situation of 30 minutes on a given day.

Just as chapter 3 discussed that if states from different routines have same state value then these states from different routines will be merged together. But if after one state there different state values appeared, so the routine will be split into two routines. Conversely if states from 2 routines have different state values at the beginning of each routine but then, starting at one state, that every state in the different routines have the same state values till the end of the routines, so the routines will be merged together from the state. Figure 4.2 at night from 00:00 to 0:30 all the states from 7 days have the same state value, so they are merged together, but after 00:30 the state routine is split into different routines. The day with number 1 splits from the routine group and the other 6 days stay at one routine.

Figure 4.3 is the structure of behavior models from motion detector 2 and 3. The structure of behavior model from sensor 2 is similar to the figure 4.2 but not the same. For example on the

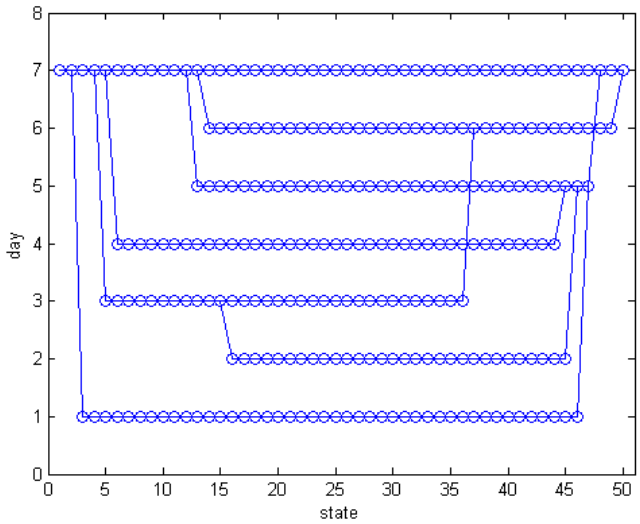
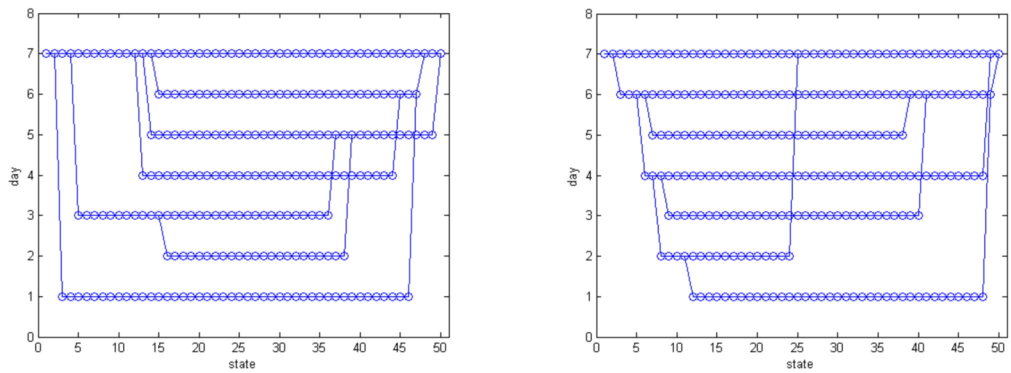


Figure 4.2: The structure of behavior model from sensor 1 in one week

day 2, 3, and 4 there are structure changes between figure 4.2 and figure *a* in figure 4.3. But generally they have the same structure. The reason is that the two motions detectors installed at the top and bottom corner in the living room, and have a similar view range.



(a) The structure of behavior model from sensor 2 (b) The structure of behavior model from sensor 3

Figure 4.3: The structure of behavior models (a), and (b) in one week

The structure of behavior model from sensor 3 has more differences in structure compared to figure 4.2 and the structure of behavior model from sensor 2. The changes of structure starts at the first state (not the initial state) on day 7. There are 2 different routines split to day 6 and 7. At last these routines are merged together at the last state.

The above figures they have different structures. The behavior models are different, in spite of the fact that they represent the same user. Just as we talked about above, for a more flexible and tolerant behavior model the above different models should be fused together, but how?

The first idea is to fuse the learnt models directly. But the problem is that because these models have different structures, so fusion of these models directly will change the individual model structure and make the whole model complicated. Another way is involves focusing not on the whole structure of the model but on each state of the model. That means these states will be fused initially, then the fused states will be used to learn the behavior model. The figure 4.4 shows the method.

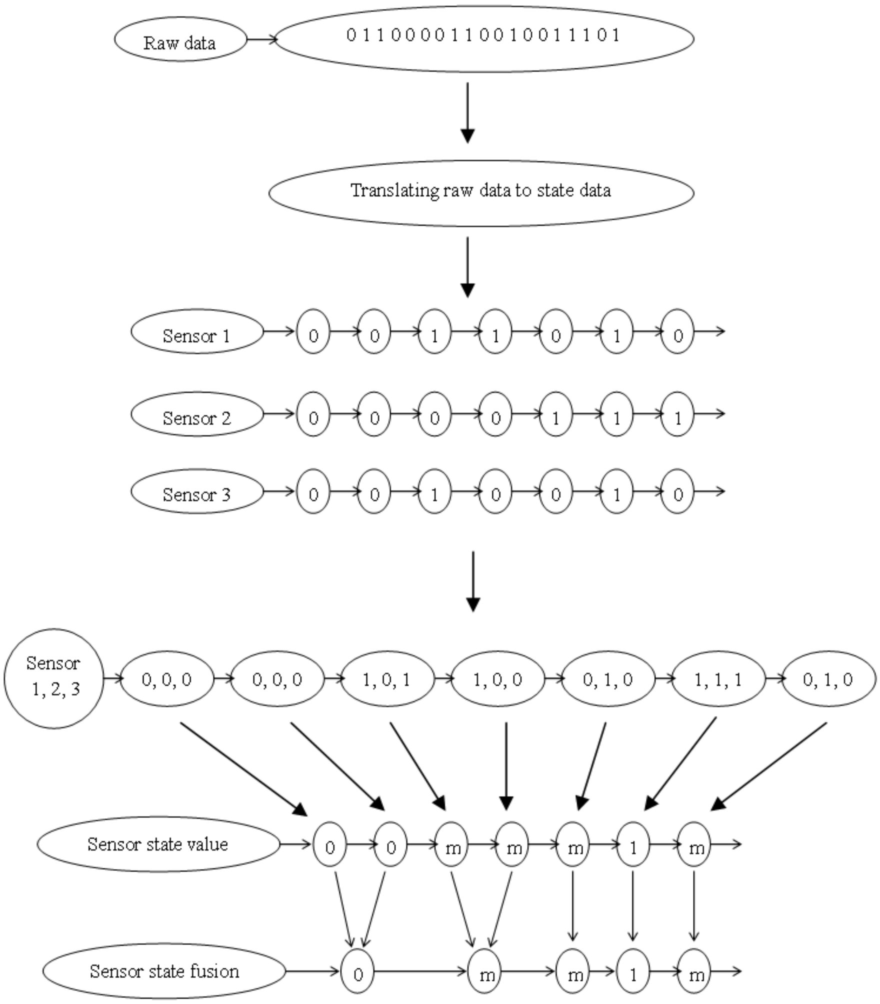


Figure 4.4: State merges with different state values

On top of the figure 4.4 raw data is shown. The raw data came from different types of sensors, in real situation there would be more than 10000 times the data amount in one single day. With a translation algorithm which was introduced in chapter 3 the raw data will be translated into state data. In this way each individual sensor has just 48 (or 96, depended on the chosen time interval for each state) value each day. An example result was showed in figure 4.4. There are three sensors. The sensor 1 is with state values 0, 0, 1, 1, 1, 1, 0. Then comes sensor 2 with state values 0, 0, 0, 0, 1, 1, 1. Last is sensor 3 with state values 0, 0, 1, 0, 0, 1, 0. The sensors have different state values. In the middle of the figure 4.4 the sensors state values merged together. Each oval means a state but within each state here there has three smaller states. For example in the first state there are three states with values 0, 0, and 0. In the third state there are three states with values 1, 0, and 1. These states values come from sensor 1, 2, and 3. These states will be merged together and the result shown in the bottom of the figure 4.4. The first sensor state value 0 comes from the first oval in the middle of the figure 4.4. That means if the states in an oval have all the same value (for example 0) so the value will be treated as the merged result. Here the first oval in the middle of the figure 4.4 has states which all have the same value 0, therefore the merged result in the state has value 0. If in an oval all the states have value 1 the merged state will have value 1. The situation happens at the sixth oval in the middle of the figure 4.4. Another situation is that if in an oval the states have different values, for example in the third oval in the middle of the figure 4.4 which has three states with value 1, 0, and 1. These states value cannot merge into one state value 0 or 1. So a symbol “m” will be used to indicate such situation, for example the third state with symbol “m” in bottom of the figure 4.4. It means in the state there are different state values from each individual sensor.

Up till now the raw data has been translated into sensor state values. Furthermore the consecutive states would be merged again in order to reduce the state amount. It is clear that if the consecutive states have the same state value, for example all are 0 (or 1), these states can merge together. But if the consecutive states have mixed values, then how to merge these states is a challenge. In the thesis a compatibility matrix (CM) [PCH08, p. 347] was utilized to deal with the problem. Especially when the consecutive states are all with mixture states value. For example on bottom of figure 4.4 the 3th, 4th, and 5th states all have mixture value, but the 5th state did not merge with them together. In the following section the explanation and solution will be given.

4.3 Compatibility Matrix for Sensor State Data Fusion

A compatibility matrix was introduced in paper [PCH08, p. 347]. The compatibility matrix (CM) was used to represent the correlations between the states of the nodes. For example two nodes i, j and two states variable S_a^i, S_b^j. Here a and b are the symbols in nodes i and j separate. The compatibility matrix $\psi_{ab}^{ij}(S_a^i, S_b^j)$ will be presented as:

$$\psi_{ab}^{ij}(S_a^i, S_b^j) = \begin{bmatrix} \psi_{ab}^{ij}(S_a^i = 0, S_b^j = 0) & \psi_{ab}^{ij}(S_a^i = 0, S_b^j = 1) \\ \psi_{ab}^{ij}(S_a^i = 1, S_b^j = 0) & \psi_{ab}^{ij}(S_a^i = 1, S_b^j = 1) \end{bmatrix} \tag{4.1}$$

Here the values of the principal diagonal are important parameters. If there are high values in the principal diagonal and small values in the outer diagonal that means the two nodes with high similarity of activity. In the thesis the compatibility matrix (CM) was used to fuse sensor state value. In the following the above mathematic formula will be changed and the new formula will be introduced with example. Because each sensor state has different values so the states variable S_a^i, S_b^j will be changed to S_A^i, S_A^j. Here A and B did not mean a single value but value sets. Now the compatibility matrix should be written as:

$$\psi_{A_kB_k}^{ij}(S_{A_k}^i, S_{B_k}^j) = \begin{bmatrix} \psi_{A_kB_k}^{ij}(S_{A_k}^i = 0, S_{B_k}^j = 0) & \psi_{A_kB_k}^{ij}(S_{A_k}^i = 0, S_{B_k}^j = 1) \\ \psi_{A_kB_k}^{ij}(S_{A_k}^i = 1, S_{B_k}^j = 0) & \psi_{A_kB_k}^{ij}(S_{A_k}^i = 1, S_{B_k}^j = 1) \end{bmatrix} \tag{4.2}$$

Here k means which pair in the sets take part in the comparing. Because in each sensor state data there are several values so the k has an interval from 1 to the amount of the values (n) in each sensor state. With the compatibility matrix the similarity of two consecutive states could be found. The count of each k pair will be added together, so the compatibility matrix should be written as:

$$\psi_{AB}^{ij}(S_A^i, S_B^j) = \Sigma_{k=1}^{k=n} \begin{bmatrix} \psi_{A_kB_k}^{ij}(S_{A_k}^i = 0, S_{B_k}^j = 0) & \psi_{A_kB_k}^{ij}(S_{A_k}^i = 0, S_{B_k}^j = 1) \\ \psi_{A_kB_k}^{ij}(S_{A_k}^i = 1, S_{B_k}^j = 0) & \psi_{A_kB_k}^{ij}(S_{A_k}^i = 1, S_{B_k}^j = 1) \end{bmatrix} \tag{4.3}$$

The similarity of the two consecutive states is related to the count of the compatibility matrix principal diagonal.

$$SL_{AB}^{ij}(S_A^i, S_B^j) = \frac{\Sigma_{k=1}^{k=n}\left(\psi_{A_kB_k}^{ij}(S_{A_k}^i = 0, S_{B_k}^j = 0) + \psi_{A_kB_k}^{ij}(S_{A_k}^i = 1, S_{B_k}^j = 1)\right)}{n} \tag{4.4}$$

Here SL_{AB}^{ij} means the similarity between two consecutive states i and j. It has a value interval from 0 to 1. If $SL_{AB}^{ij} = 1$ it means the two states are identical. If $SL_{AB}^{ij} = 0$ it means the two states are totally different. For example $S_A^i = [1110010011]$ and $S_B^j = [1001110110]$, so the $SL_{AB}^{ij} = (1+3)/10 = 0.4$.

Till now the similarity of the consecutive sensor states are known, based on the similarity value and with a predefined threshold value to decide whether the states should be fused together or not. Now another problem emerges: if some consecutive states are fused together the amount of the states will increase. Because each state has different sets values, with a similarity value between consecutive states so that only the relationship between the two states can be found. But it is not guaranteed that all the states in one fusion group have a similarity value relative to each other which is bigger than the predefined threshold value. To deal with the problem all the states in the fusion group have to be compared with each other. If one state was compared with other states and the similarity value was smaller than a predefined threshold value T_h (how to estimate an optimal threshold value will be introduced in the following), so the state would not add to the group. The method will be indicated in the following figure 4.5.

On the top of figure 4.5 is the sensor state value. Each value here is a sensor state set. If the similarity value SL between consecutive states are larger than a predefined threshold value T_h (in the following, how to estimate an optimal threshold value T_h will be introduced and discussed with examples) then both states will be fused together. Here the threshold value T_h influenced whether the consecutive states were fused or not.

4.3.1 Searching Optimal Threshold Value with State Data

In this section how to search optimal threshold value will be introduced and discussed. The first example and the used data came from 5 sensors with 12 states for each sensor. The threshold value T_h will be changed from 0.1, 0.3, 0.5, 0.7 to 0.9. Then data from 10 sensors was used with 12 states from each sensor. The threshold value T_h will be changed from 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 to 0.9. Third, the order of the states will be changed and the relationship between the fused states and the optimal threshold value T_h will be determined.

Figure 4.6 shows the states from 5 sensors. Based on the similarity and threshold value these states would be fused. On top of the figure are the states from 5 sensors. There are 12 states for each sensor. In the middle of the figure is the merged sensor state value. At the bottom of the

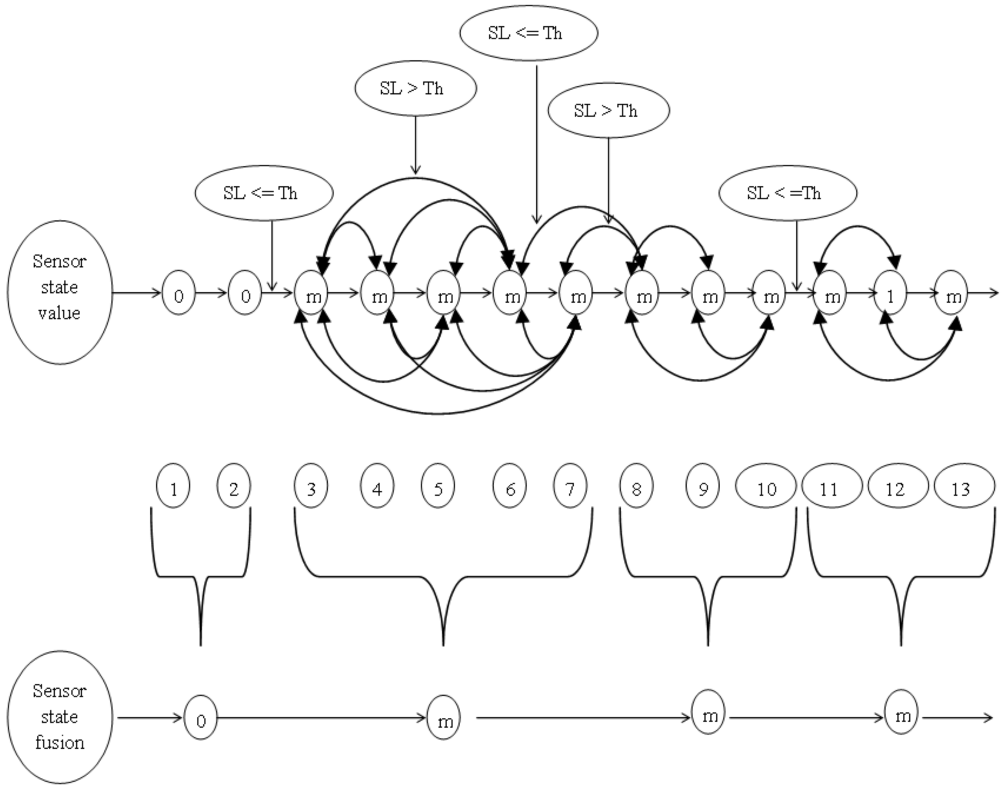


Figure 4.5: Sensor state fusion based on similarity and threshold value

figure are the fused states based on the similarity and the threshold value. If the threshold value T_h is 0.1 then all the states will be fused together. In fact there are 3 groups in the states. The states from 1 to 4 belong to one group. There are few states with value 1. That means the user has not much activity. The states from 5 to 8 belong to another group. There are more states with value 1. That means the user has more activity. The states from 9 to 12 belong to one group. There are few states with value 1. That means there is little user activity. But with the threshold value $T_h = 0.1$ all these states are fused together. That means the threshold value T_h is too low, and cannot distinguish different states in the above situations. If the threshold value $T_h = 0.3$ then the 12 states fused to 2 states. The state group from 9 to 12 was detected. But the state group from 1 to 9 was not detected. If the threshold value $T_h = 0.5$ then the 12 states fused to 3 states. All the 3 state groups were detected correctly. If the threshold value $T_h = 0.7$ the 12 states fused to 11 states. The state groups were not detected. If the threshold value $T_h = 0.9$ the 12 states fused to 12 states. The state groups were not detected.

The above result indicated that if the threshold value T_h chosen was closer to 0 all the different states will be fused together. The differences between states cannot be detected. If the threshold value T_h chosen was closer to 1 the states cannot be fused together. If the threshold value T_h is closer to 0.5 all the 3 different groups were detected. That means the threshold value $T_h = 0.5$ is optimal. But the result is just from an example. In the following different examples will be given in order to search the optimal threshold value T_h.

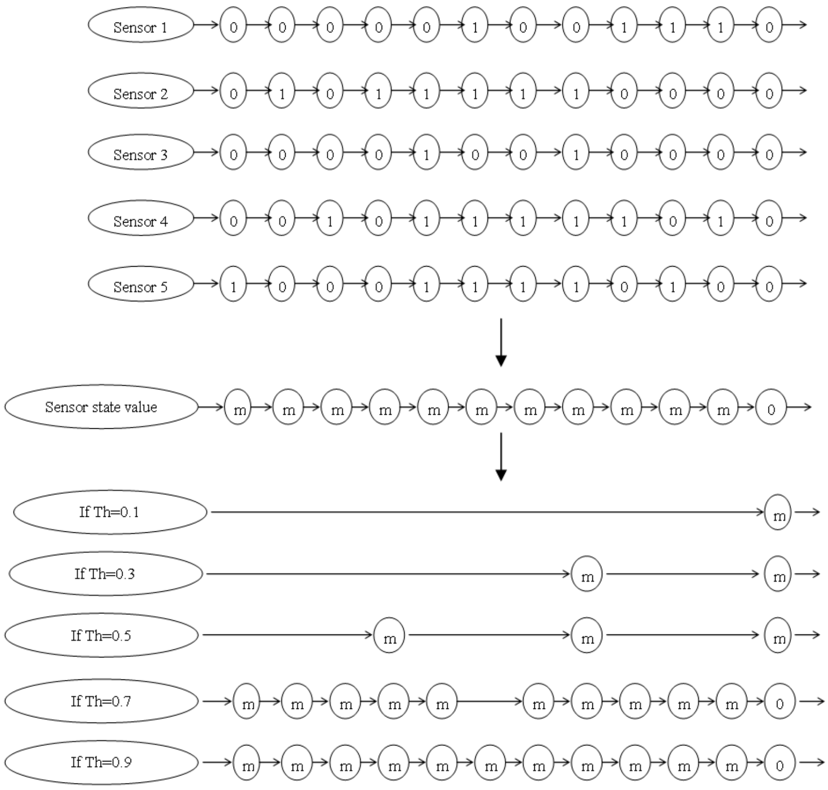


Figure 4.6: Sensor fusion with 5 sensors based on similarity and threshold value

Figure 4.7 shows the states from 10 sensors. Based on the similarity and threshold value these states would be fused. Similarly to the above example the states from 1 to 4, from 5 to 8, and from 9 to 12 belong to different groups with different activity situations. If the threshold value T_h is 0.1 then all the states will be fused together. If the threshold value T_h is 0.2 or 0.3 that all the states will be fused into 3 groups. But the state 5 was fused to the first group. In fact it belongs to the second group. If the threshold value T_h is 0.4 or 0.5 that all the states will be fused in 3 groups. If the threshold value T_h is 0.6 then all the states will be fused in 9 groups. If the threshold value T_h is 0.7 or 0.8 then all the states will be fused in 11 groups. If the threshold value $T_h = 0.9$ the 12 states fused to 12 states (i.e. no state fusion took place). The state groups were not detected.

The above results indicate again that if a threshold value T_h closer to 0 is chosen then all the different states will be fused together. The differences between states cannot be detected. If a threshold value T_h closer to 1 is chosen the states cannot be fused together. If the threshold value T_h is closer to 0.4 or 0.5 all the 3 different groups were detected. That means the threshold value T_h closer to 0.4 or 0.5 is optimal. In the following the order of the states will be changed in order to determine the optimal threshold value T_h.

Figure 4.8 shows the states from 10 sensors. Based on the similarity and threshold value these states would be fused. As in the above example the states from 1 to 4, from 5 to 8, and from 9 to 12 belong to different groups with different activity situations. But in the figure the order of the states were changed. In the first group the order between state 1 and 3 was changed. In

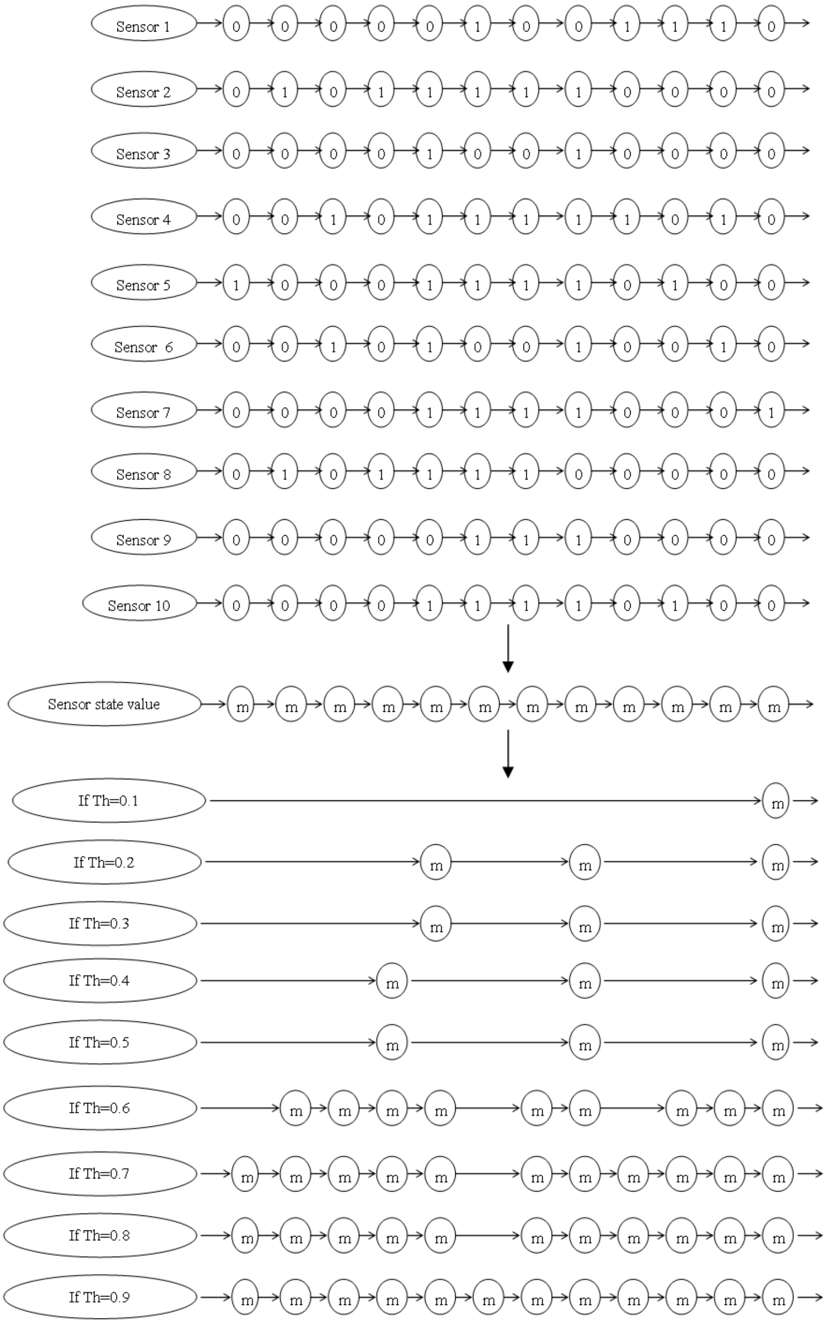


Figure 4.7: Sensor fusion with 10 sensors based on similarity and threshold value

the second group the order between state 5 and 8 was changed. In the third group the order between state 9 and 11 was changed. If the threshold value T_h is 0.1 then all the states will be fused together. If the threshold value T_h is 0.2 (or 0.3, 0.4, 0.5) then all the states will be fused correctly into 3 groups. If the threshold value T_h is 0.6 then all the states will be fused into 8 groups. If the threshold value T_h is 0.7 or 0.8 then all the states will be fused into 10 groups. If the threshold value T_h = 0.9 the 12 states fused into 12 states. The state groups were not detected.

The above result indicated again that if the threshold value T_h was chosen closer to 0 all the different states will be fused together. The differences between states cannot be detected. If the threshold value T_h was chosen closer to 1 the states cannot be fused together. If the threshold value T_h from 0.2 to 0.5 all the 3 different groups were detected. That means the threshold value T_h from 0.2 to 0.5 is optimal. Comparing with the above results the value interval of the optimal T_h increased. In the following the order of the states will be changed totally (the states will not be limited in the 3 groups) in order to determine the optimal threshold value T_h.

Figure 4.9 shows the states from 10 sensors. Based on the similarity and threshold value these states would be fused. The order of the states changed totally compared to the above examples. The state 1, 3, 8, and 11 in the figure have more states with value 1. The other states in the figure have more states with value 0. If the threshold value T_h is 0.1 then all the states will be fused together. If the threshold value T_h is 0.2 or 0.3 then all the states will be fused into 3 groups. If the threshold value T_h is 0.4 then all the states will be fused into 7 groups. If the threshold value T_h is 0.5 or 0.6 then all the states will be fused into 9 groups. If the threshold value T_h is 0.7 or 0.8 then all the states will be fused into 11 groups. If the threshold value $T_h = 0.9$ the 12 states fused to 12 states. The state groups were not detected.

The above result indicates again that if the chosen threshold value T_h is close to 0 all the different states will be fused together. The differences between states cannot be detected. If the chosen threshold value T_h is close to 1 the states cannot be fused together. If the threshold value T_h is increased from 0.1 to 0.9 the count of the states increases too. That means with higher threshold value T_h the consecutive states will be more difficult to fuse together. On the contrary with lower threshold value T_h all the consecutive states will be fused together. That means threshold values T_h close to 0 and close to 1 are not optimal. From above examples the optimal value should be in middle of 0 and 1. In the following the designed state data will be used to determine the optimal threshold value T_h.

4.3.2 Searching Optimal Threshold Value with Designed Data

In this section the designed data will be used to determine the optimal threshold value T_h. The reason for using designed data is to reduce the influence of the large changes in value between consecutive states.

Figure 4.10 shows the sensor signal counts from 12 sensors. There are 12 states for each state. The signal count of the first sensor changed from the first state with count 1 to the state 12 with count 12. The signal count of the sensor 2 changed from the second state with count 1 to the state 12 with count 11. And so on up to Sensor 12, which has signal count 1 at state 12.

Figure 4.11 indicates the sensor state values with signal count larger than (or equal to) 1 in figure 4.10. That means, in figure 4.10, if the signal count is larger than (or equal to) 1, the state value in figure 4.11 will be translated to 1. Then if the signal count was chosen as 2 then all the signal counts in figure 4.10 which are larger than (or equal to) 2 will be translated to 1. Then the signal count was chosen as 3. In this case all the signal counts in figure 4.10 larger than (or equal to) 3 will be translated to 1. And so on for each signal count up to 12, where all the signal counts in figure 4.10 larger than (or equal to) 12 will be translated to 1. Only the top right state of sensor 1 has state value 1. All the other states from all sensors have value 0.

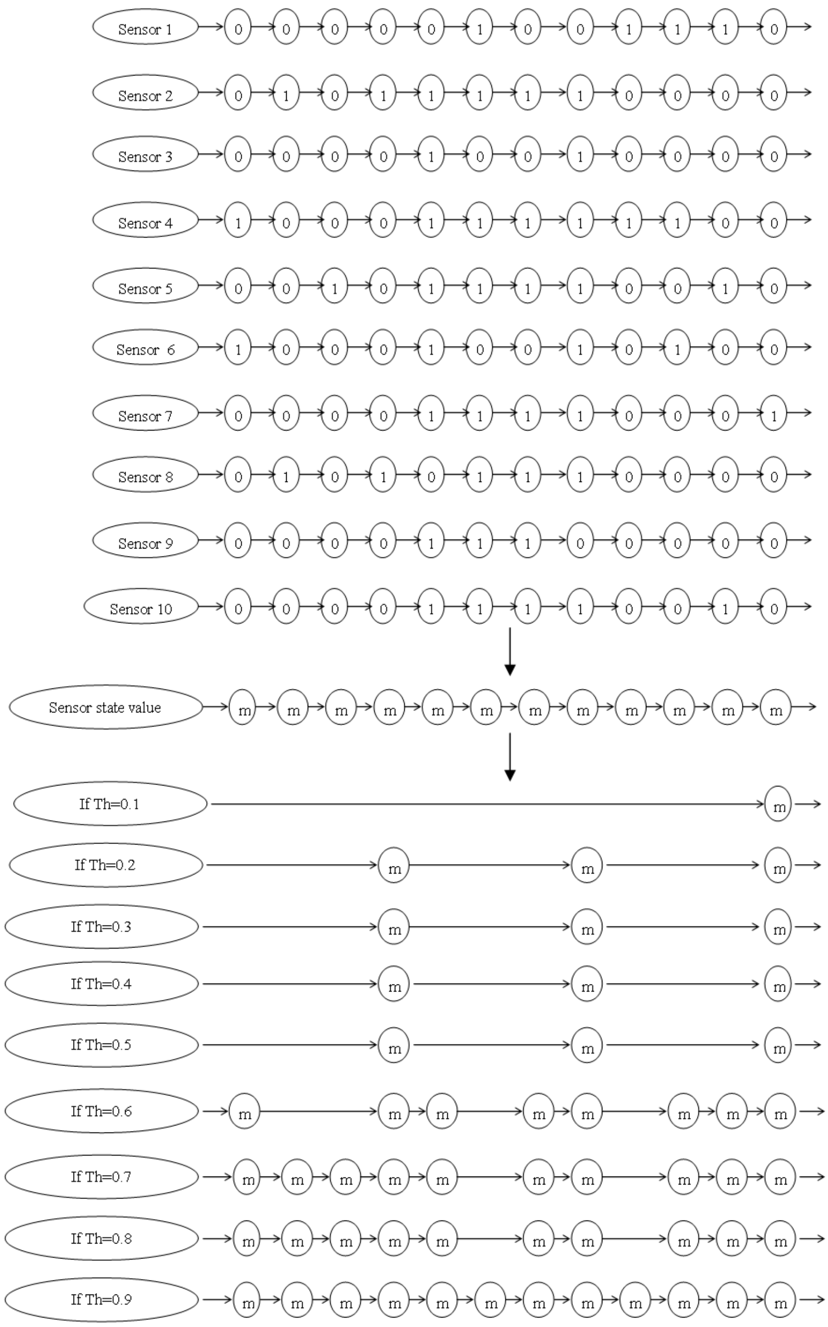


Figure 4.8: Sensor fusion with 10 sensors based on similarity and threshold value with states order changing

Figure 4.12 indicates the sensor states fuse results with the changing similarity, threshold value, and signal count. The x-axis of each small figure is the similarity from 0.1 to 1. The y-axis of each small figure is the state count after state fusion. From top to bottom and from left to right are the sensor state fusion results when the signal count was chosen from 1 to 12. For example the top left figure indicated that the signal count was chosen larger than (or equal to) 1. The

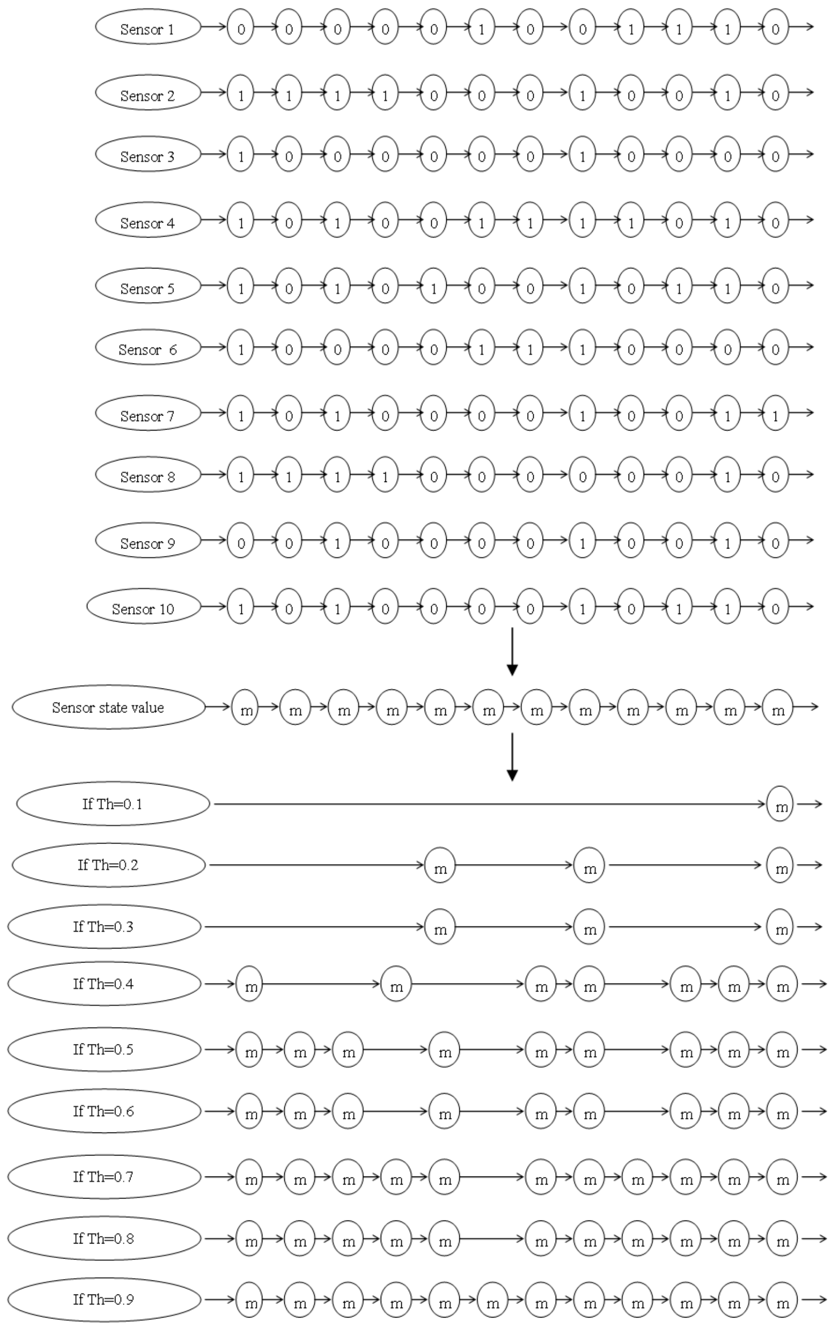


Figure 4.9: Sensor fusion with 10 sensors based on similarity and threshold value with states order changing again

state values are shown in figure 4.11. Then the threshold value T_h was changed from 0.1 to 1. The y-axis showed the state fusion result. If the threshold value T_h is between 0.1 to 0.5 the fused result is 2. That means all the 12 states fused to 2 states. If the threshold value T_h is between 0.6 to 0.7 the fused result is 3. That means all the 12 states fused to 3 states. If the threshold value T_h is 0.8 the fused result is 4. That means all the 12 states fused to 4 states. If

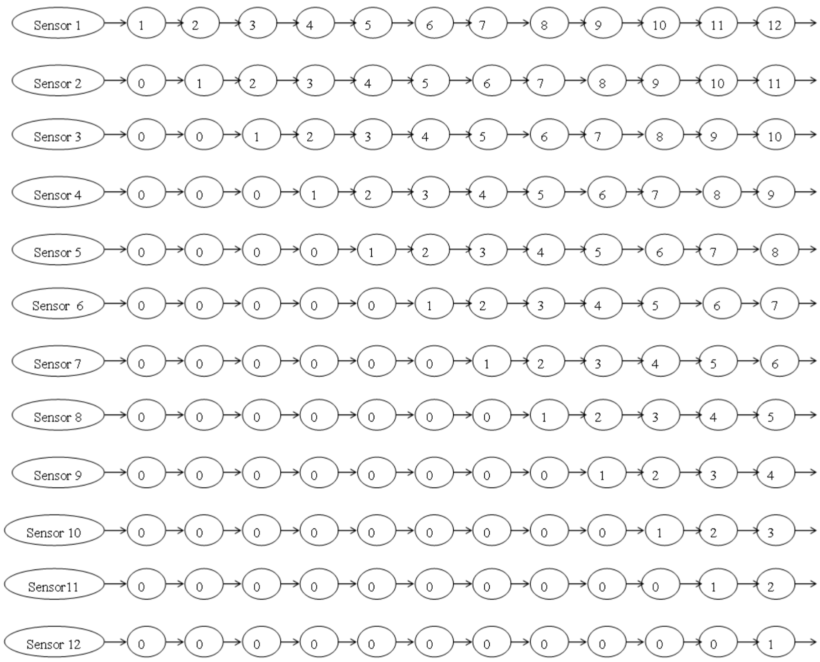


Figure 4.10: Sensor signal count with 12 sensors

the threshold value T_h is 0.9 the fused result is 6. That means all the 12 states fused to 6 states. If the threshold value T_h is 1 the fused result is 12. That means all the 12 states fused to 12 states (i.e. there was no sensor fusion). The second figure on the right side of the first figure indicates the state fusion result with signal count 2. And so on for each signal count, up to the figure on the bottom right which indicates the state fusion result with signal count 12.

Figure 4.12 indicates again that if the threshold value T_h chosen is close to 0, then most of the different states will be fused together. The differences between states cannot be detected. If the threshold value T_h chosen is close to 1 the states cannot be fused together. If the threshold value T_h increased from 0.1 to 1 the count of the states increases. About after the threshold value $T_h = 0.5$ the state counts increase exponentially. Together with the above examples the threshold value $T_h = 0.5$ is the optimal value.

In fact if the threshold value T_h is smaller than 0.5 the consecutive states with less similarity (smaller than 50% similarity between both states) will be merged together. In this thesis the threshold value T_h was chosen as 0.5 and only states with similarity larger than 0.5 will be merged together. This is the reason why the 5th state in figure 4.4 did not merge with 3th and 4th states. Its similarity to the 4th state (0.34) is less than the threshold value.

In figure 4.5 both states there are 0 so the sensor states fuse together. But the third state has a mixture value and the similarity value SL between second and third states is smaller than the threshold value T_h, so the states will not be fused together. From the third state to 7th state, the states are all mixture states. The SL value between the consecutive states are all bigger than the threshold value T_h. But in order to make sure the similarity value SL between all the states in the fusion group are bigger than T_h, each state has to be compared with all the other states. If two consecutive states have SL bigger than the T_h but if the last consecutive state compares to

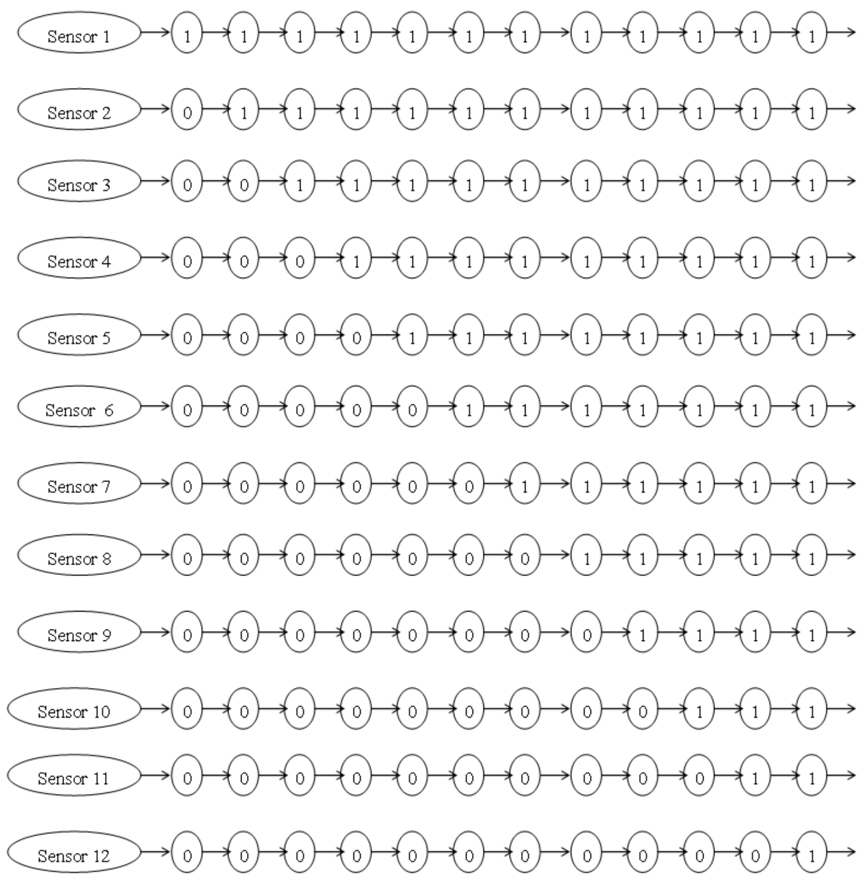


Figure 4.11: Sensor state value with signal count bigger than (or equal to) 1

all other states in the group and there is one SL smaller than T_h, so the last consecutive state will not be included in the group. In figure 4.5 the 8th state and 7th state have the same situation. The SL between 7th and 8th is bigger than the T_h but the SL between 6th and 8th states is smaller than the T_h, so the 8th state will not be included in the group (states from 3 to 7).

On the other side in figure 4.5 the states 11, 13 are states with mixture value and the state 12 includes values which are all 1. If the similarity value SL between these states is bigger than the threshold value T_h, these states will be fused together.

Now on bottom of figure 4.5 there are only 4 states: one state with value 0 and the other 3 states with mixture value. The 4 states cannot be fused again because the SL value is smaller than the T_h between the consecutive states and the states between groups.

4.4 Behavior Model with Sensor State Data Fusion

Till now the states values from different sensors have been merged together. With these merged states the behavior model will be learnt. Because these states from different sensors are fused

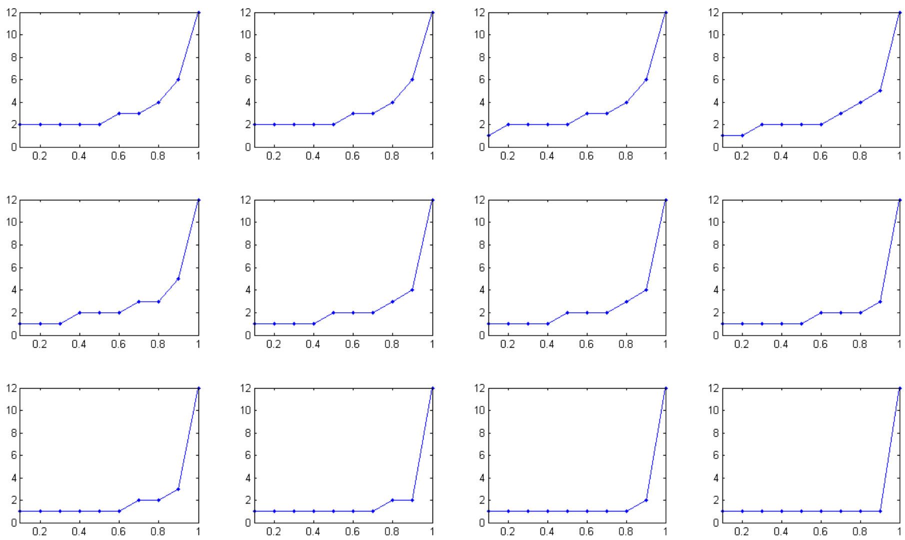


Figure 4.12: Sensor state fuses according similarity, threshold value, and signal count

together, there will be no single sensor state value. The above problem, that behavior models from individual sensors produce different structures, is solved. In the following a behavior model will be introduced. The structure problem described above will be avoided in the behavior model.

At first the structure of the behavior model will be introduced in figure 4.13. In the structure has not split or merge between routines anymore. With the way the structure of model keeps unchanged. For example take the structure of a behavior model covering 7 days. The structure is shown in figure 4.13.

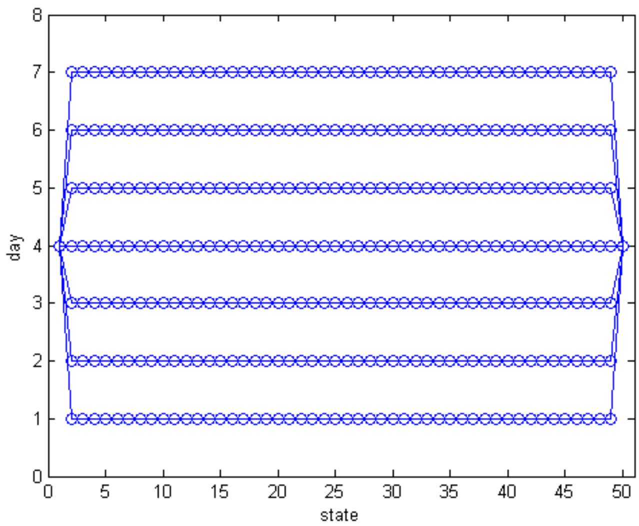


Figure 4.13: The structure of behavior model

The x-axis of figure 4.13 represents the sate value for each day and the y-axis is the day count.

There are 7 days in the structure of model. Each circle means a state and each state has time duration 30 minutes. The first and last states in day number 4 are the initial states, they have no value. Because each state has its own value, figure 4.13 cannot on its own illustrate the state information. In figure 4.15 the state value will be displayed in the model.

In the previous section the problem of how to determine optimal parameters for the $T_{interval}$ and for the threshold value T_{th} has not been solved. In this chapter the method will be introduced which solves this problem. First 7 daily activity routines of the user will be used to form the structure of the behavior model. This is shown in figure 4.13. In figure 4.13 each state has a time duration $T_{interval}$ of 30 minutes. But in order to search the relationship between the $T_{interval}$ and the state count, the $T_{interval}$ value will be changed from 1 minute to 60 minutes (from 1 minute to 5 minutes, and then with time interval 5 minutes up to 60 minutes). At the same time the threshold value T_{th} will be changed from signal count 1 to signal count 20 (from count 1 with interval 1 up to count 10, and then from count 10 to count 15, then to count 20. Each signal count has a time duration of 3 seconds, so the threshold value T_{th} is from 3 seconds to 60 seconds). Figure 4.14 shows the result.

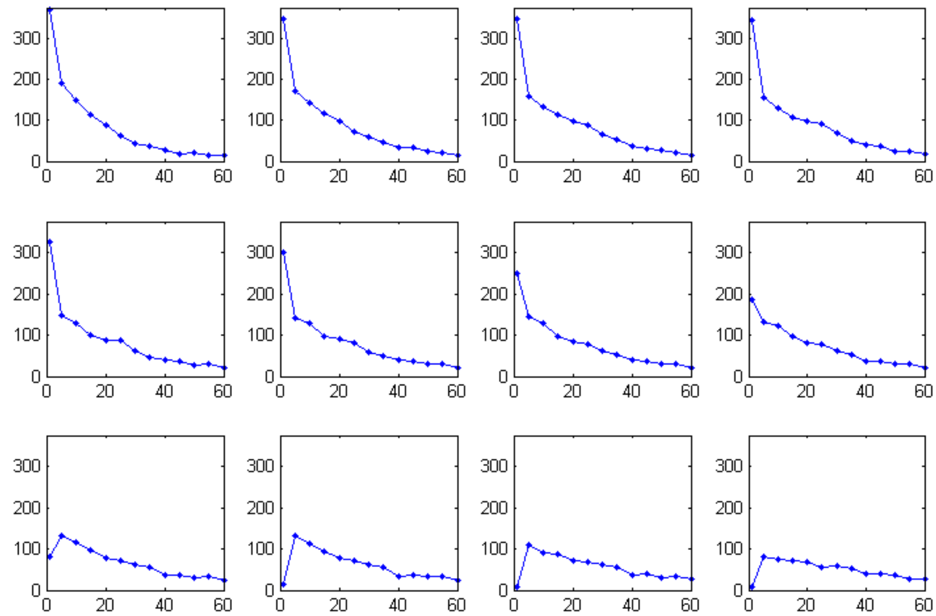


Figure 4.14: The relationship between $T_{interval}$, T_{th}, and state count of the behavior model

The small figure on top left of the figure 4.14 indicates the relationship between $T_{interval}$, T_{th}, and state count of the behavior model when the T_{th} was chosen as 3 seconds. The X-axis is the $T_{interval}$ value. It changed from 1 to 60 minutes. The Y-axis is the state count of the behavior model. It is more than 300 after state merging. When the $T_{interval}$ was changed from 1 minutes to 5 minutes the state count was reduced to under 200. When the $T_{interval}$ was changed to 60 minutes the state count was reduced to about 20. The top left figure indicates that with increased $T_{interval}$ value the state count is reduced. Especially when the $T_{interval}$ was changed from 1 minute to 5 minutes, the state count was reduced by more than 100.

The small figure on the right side of the top left corner of the figure 4.14 indicates the relationship between $T_{interval}$, T_{th}, and the state count of the behavior model when the T_{th} was set to 6 seconds. It is similar to the small figure on top left of the figure 4.14 with T_{th} set to 3 seconds. Furthermore the third small figure on the first row of the figure 4.14 indicates the relationship between $T_{interval}$, T_{th}, and state count of the behavior model when the T_{th} was set to 9 seconds. It is similar to the left two figures. The small figures on the second row of the figure 4.14 are all similar to each other but the state count was only reduced from more than 300 to under 200 when the $T_{interval}$ was set to 1 minute.

The 8 small figures on the first and second row of the figure 4.14 are similar when the $T_{interval}$ was chosen from 5 minutes to 60 minutes. The T_{th} value changed from 3 seconds to 24 seconds but the figures remained similar. That means the T_{th} value did not influence the relationship between $T_{interval}$ and state count much when the T_{th} value was smaller than 24 seconds.

The small figures on the third row of the figure 4.14 changed more especially when the $T_{interval}$ was set to 1 minute. The state count was about 100 on the bottom left small figure when the $T_{interval}$ was set to 1 minute and the T_{th} value was 27 seconds. When the $T_{interval}$ was chosen from 5 minutes to 60 minutes the figure would again be similar to the other figures above. The same sort of changes occurred with the others figures on the third row of the figure 4.14. The reason is that when the $T_{interval}$ was set at 1 minute there would not be enough signals in the short time interval. At the same time the T_{th} value must be more than 27 seconds (9 sensor signals if each signal has time duration 3 seconds). That means if in 1 minute there are less than 9 signals in the state then the state value will be treated as 0. In the 7 activity routines there are only a few states which have more than 9 signals in 1 minute, so most of the states values would be treated as 0. These states with same value would be merged together, so the state count would be reduced to less than 100.

In the second small figure of the third row in the figure 4.14 the state count is reduced to about 7 when the $T_{interval}$ was set at 1 minute and the T_{th} value was about 30 seconds. The other 2 figures on the third row have the same result when the T_{th} value was about more than 30 seconds and the $T_{interval}$ was set at 1 minute.

From above introduction, discussion, and analyzing the $T_{interval}$ should not be set too small (closer to 1 minute). In fact if the $T_{interval}$ was chosen as 1 minute there should be 1440 states ($60 \times 24 = 1440$ states) on daily routine. The state values did not have much difference as the sensor raw data. Furthermore after state merging there would be still many states (from about 100 to 300) in the model. That made the behavior model complicated. According to the small figures in the figure 4.14, when the $T_{interval}$ value was between 20 to 40 minutes the state count reduced regularly. So the $T_{interval}$ value between 20 to 40 minutes was optimal. The blue lines in the 12 small figures in the figure 4.14 were similar when the $T_{interval}$ value was between 20 to 40 minutes. That means the T_{th} value did not influence the state count of the behavior model much. But generally with increased T_{th} value, the state count is reduced. Furthermore, taking into consideration the noise signal and the loss of signal in the real world, so the T_{th} value between 9 seconds to 24 seconds would be optimal.

The approach introduced above describes how to determine optimal parameters in such situation. In reality, the result in different situations perhaps changes a little. But the introduced approach should be adaptable to different situations.

With above introduced approach the figure 4.13 could be formed to figure 4.15 with real state value. Figure 4.15 is the state value in the structure of the behavior model of sensor 1. Because each state in the behavior model has a concrete value (0 or 1), but the first and last initial states are without value, in order to reduce confusion, so they will be change to points (in figure 4.13 they are shown as a circle). In figure 4.15 the circle means state with value 0 and the square means state with value 1. From state 2 to state 13 there are mostly circles. That means the time period from 0:00 to 6:00 (each state has time duration 30 minutes). From state 2 to state 14 there are 13 states, and state 2 means the time interval from 0:00 to 0:30, so from state 2 to state 14 (time duration from 0:00 to 6:30) the user exhibits little activity in the detection area of sensor 1. From state 15 to state 44 (time duration from 7:00 to 21:30) there are mostly squares in figure 4.15. That means the user has more activities in the day time and in the evening. Only on day 1, 2, and 3 are there some circles in the afternoon. That means the user has a rest in the afternoon on some days. From state 45 to state 49 (time duration from 22:00 to 24:00) there are again mostly circles in figure 4.15 . That means the user is again mainly resting during this period.

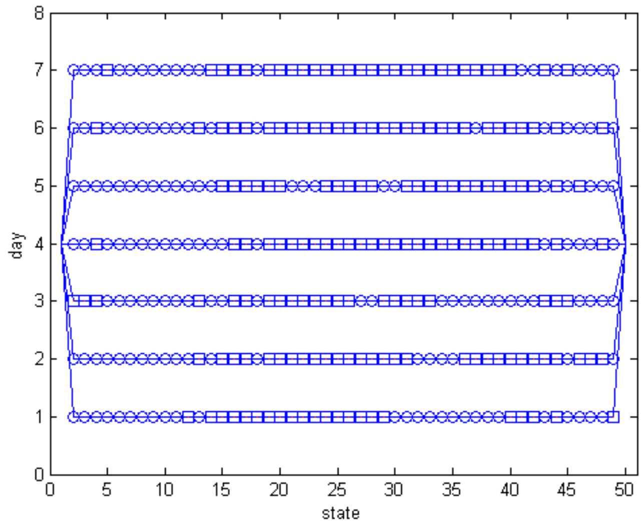


Figure 4.15: The state value of sensor 1 in the structure of the behavior model

The figure 4.15 indicates the activities of user in all 7 days. But because the model is just from sensor 1, so the model can only shows user activity in the detection area of sensor 1. The figure 4.16 shows user activity in the detection area of sensor 2.

The figure 4.16 is the behavior model of the user from sensor 2. A comparison of figure 4.15 and figure 4.16 shows marked similarities. This is because the detection areas of the sensors overlap.

The figure 4.17 is the behavior model of the user from sensor 3. The figure 4.17 is very different from figure 4.15 and figure 4.16. In figure 4.17 there are mostly squares in all of the behavior model. That means the sensor 3 detects activities in its detection area all the time. The detection area of sensor 3 is in the middle on the right side of the living room. The bed is located there in figure 4.1.

Comparing the figure 4.17 with the figure 4.15 and the figure 4.16 they are very different. It has already been noted that the figure 4.15 and the figure 4.16 are very similar. This is because the detection areas of the sensors (1 and 2) overlap. But sensor 3 has a different detection area from sensor 1 and 2, so the behavior model from sensor 3 is very different. So in order to get a complete behavior model of the user the behavior model of all three sensors will be merged together. The following figure 4.18 is a mixture from all above three figures.

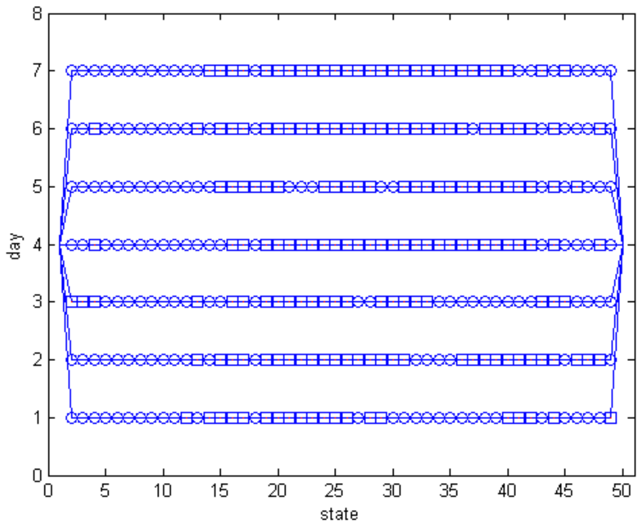


Figure 4.16: The state value of sensor 2 in the structure of the behavior model

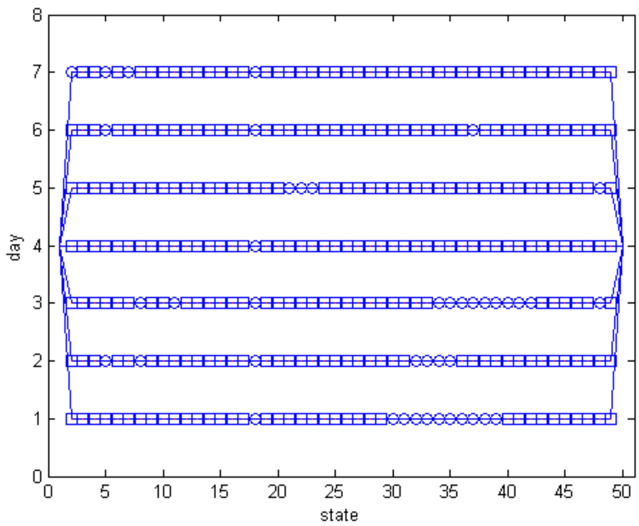


Figure 4.17: The state value of sensor 3 in the structure of the behavior model

Figure 4.18 has the same structure as figure 4.13 but with some star and square symbols instead of circle symbols in some positions. In figure 4.18 each circle, star, and square symbol has more information than figure 4.13. Here each symbol not only means a state but also means the state value at the same time. For example a circle here means state value 0, square means state value 1, and star means state value mixture (state data 0 and 1 are mixed in the same state).

Figure 4.18 indicates more information about the activity of the user. From states 2 to 14 each day there are more star symbols, that means the user has mixture activity (the activities of the user happened only in one or two sections of the whole detection area of all three sensors, i.e. some user activity was detected by only one or two of the three sensors). This is especially noticeable in the days with numbers 1, 2, 3, 4, 6, and 7. These days all have states with squares.

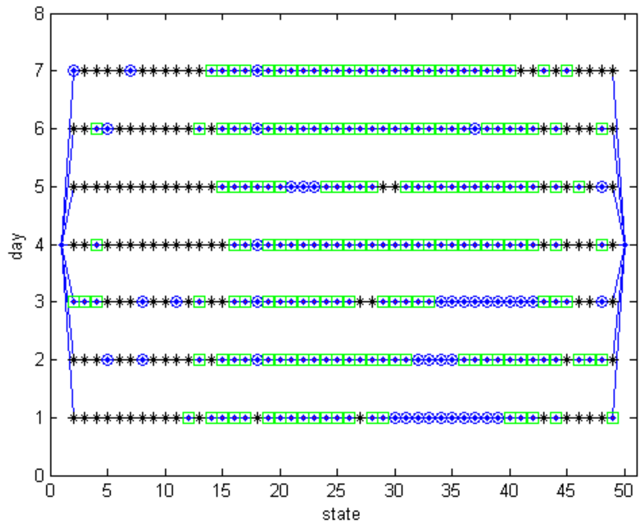


Figure 4.18: The fused state value of sensors 1, 2, and 3 in the structure of the behavior model

That means the user has activity that is detected by all 3 motion detector sensors simultaneously. Perhaps the user goes to WC at night. The symbols from state 2 to state 14 each day are mostly star symbols. This indicates that the user probably has a sleep disorder. In the daytime from state 15 to state 44 there are mostly squares, that means the user has activity in the living room. Only in the afternoon from state 30 to state 42 are there some circle symbols. That means the user is resting (or is not at home) in the afternoon and evening some days. From example the days numbered 1, 2, 3, and 6 have some circle symbols in the afternoon and evening. At night in states 43 to 49 there are more star symbols again, which mean the user activity is confined to a limited area in the living room. If the user has activities in the whole living room all the 3 sensors should have state value 1, so the state fusion value should be 1. If one or two sensors have state value 0 and other sensors have state value 1, so the state fusion value is a mixture with a star symbol. Each sensor has a limited observation area within the living room. If the user has activity in the location which belongs to one sensor observation area then the sensor state value will be 1. If the user has activity only in observation areas from some sensors but not all sensors in the living room, so the fusion value from all sensors will be a mixture value with a star symbol. Figure 4.18 shows the state value fusion result from different sensors. There are state values 0 and 1 and mixture values with symbols circle, square, and star. Based on the similarity between states and the threshold value, these states should be fused again. The result shows in figure 4.19.

Figure 4.19 is the behavior model. In figure 4.19 all the states with similarity value bigger than threshold value will be merged together. The states count is thereby reduced from about 340 to about 80. The figure 4.19 with the reduced state count indicates a brief and clear behavior model of the user. The behavior model is a sensor fusion model from three behavior models of sensor 1, 2 and 3 in the living room. Because the three models have different structures, it is difficult to merge or fuse them together. But in the thesis a new method has been introduced which allows fusion of the sensor state values. Based on the similarity value and the threshold value, a judgement is made whether to fuse the consecutive states or not. Just as discussed above, each sensor has a behavior model but these behavior models have different structures. It is difficult to merge or fuse the different structures. So with the sensor fusion method the state values from

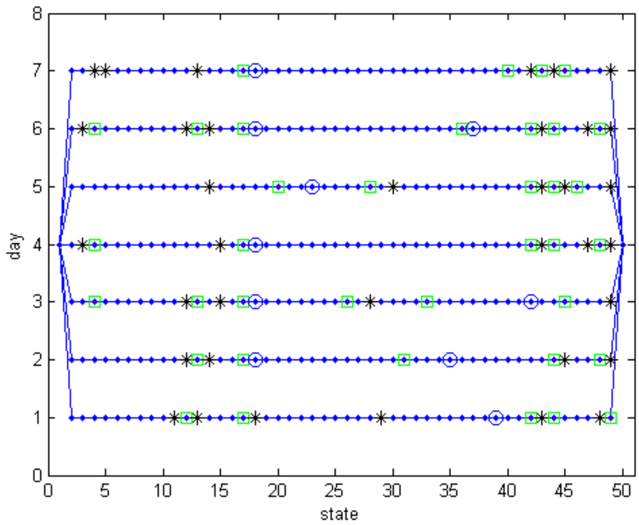


Figure 4.19: The behavior model of sensors 1, 2, and 3

different sensors will first be merged and fused. The problem that merges the different structure from different sensors will be avoided.

4.4.1 Fusing the Sensors in Different Rooms

Above sensor fusion (SF) have been discussed with the sensors in the same room, for example motion detectors in a living room. But if there is more than one room in an apartment and if we want to get a complete behavior model of the user we have to find a way to achieve fusion of these sensors in different rooms. Such a situation is represented in figure 4.20. There are three motion detectors in the living room and a further detector in the entrance hall. Because these sensors are installed in different rooms, when user changes his location from one room to another room, for example from living room to entrance hall, so the sensors in the living room will show no activity but the sensor in the entrance hall will show activity there. In the following the model from the sensor installed in the entrance hall will be introduced. Then the merged model from sensors from different rooms will be introduced.

Figure 4.21 is the behavior model from motion detector 4 installed at entrance hall. From states 2 to 14 on each day there are mostly circle symbols, which means in the time interval there is no activity in the entrance hall. This corresponds to reality. At night the user sleeps in living room and nobody goes into in the entrance hall. But on day 6 at the state 4 there are a square symbol, which means activity from the user at the entrance hall. On day 4 at state 4 there is square symbol too. These states are located at the time interval from 0:00 to 6:30. What has happened at night with the user? If we look at the figure 4.20 we will find that there is a bathroom and WC leading off the entrance hall. The user gets up at night and goes to the WC. That is the reason why there are activities at night in the entrance hall.

In the daytime there are mostly activities (with square symbols) at the entrance room, only in the afternoon are there some states with circle symbols (no activity). In the later evening it will be quiet again in the entrance hall. But on day 1 at the state 49 there is a square again, which

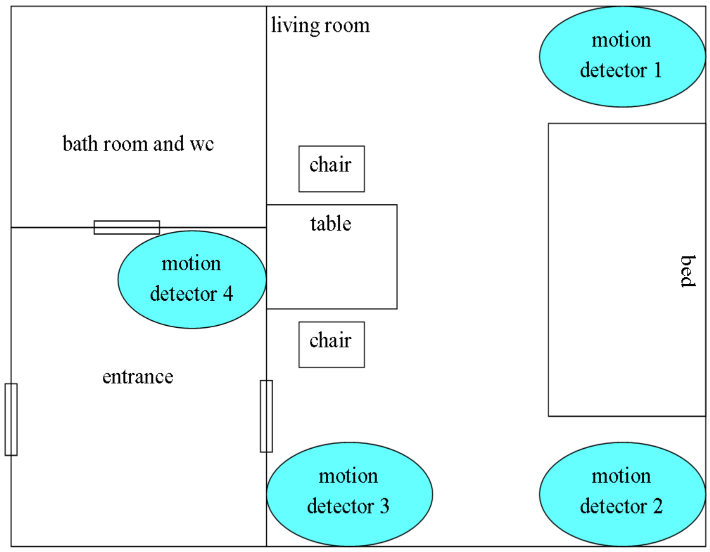


Figure 4.20: Sensors in living room and entrance

means the user has activity in the entrance hall. This is perhaps because the user goes to the WC at night.

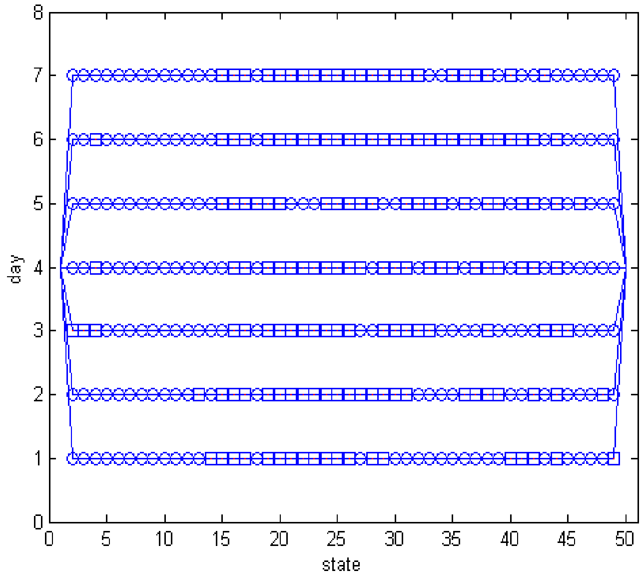


Figure 4.21: The state value of sensor 4 in the structure of the behavior model

Figure 4.22 shows the fused sensor values of sensors 1, 2, 3, and 4 translated to state value in the structure of the behavior model. The extra motion detector 4 is installed in the entrance hall. Above figure 4.18 shows the fused sensor value of sensors 1, 2, and 3 translated to state value in the structure of the behavior model. Now because of data from sensor 4 at entrance hall may be fused together with data from sensors (with number 1, 2, and 3), so the state values will be changed.

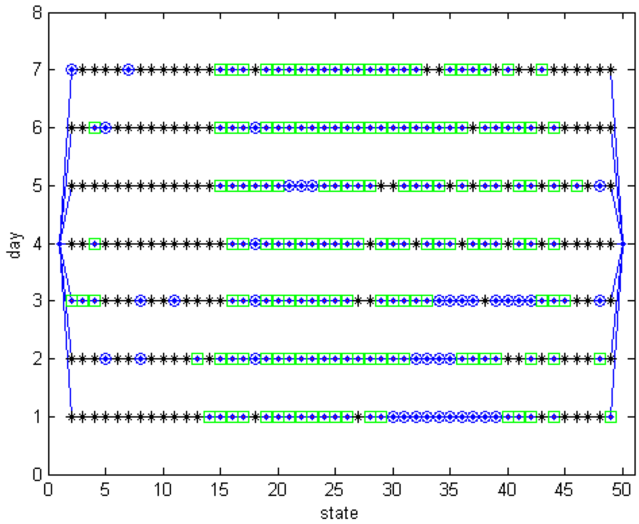


Figure 4.22: The fused state values of sensors 1, 2, 3, and 4 in the structure of the behavior model

Figure 4.22 indicates that at state from 2 to 14 there are mostly stars, which mean mixture activity from the user. Then in the daytime there are mostly square symbols, which mean activities from the user in the living room and entrance room. States with circle symbols occur only in the afternoon. In the later evening there are again mostly star symbols, which mean there is no user activity in either the living room or entrance hall areas. On day 1 at state 49 (indicating the time interval from 23:30 to 24 o'clock) there is square symbol. This means user activity in the living room and entrance hall. This is probably caused by the user going to the WC at night.

If the figure 4.18, figure 4.21, and figure 4.22 were compared to each other the result from sensor fusion (SF) with sensors from different locations will be showed clearly. For example in figure 4.18 there is activity from the user between state 2 to state 14 on day 1, 2, 3, 4, 6, and 7. But in figure 4.21 there is only activity between state 2 to state 14 on day 1, 2, 3, 4, and 6. These instances of activity happened in the same states on the same day, so in figure 4.22 in the same states there is activity too. That means the user has activity on day 1 ,2 ,3 ,4, and 6 in the same states in both the living room and entrance hall. The other states between state 2 to state 14 on day 7 in figure 4.18 with square symbols only mean that these instances of activity happened in the living room but not in the entrance hall. Figure 4.22 indicates the activity information from user at all living room and entrance room. The model is more complete than figure 4.18 and figure 4.21.

In figure 4.23 the consecutive states with greater similarity will be merged together. The state count is reduced from about 340 to 80. This is a brief and clear behavior model of the user with 4 sensors in different rooms.

Next, motion detector sensors were installed in the rest of the living environment, and the sensor data can be fused together to form a complete behavior model of the user. Figure 4.24 shows the total living environment of the user, with an extra motion detector (numbered 5) installed to cover the bathroom/WC area.

Figure 4.25 is the behavior model from motion detector 5 installed at bath room and WC. From state 2 to 14 on each day there are mostly circle symbols, which means in the time interval (from

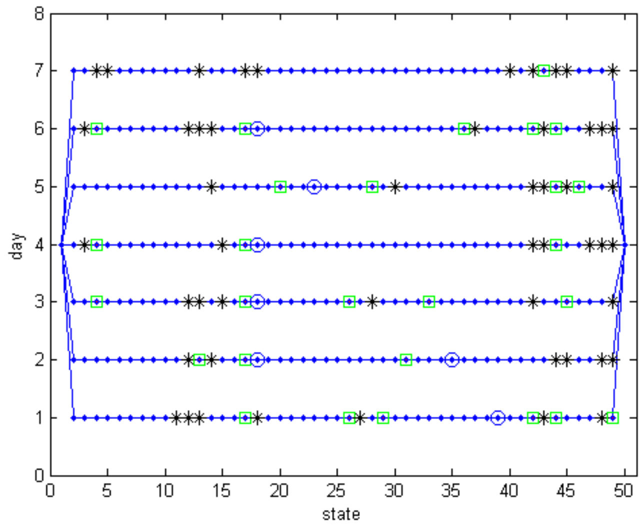


Figure 4.23: The behavior model of sensors 1, 2, 3, and 4

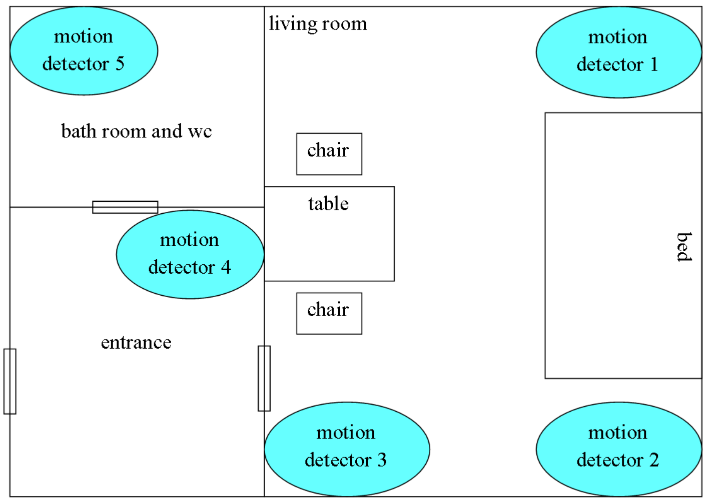


Figure 4.24: Sensors in living room, entrance, bath room and WC

0:00 to 6:30) there is seldom activity in the bathroom/WC area. At night the user sleeps in the living room and there is no activity in the bathroom/WC area. But on day 1, 2, 3, 4, 6, and 7 there are some square symbols between state 2 and state 14, which mean activity from the user in the bathroom/WC area during the night. In the daytime there are different symbols in the behavior model of the user (with square and circle symbols) in the bathroom/WC. In the later evening there will be more circle symbols again at the bathroom/WC area.

Figure 4.26 shows the behavior model from the fusion data from all motion detectors 1, 2, 3, 4, and 5. From state 2 to 14 on each day there are mostly circle symbols, but on day 1, 2, 3, 4, and 6 there are square symbols. That means the user gets up at night, through the entrance and goes to use the WC. This is a complete scenario of the user (goes to WC at night). Various different

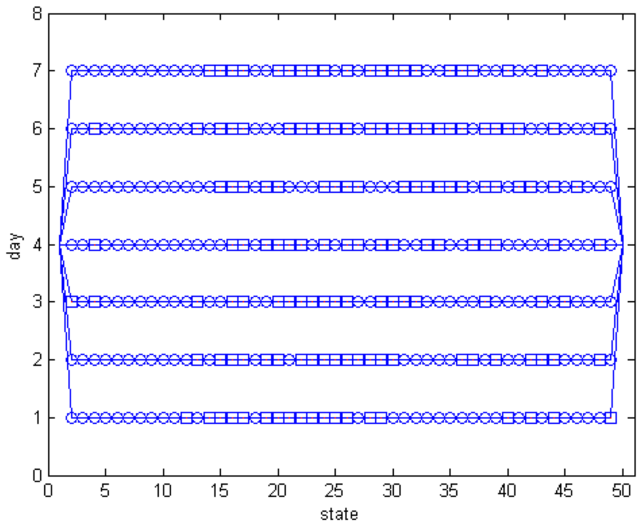


Figure 4.25: The state value of sensor 5 in the structure of the behavior model

user activities occur from daytime through to night-time, and in the behavior model there are star, square, and circle symbols to represent these activities.

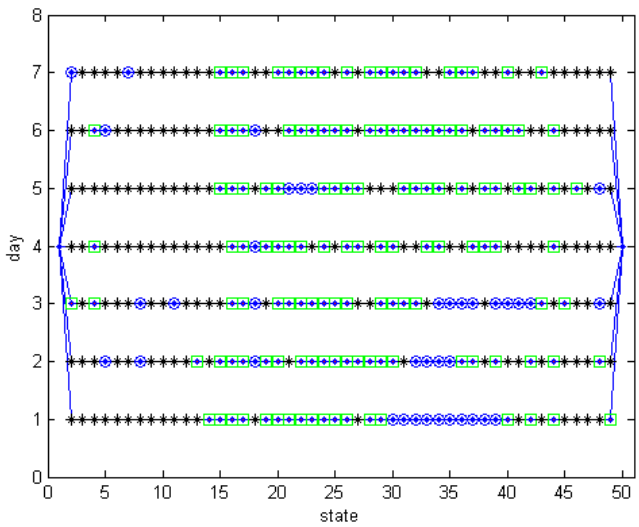


Figure 4.26: The fused state value of sensors 1, 2, 3, 4, and 5 in the structure of the behavior model

In figure 4.27 the consecutive states with bigger similarity value than the threshold value will be merged together. The state count is reduced from about 340 to 100. The state count of above figures (the behavior models with 3, 4, and 5 sensors) increased when more sensors were used in the behavior model, which meant more activity information of the user.

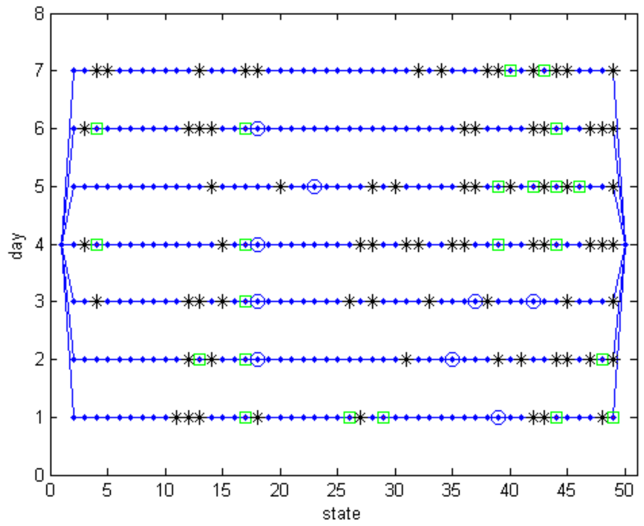


Figure 4.27: The behavior model of sensors 1, 2, 3, 4, and 5

4.4.2 Fusing the Different Types of Sensors

Above sensor fusion with the same type of sensors was introduced. In this section the data from different type of sensors will be fused together. Figure 4.28 shows that there are 4 sensors in the living room. In addition to motion detectors 1, 2, and 3 there is another type of sensor, in the form of pressure mat. The pressure mat sends data to a controller if the user stands or walks on it. In this section how to make sensor fusion with different sensor types will be introduced.

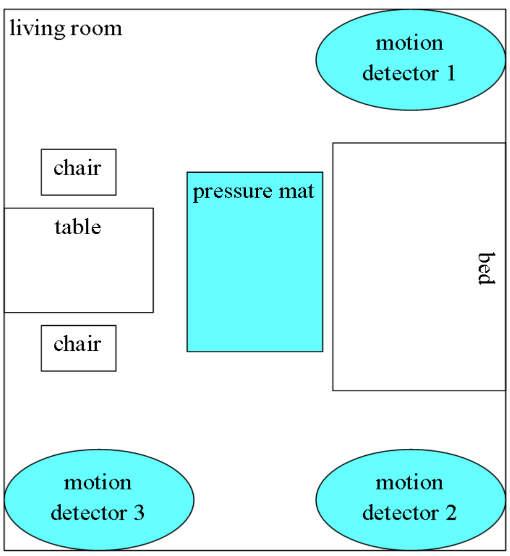


Figure 4.28: Different sensors in living room

Figure 4.29 is the behavior model from the pressure mat in the living room for 7 days. The circle there means a state without activity and the square there means a state with activity. From state

2 to state 14 each day there are more circle symbols, which mean from 0:00 to 6:30 at night there is little activity from the user at the pressure mat area. This corresponds to reality, because at night the user slept in their bed. But on day 1, 2, 3, 4, 6, and 7 there are always one or more states with a square symbol. That means the user exhibits instances of activity at night in the pressure mat area.

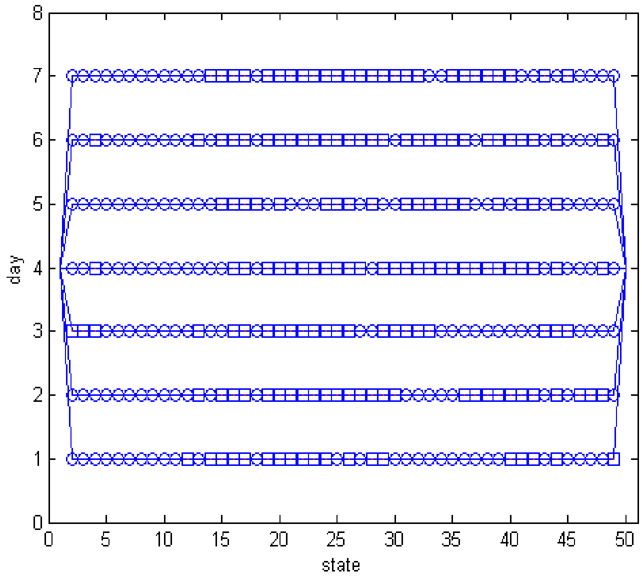


Figure 4.29: The state value of pressure mat sensor in the structure of the behavior model

From state 15 to state 44 on every day in figure 4.29 there are more square symbols, that means the user is active in the pressure mat area. In the evening from state 45 to state 49 there are more circle symbols (from 22:00 to 24:00). Perhaps the user went to bed after 22:00. The behavior model in figure 4.29 is from the pressure mat. When the user is active in the pressure mat area the motion detectors which are installed in the same living room perhaps will detect the activity too. If we fuse the data from the pressure mat and motion detectors 1, 2, 3 together, we will get a behavior model figure 4.30. Then we will compare figure 4.29 and figure 4.30 in order to see if the activity detected by the pressure may is also detected by one or more of the motion detectors.

Figure 4.30 shows the behavior model from different types of sensors (motion detectors and pressure mat in the living room). If we compare figure 4.30 and figure 4.29 we will find that from state 2 to state 14 at night on every day the motion detectors also detect the user activity which was detected by the pressure mat. For example on day 4 at state 4 there is a square symbol in figure 4.29. At the same day and the same state in figure 4.30 there is square symbol too. That means the activity from the user at night was detected by not only the pressure mat but also all three motion detectors. Another example is on day 3 at states 2, 3, and 4 there are square symbols in figure 4.29. At the same day and the same states in figure 4.30 there are square symbols too. That means the user activity has been detected by all the different types of sensors (motion detectors and pressure mat in living room). Figure 4.30 indicates that through sensor fusion (SF), there is a high degree of confidence that the activities of the user would be detected. For example, some activity detected by the pressure mat would normally be detected by the motion detectors too. But if the activities were not detected by any of the motion detectors, perhaps the pressure mat is sending incorrect signals. If an activity is detected by both the pressure mat and

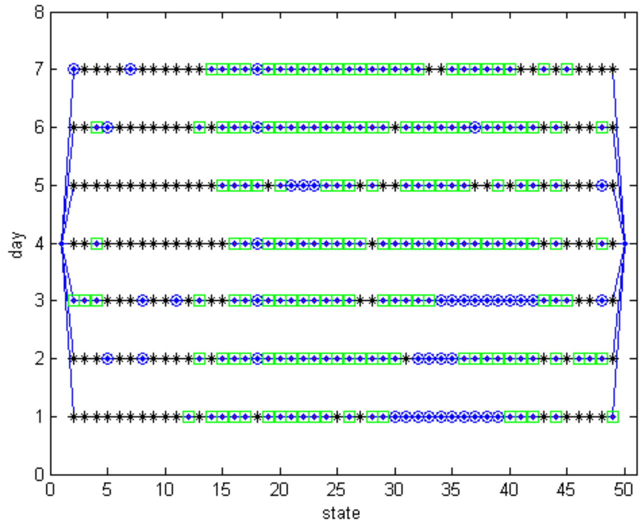


Figure 4.30: The fused state value of sensors in living room in the structure of the behavior model

motion detectors in the same state, we can say with confidence the activity actually happened.

In figure 4.31 the consecutive states with larger similarity value in figure 4.30 will be merged together. The state count is reduced from about 340 to about 80. This is a brief and clear behavior model of the user with different types of sensors.

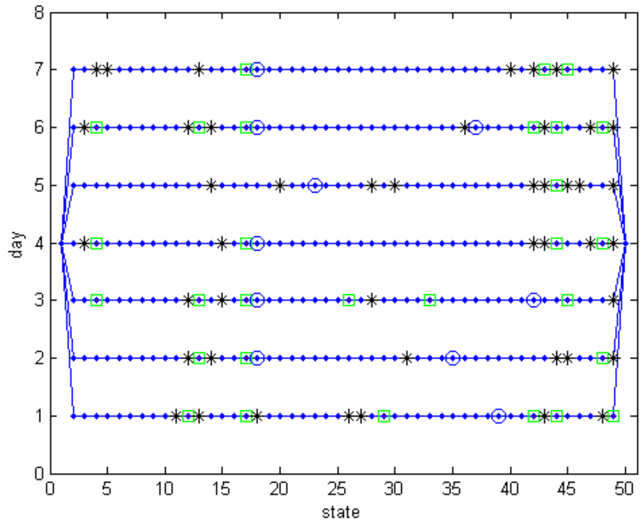


Figure 4.31: The behavior model with different types of sensors in living room

4.4.3 Fusing the Different Types of Sensors in Different Rooms

In the above sections we discussed sensor fusion (SF) with the same types of sensors in the same room (covering different rooms), and sensor fusion (SF) with different types of sensors in same room. Now a further situation will be discussed: sensor fusion (SF) with different types of sensors in different rooms. Figure 4.32 shows a whole user living environment. There is a living room, an entrance hall, and bathroom/WC area. Three motion detectors (with numbers 1, 2, and 3) and a pressure mat are installed in the living room. A motion detector (with number 4) is installed in the entrance hall. A further motion detector (with number 5) is installed in the bathroom/WC area.

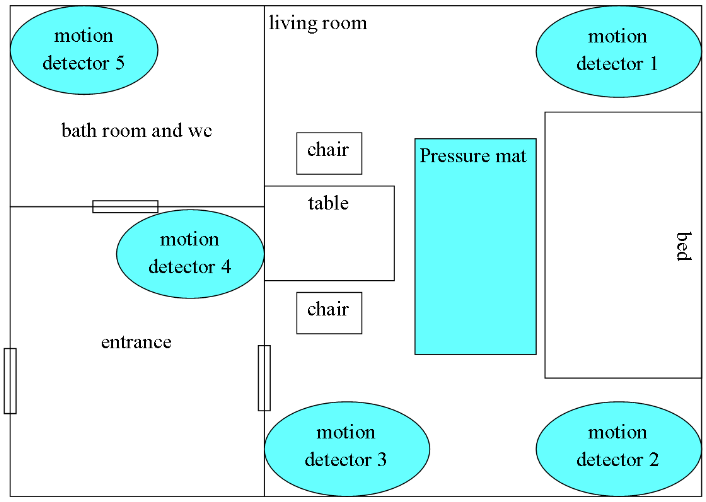


Figure 4.32: Different types of sensors in living environment

Figure 4.33 is the behavior model of the user with state data from all 6 sensors installed in the living environment. From state 2 to state 14 (from 0:00 to 6:30) there are mostly star symbols, that means the activities of the user at night are detected by some of these sensors but not all of them. But on day 1, 2, 3, 4, and 6 there are square symbols, which mean the activities of the user are detected by all 6 sensors. What happened with the user at night when all 6 sensors detected the activity? From figure 4.32 we know that the 6 sensors are installed in different rooms. When the user gets up at night, walks over the pressure mat and then from living room goes to the entrance hall, finally arriving at the bathroom/WC. In such a situation the user activity would be detected by all 6 sensors. And because each state has duration 30 minutes, so from the behavior model the system can judge that the entire user activity (the user goes to bath room and WC at night) happened within 30 minutes.

In figure 4.33 from state 15 to state 44 there are more square symbols, that means the user has activities in the whole living environment during the daytime. In the afternoon on some days the user has a rest, indicated by the fact that none of the six sensors shows any activity (for example on day 1, 2, and 3 in the afternoon from state 30 to state 42). From state 45 to state 49 there are more star symbols. Because the activity was not detected by all 6 sensors, the user activity must have been confined to a limited area.

In figure 4.34 the consecutive states with larger similarity value in figure 4.33 will be merged together. The state count is reduced from about 340 to about 90. This is a brief and clear behavior model of the user with different types of sensors.

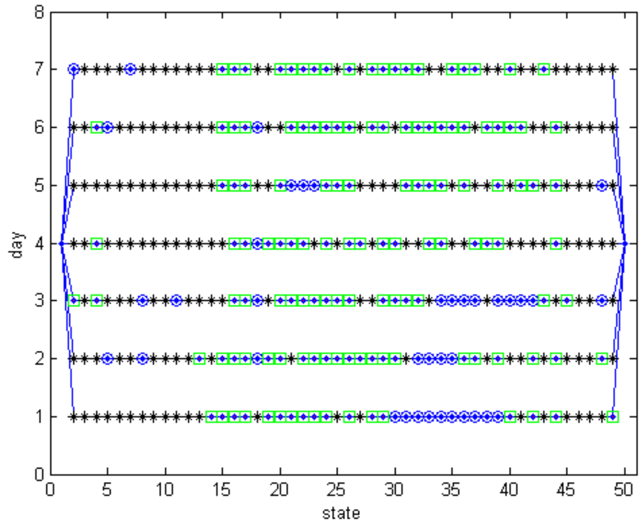


Figure 4.33: The fused state value of different types of sensors in living room in the structure of the behavior model

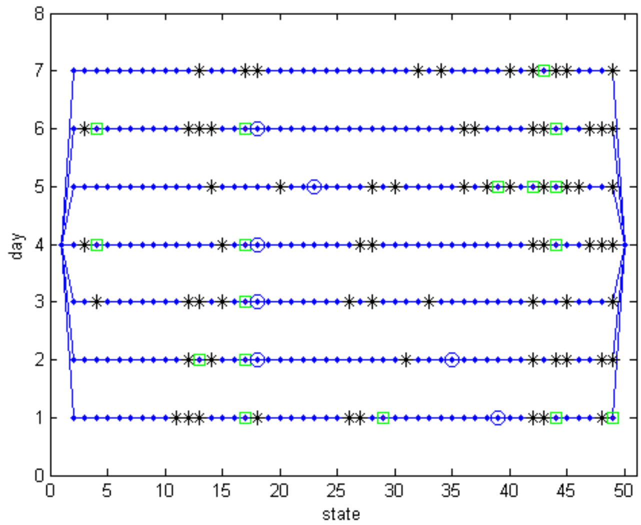


Figure 4.34: The behavior model with different types of sensors in living environment

4.5 Sensor State Data Fusion with Multi-Sensor

In this section the sensor state Data fusion with multi-sensors will be introduced and discussed. Firstly the relationship between raw data amount, state count, and similarity rate will be discussed. Secondly the relationship between similarity rate and state count will be figured out in the same time period. Lastly a behavior model from all multi-sensors in the living environment will be shown and discussed.

Figure 4.35 shows a living environment installed with different types of multi-sensors.

There are 13 multi-sensors in the living environment. Each multi-sensor is composed of some of the

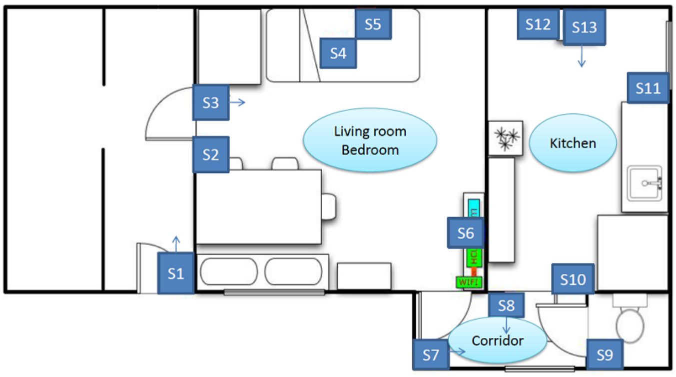


Figure 4.35: The living environment with installed multi-sensors

following sensors: door contactor, motion detector, accelerometer, light sensor, and temperature sensor. That means some multi-sensors will be composed of a door contactor and a motion detector. Others are composed of an accelerometer, a light sensor, and a temperature sensor. For example sensor 1 consists of a door contactor, a motion detector, light sensor, and temperature sensor. Sensor 9 consists of a door contactor, a light sensor, and a temperature sensor. The light sensor and temperature sensor are more influenced by the environment (time or weather) and less influenced by the activity of the user (for example the temperature in a room will not be changed immediately when the user comes into the room, and the light sensor changes its value with time and ambient light, in the morning or at night, and is not directly influenced by the user. Only when the user turns the light on or off will the value for the light sensor change; after this happens, there will be no further user related change until the user turns the light off or on again). In the thesis only the activities of the user will be observed, so the sensor values from the light and temperature sensors will not be included in the data.

The other sensors such as door contactor, motion detector, and accelerometer will send values to controllers when they detect user activity. For example a motion detector will send value 1 to the controller when it detects user activity. If the user has activity in the observed area continually the motion detector will send the second value to the controller every 3 seconds. That means the time interval between two successive signals from motion detector is 3 seconds. When no activity is detected by the motion detector no value will be sent. For the door contactor the time interval is about 100 milliseconds. For the accelerometer the time interval is about 200 milliseconds.

In chapter 3 the translation of raw data into state data was discussed. The sensor fusion between consecutive states based on sensor state value similarity rate was introduced too. In this chapter the relationship between these three parameters will be found.

Figure 4.36 shows the relationship between raw signal count, state count, and similarity rate. X-axis is the similarity rate which judges whether the consecutive states are fused together or not. The Y-axis is the state count from a behavior model with 7 days (each day has 48 states before state fusion). The lines in the figure are the relationship between similarity rate and state count. The symbol S in the figure is the signal count that translates to state. For example S bigger than 150 means that in a time interval if there are more than 150 signals sent to the controller the time interval will has a state value 1.

Figure 4.36 shows that the relationship between similarity rate and state count changes with raw signal count. If the signal count is larger than 150 in one state and the similarity changes from 0.1

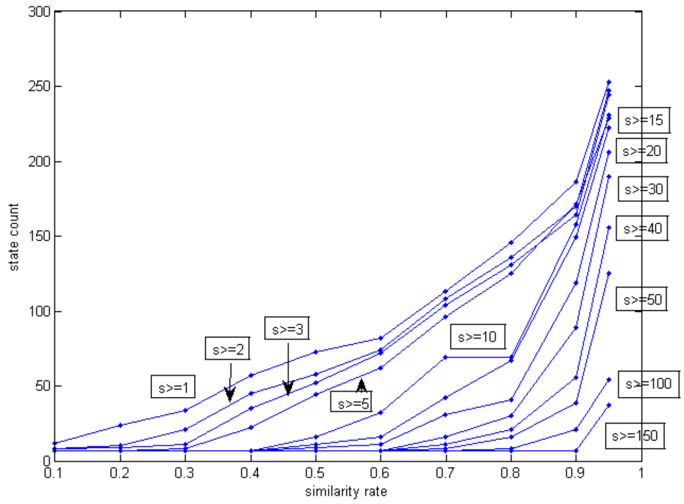


Figure 4.36: The relationship between raw signal count, state count, and similarity rate

to 0.9 the state count in the whole 7 days stays the same. When the signal count is larger than 10 in one state the state count increases with changed similarity rate. When the signal count is larger than 3 or 5 in one state the relationship between similarity rate (from 0.4 to 0.6) and state count are nearly linear. With the figure the optimal parameters will be chosen. In the thesis, for multi-sensor data the similarity rate between consecutive states was chosen as 0.5 (or 0.6). The signal count in one state was chosen as larger than 3 (or 5).

Figure 4.37 shows the relationship between similarity rate and the state count based on a signal count larger than 3. The relationship searched with different signal count. All the 4 blue lines are similar. The green line is the mean value of all 4 blue lines. The relationship between similarity rate and state count is nearly linear when the similarity rate is chosen between 0.4 and 0.6.

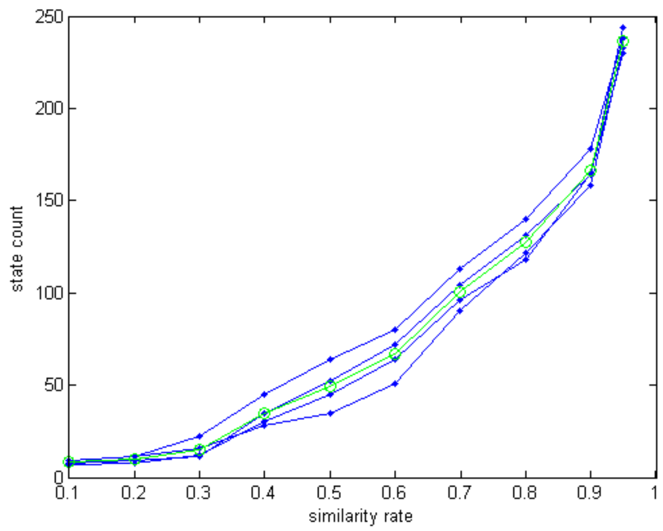


Figure 4.37: The relationship between similarity rate to state count

Figure 4.38 is the behavior model with all 13 sensors from figure 4.35. X-axis is the state count. There are 48 states on each day. Y-axis is the day count. There are 7 days in the model. Generally there should be about 340 states in the model. After state fusion there are about 50 states left. Many states are fused together in the model, especially from state 1 to state 15 on each day. That means the user has similar activities at night. From state 20 to state 30 there are many states which are fused again. In the evening before state 48 there are many consecutive states which are fused again. That means the user has similar activities in the evening. Only on the second day is there a state with value 0. That means there none of the sensors detected any activity within this state. This situation is rare. There is only one such state in all 7 days. In the next chapter the behavior model will be connected with the hidden Markov model to get the parameters about normal activity. Based on the parameters any unusual activities will be detected.

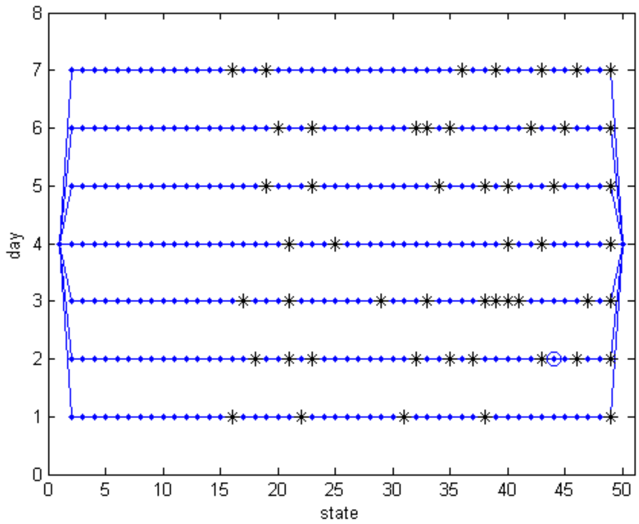


Figure 4.38: The behavior model with all sensors after sensor fusion

4.6 Multi-Sensor Data Correlation

The activities of the user changed from time to time in different locations, the sensors observed the activities, so the correlation between different sensors indicated the activities. For example at night the whole living environment should be very quiet and there should be perhaps only a small amount of activity in the bedroom of the user, but if the user gets up at night and goes to the WC, more activities should be detected by many sensors in bedroom, corridor, and WC. In the morning the user goes to kitchen and ready for breakfast, so there activities should be detected by sensors which are installed in the kitchen. Or in another situation if the user watches TV in the living room in the evening, so the sensors in the living room should be activated and sent signals to the server.

Because the activities from the user changed and occurred in different locations from time to time, in order to detect when the activity happened a predefined time interval (T_{int}) is necessary. In the following, T_{int} is defined as 15 minutes. This value is just an example; based on the real situation, the parameter could have a larger or smaller value.

There are in total 14 multi-sensors in the living environment. First an example will be given for sensor data correlation with two sensors A and B. In time interval T_{win} sensor A gathered values from multi-sensor $A = [a_1, a_2, \ldots a_t]$; sensor B gathered values from multi-sensor $B = [b_1, b_2, \ldots b_t]$; In order to synchronize these values from different sensors in different time points, a time window will be used. The small time window has a time interval (T_{win}) of a few seconds. Within the time interval the values sent to the server have the same time point. These values from both sensors comprise a pair $[a_t, b_t]$. With sensor data correlation the relationship between sensors A and B will be found.

The variance of A and B

$$Var(A) = E[(A - \mu_a)]; Var(B) = E[(B - \mu_b)]; \tag{4.5}$$

Here μ_a and μ_b are the mean value of A and B.

The covariance of A and B

$$Cov(A, B) = E[(A - \mu_a)(B - \mu_b)] \tag{4.6}$$

The correlation coefficient between A and B

$$R_{ab} = \frac{Cov(A, B)}{\sqrt{Var(a)Var(B)}} \tag{4.7}$$

The correlation coefficient interval

$$|R_{ab}| <= 1 \tag{4.8}$$

If values from A and B are independent

$$R_{ab} = 0, if Cov(A, B) = 0 \tag{4.9}$$

If values from A and B are identical

$$R_{ab} = 1, if Cov(A, B) = \sqrt{Var(a)Var(B)} \tag{4.10}$$

Above are the formulas describing how data correlation works between values from 2 sensors. But if there are more than 2 sensors, so the data correlation has to be done between each of them. That means for example sensor A correlated with sensor B, then with sensor C ...; sensor B correlated with sensor C, then with sensor D ...; sensor C correlated with sensor D, then with sensor E ... till all sensors are correlated with each other. In such way the relationship between all the sensors will be found. In the following paragraph the sensor data correlation result will be shown and discussed.

Figure 4.39 displays the sensors' data correlation result from 3:45 to 4:00 at night in living environment of an elderly person. There are in total 14 multi-sensors but in the time interval (T_{int} = 15 minutes) only 4 sensors have signals and they will be correlated. Sensor 12 is installed in the bedroom, it was correlated with sensor 11 which is in corridor. The correlation value was 0.6. In addition sensor 12 was correlated with sensor 7 (in WC) and sensor 1 (at entrance). The correlation values were 0.4 and 0.5 respectively. Sensor 11 was correlated with sensor 7 and 1, with values 0.4 and 0.5. Sensor 7 was correlated with sensor 1, giving a value of 0.3.

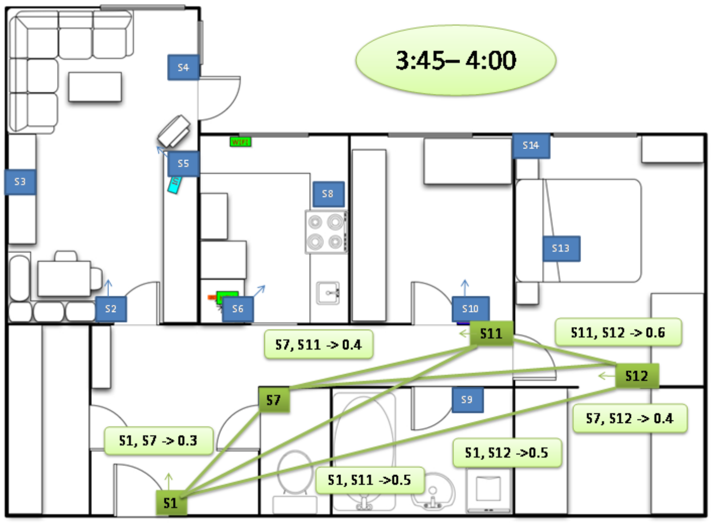


Figure 4.39: Sensor data correlation result from 3:45 to 4:00

From the result we can see that the user has activities in the bedroom, corridor, entrance area, and WC between 3:45 and 4:00 at night. A likely scenario is that the user gets up in bedroom, passes through the corridor, and goes to the WC. Furthermore the correlation value illustrates that sensors 12 and 11 have a stronger relationship than sensor 12 has with sensor 7, or sensor 12 with sensor 1. This is because there are motion detectors in sensors 12, 11, and 1 (indicated with arrow). A motion detector has a bigger detection area than a door contactor or vibration sensor. If any activity occurs in the area between sensors 12 and 11, they will both detect it. So they have a bigger correlation value (0.6) than sensor 12 and sensor 7 (0.4). On the other hand sensor 11 is nearer to sensor 12 than sensor 1, so the correlation values between sensors 12 and 11 are bigger than sensor 12 and sensor 1 (0.5). This is the reason at the same time why sensor 12 is nearer to sensor 7 than sensor 12 is to sensor 1 but the correlation value between sensor 12 and sensor 1 is larger than the correlation value between sensor 12 and 7.

The sensor data correlation result shown in figure 4.40 is in another time interval. In figure 4.40 the user has activities between 17:15 to 17:30. The activities are detected by sensors 12 and 11. The activities detected by sensors from 1 to 5 which are installed in the living room and entrance. Again here the correlation value between sensor 2 and 5 are the biggest of all of the correlation values. This is because they are located nearer to each other, have nearly the same detection area, and both of them have motion detectors.

Another interesting thing is that if we compare the result from figure 4.39 and figure 4.40 related with sensor 12 and 11. They have different correlation values with 0.6 and 0.4. The location of the both sensors are not changed but why the value? This is because of the activities of the user and the detection area. If the activities happened mostly at the detection area common to both sensors, they should have bigger correlation value but if the activities happened mostly in the detection area of only one of the sensors, the correlation value should be smaller. At night in figure 4.39 the user only goes through the corridor, and the activity is mostly detected by both sensors. But in figure 4.40 the activities happen mostly in the living room or between the entrance and living room, so sensor 11 can detect it better than sensor 12. The differing areas involved in different activities account for the different correlation values between the same sensors.

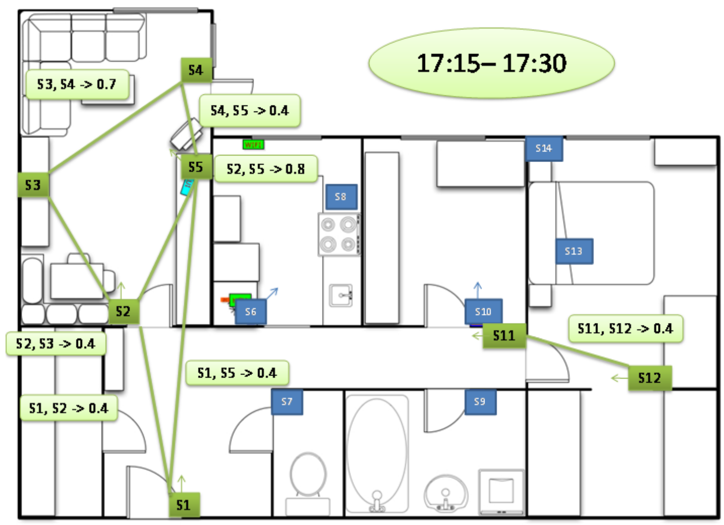


Figure 4.40: Sensor data correlation result from 17:15 to 17:30

The above two figures illustrate that through sensor data correlation the relationship between sensors may be detected; furthermore when and where most of the activities of the user happen were also shown. That means the relationship between sensors indicates the activity topology of the user.

4.7 Result and Discussion

In this chapter, sensor fusion was introduced and discussed. Data from different single sensors were used to learn behavior models. Because these learnt models have different structures, it is difficult to fuse them together. A novel method was introduced to mitigate this problem: instead of merging the different structures, the sensor state data was merged together first and the behavior model was learnt from this merged sensor state data. Furthermore, the consecutive states were either fused together or not based on their degree of similarity and a predefined threshold value. In order to find the optimal threshold value, the real data as well as designed data were used. The relationship between similarity value, threshold value T_h for state fusion, and signal count was explored and discussed. The relationship between $T_{interval}$ of each state, the threshold value T_{th} for signals translated to this state, and the state count of the behavior model was also examined. The novel sensor fusion method was used to fuse measurements from different kinds of sensors in different rooms within the living environment in order to explore changes in the behavior model when using multiple sensors. The behavior model obtained using fusion data from all sensors shows the behavior of the user in the entire living environment during the whole day.

In order to understand the relationship between different kinds of sensors, data correlation is needed. The relationship between sensors indicates not only the behavior of the user but also the behavior topology of the user.

Up to now 2 different kinds of behavior models have been learnt. One behavior model was for regular behavior (such as the user taking medicine tablets several times each day) and the second model was for random behavior (such as the user's daily routines). In the next chapter these 2 models will be used to realize automatic scenario detection. For regular behavior, the mean values and standard deviation values from the regular behavior model will be used and for random behavior, the model introduced in this chapter will be used. However, due to the fact that the models are only abstractions of user behavior, automatic scenario detection cannot be realized, and unusual behavior cannot be detected, based only on the learnt models. The models themselves do not have the ability to judge whether a detected behavior is normal or unusual. Other methods must be used for this based on the learnt models, and these methods will be introduced and discussed in the next chapter.

5 Automatic Scenario Detection and Case Study

In this chapter the two behavior models introduced previously (for regular behavior and for random behavior) will be used to realize automatic scenario detection. The behavior model for regular behavior (learnt using the Gaussian mixture model and split-merge algorithm) will be utilized first. Based on the mean value and standard time deviation in each cluster, the incorrect instances of behavior will be detected. The behavior model for random behavior (learnt using sensor fusion and a hidden Markov model) will be used to analyze the daily routines of the user. A novel method will be introduced to find the top and bottom value boundaries of the behavior model. Automatic scenario detection will be realized based on the parameters of the hidden Markov model, the forward algorithm and the value boundaries. Furthermore, the learnt behavior model will be used to determine which daily routine in the model has the most similarity to the test daily activity routine. The behavior model will then be shown to be tested with regard to its ability of detecting unusual scenarios in the test daily routine using real data. The final section includes the results and their discussion.

5.1 Analyzing Behavior Model with Split-Merge Algorithm

In chapter 3 the split-merge algorithm was used to learn the behavior model of the user for activities which occurred regularly. One example concerned a regular mealtime. The user got a meal from the caregiver every day at nearly the same time. A data set included data gathered throughout one month. With the data a behavior model was learnt. In the model there are 3 clusters and each of them has a mean value and a standard time deviation.

In this section an approach will be introduced and discussed which uses the model mean value and time deviation to detect wrong activity. The approach postulates that if an activity happens outside the maximum time deviation from the model mean value, then the activity is considered to be wrong.

Figure 5.1 is the example from chapter 3. There are 3 clusters (indicated with 3 green bell curves, these curves are the learnt regular behavior model of the user. The model indicates when the user gets meal with what kind of time deviation). X-axis is the daily time from 0 to 24:00; Y-axis is the count about the activities that the caregiver sends meal to the user (the user gets meal) and at the same time Y-axis indicates the probability density of the green bell curves. The parameters such as mean values of these clusters, the probabilities that at these clusters the user get meal, and the standard deviations for each cluster are shown in the figure. The data from the example was gathered in a continuous period of about one month. The model shows that the user gets a meal very regularly: in the morning and at noon. In the afternoon few seldom signals appear in the model and in the evening there are no signals. That means the user does not get a meal in the evening. The signals from the afternoon are rare and are not included in the model.

The mean value and standard deviation of each cluster build boundaries for each cluster (mean value ± standard deviation). If the time of the activity is outside the boundary the activity will be treated as a wrong activity. For example in the morning a cluster has time mean value 7.7 and standard deviation 0.4. That means the user gets meal from caregiver sometime between 7.3 (7:18) and 8.1 (8:06). If one day the user gets no meal after 8.1 (8:06) in the morning, a reminding signal should be sent to the caregiver. On the other hand, the boundary can be made a little wider to adapt to the real situation (add the left two clusters together, so the boundary will now be between 5.5 (05:30) and 8.3 (08:18)). For example one day the caregiver sends the meal to the user a little earlier or later. Applying the cluster mean and standard deviation value, the user should have got their noon meal between 11.7 (11:42) and 12.7 (12:42). The boundaries can be changed to a little wider in order to adapt the real situation (perhaps the caregiver sent the meal to the user a little earlier or later one day) for example from 11.4 (11:24) to 13 (13:00) based on the model. If no meal is sent to user within these boundaries, a reminding signal should be sent to the caregiver.

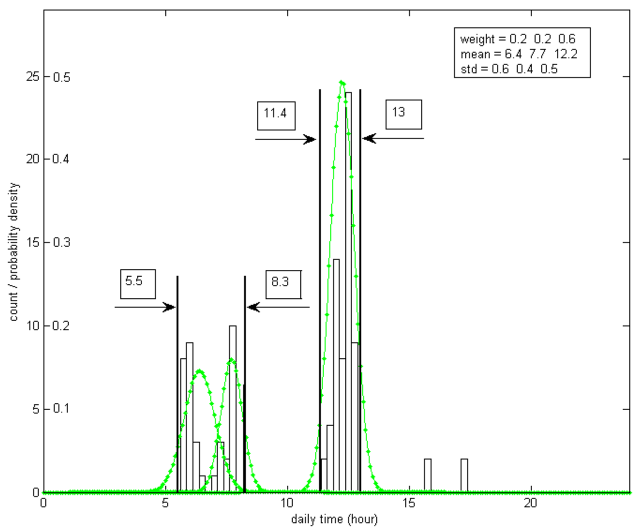


Figure 5.1: To find the activity boundary based on the learning result

On the left side in figure 5.1 there is a cluster with a standard time deviation about 0.6. This is a relatively large deviation value. That means the user does not get their meal in the time cluster very regularly. Or perhaps the activity may actually be the caregiver collecting the tableware from the day before. For this kind of activity, which has a large time deviation, the above method is not appropriate. But in the real world there are many situations which happen irregularly. In such situations a hidden Markov model should be used. In the following, this approach will be introduced and discussed. This is the main work of the thesis.

5.2 Behavior Model with Single Sensor and Sensor Fusion Data

In this section the behavior model from a single motion detector sensor will be learnt. Then the state data from single sensors will be fused together to learn the behavior model. From the behavior model the daily activity routine of the user will be indicated. The living environment of the user and the installation location of sensors is shown in the following figure 4.35. In figure 4.35 there are 13 multi-sensors. These sensors composed with motion detector, door contactor, accelerometer, light sensor, and temperature sensor. The sensors 1, 3, 7, 8, and 13 contain motion detector.

5.2.1 Analyzing Behavior Model with Single Motion Detector Data

In order to research the activity of the user the multi-sensors which contain motion detector will be chosen. Each sensor has a limited observation area, and the user activity in different locations will vary. The user will carry out different activities at different locations at different times. So the behavior model of each of these sensors should be different. In this section the difference of these behavior models will be researched and discussed.

Figure 5.2 is the state value in the structure of the behavior model from motion detector sensor 3 in the living room/bedroom area. The circle symbol means there is no activity detected by the sensor. The symbol square indicates that the sensor has detected activity. Each symbol has a time interval of 30 minutes. There are 48 states in the whole day. In the model there are 7 days in total. The figure shows that the user has much activity in the daytime from state 15 to state 45 every day. But at night from state 1 to state 14 there is a lot of activity too. Is it the real situation or is something wrong with motion detector 3? We cannot get the answer from sensor 3 alone. However there are other motion detectors in the living environment. The real situation may be found through the behavior models from the other sensors.

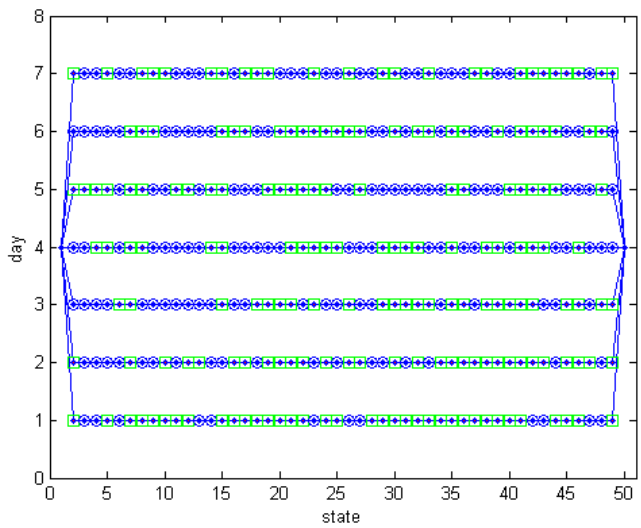


Figure 5.2: The state value in the structure of behavior model from motion detector in living and bedroom

Figure 5.3 is the state value in the structure of behavior model from motion detector sensor 7 in the corridor. The behavior model shows that, from state 1 to state 15 in all 7 days , there is little user activity in this area. Activity was detected by motion detector 7 only on the second day at state 6. This indicates that the user carried out some kind of activity in the corridor area that night at about 3:00 am. But on other days at night there was no activity in that area.

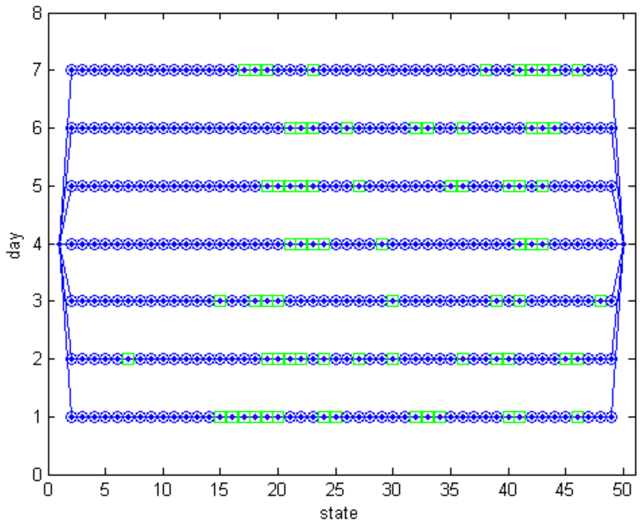


Figure 5.3: The state value in the structure of behavior model from motion detector in corridor

Figure 5.4 is the state value in the structure of behavior model from motion detector sensor 8 in the corridor near the WC. The model is very similar to the figure 5.3. This is because the both sensors have nearly the same observation area. When the user is active at the observation area common to both sensors, then both sensors will detect it. Additionally, on the second day at state 6, there is also a square symbol. That means the user has been active in that area at night and the activity was detected by both of the sensors.

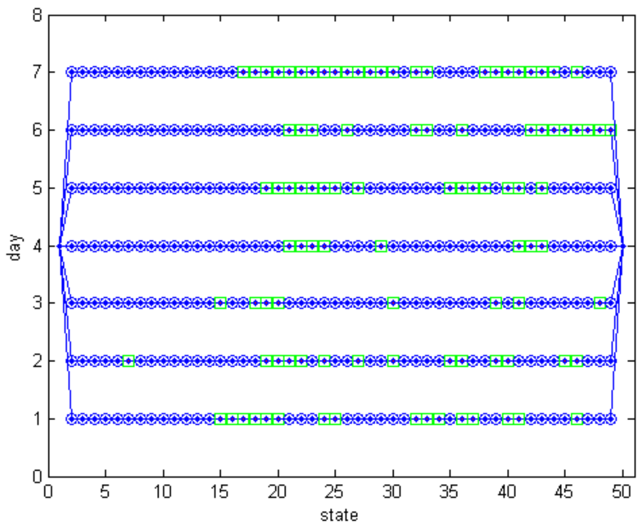


Figure 5.4: The state value in the structure of behavior model from motion detector in corridor near WC

Figure 5.5 (figure 5.6) is the state value in the structure of the behavior model from motion detector sensor 1 at entrance (sensor 13 in kitchen). The model indicated there was no user activity at night from state 1 to state 15 for all 7 days at entrance (kitchen).

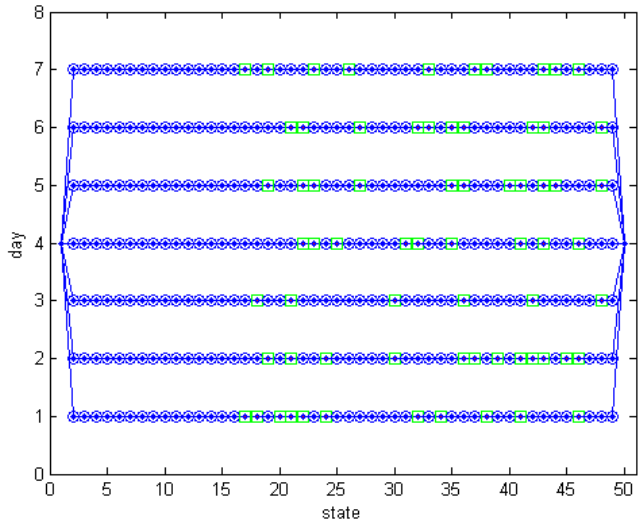


Figure 5.5: The state value in the structure of behavior model from motion detector at entrance

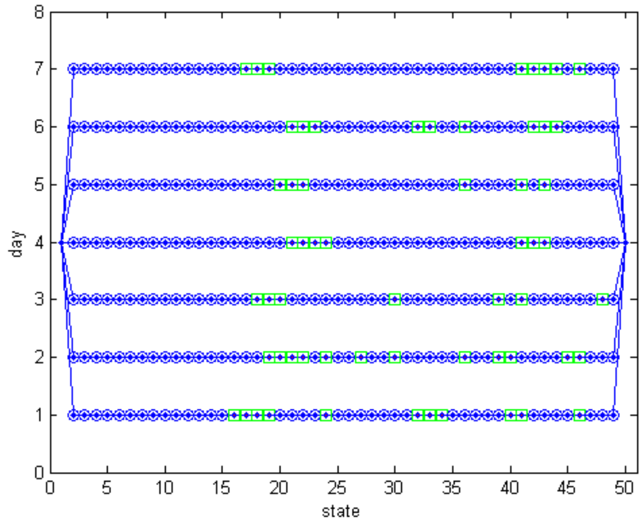


Figure 5.6: The state value in the structure of behavior model from motion detector in kitchen

All the above 5 behavior models indicate that the user shows activity at night only in the living room/bedroom area. On just one night the user perhaps went to the WC about 3:00. This activity was detected by sensors 7 and 8 that were installed in the corridor near the WC. Through comparing the state values of these models from different sensors (which installed in different rooms) the user behavior should be detected with more detail. For example the user had not activity at entrance and kitchen but in the living room and bedroom at night. But the relationship between the behavior models from single sensors cannot be directly shown. In the following section these models will be fused together in order to show the activity behavior of the user in the whole living environment.

5.2.2 Analyzing Behavior Model with Motion Detectors Data

Figure 5.7 is the fused state value in the structure of behavior model from all 5 motion detectors in living environment. The model shows that at night from state 1 to state 15 there are more circle symbols. That means the user has very little activity at night. The star symbol means that, within that state, some sensors detected user activity, while some did not. There are some states with square symbols. These mean that the user had activity in the whole living environment and the activity was detected by all motion detectors. After state 45 there are many stars and circle symbols; again that means the user has activity only in some areas, or just rests quietly in the living environment. From the behavior model the general activity of the user will be shown, but the activity information from individual sensors cannot be shown in the states with star symbols.

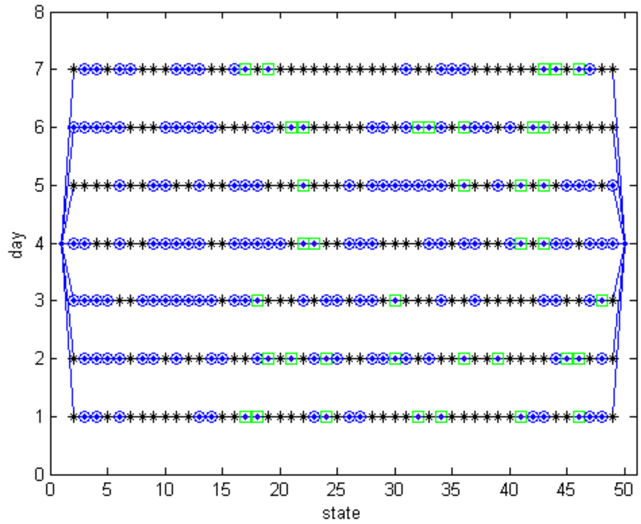


Figure 5.7: The fused state value in the structure of behavior model from all motion detectors in living environment

5.2.3 Analyzing Behavior Model with Multi-Sensors Data

Figure 5.8 is the fused state value in the structure of the behavior model from all sensors in the living environment. In the model there are only star and circle symbols. That means the user did not have much activity in the whole living environment in each state. On the other hand, in the afternoon and the evening there are several circles. Thise means there was almost no user activity in any part of the living environment. Most of the states in the model are star symbols, which mean the user has activity within some areas. The activity was detected by some sensors and the sensors in other locations detected no activity, so the result is a mixed activity situation with star symbols.

Figure 5.9 is the behavior model after consecutive state fusion. In chapter 4 the sensor fusion methods were introduced and discussed. After sensor state fusion the state count is reduced from about 340 to about 50. The similar states would be fused together. The model showed that at night the user has similar activity and these states at night would be fused together. In the morning there are some different activities and at noon the activities are similar again. In the

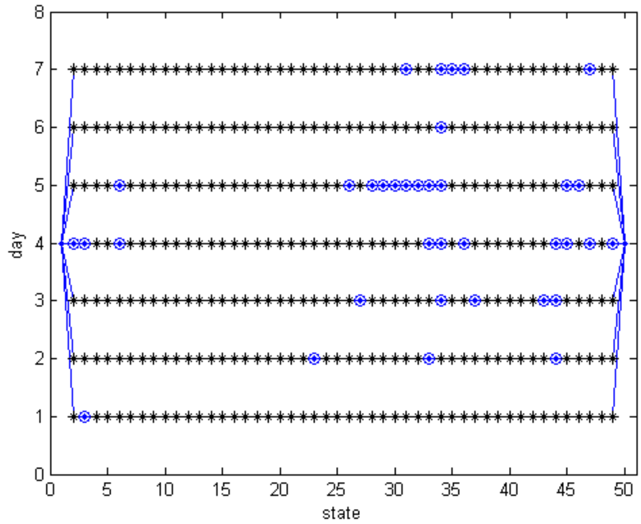


Figure 5.8: The fused state value in the structure of behavior model from all sensors in living environment

afternoon there are some different activities and in the evening to night the activities are similar again. From the model the general activity information of the user is shown. There are always some sensors which detected activities and other sensors which did not. In the following section a hidden Markov model will be used to analyze the behavior model from a single sensor, and fusion data from multiple sensors.

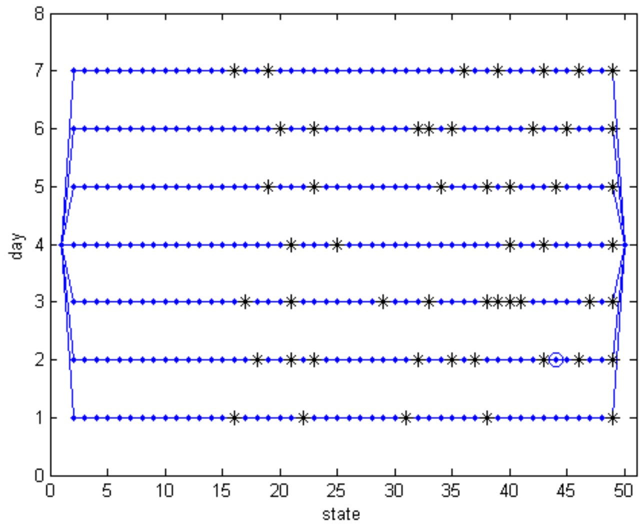


Figure 5.9: The behavior model with all sensors after state fusion

5.3 User Daily Routine Analyses with Hidden Markov Model

In the previous section the behavior models from single sensor and sensor fusion data were introduced and presented. In this section these models will be analyzed and compared in order to find out the advantage and disadvantage of these models.

5.3.1 User Daily Routine Analyses with Single Motion Detector Data

In a behavior model there are 7 daily routines such as the behavior model showed in figure 5.2. Here 7 daily routines were chosen in the model just for explaining how the model works. In the real situation the behavior model should include data covering a longer time period such as one month or two months. With a longer time period there will be more different types of daily routines included in the model and these will make the model more complete. The time period should be longer in order to gather more different types of daily routines, especially when a user does not have a stable lifestyle.

From the routines in the behavior model we obtain the basic parameters of a hidden Markov model such as the number of states, the number of distinct output observation symbols in each state, the transition probability distribution matrix, the state emission probability distribution matrix and the initial state distribution. This approach has been introduced in chapter 3. These parameters defined a hidden Markov model. Then a forward algorithm will be used to get the probability of an observation daily routine by comparing it with the model.

The method is that at first a daily routine from the behavior model will be chosen. Then the routine will be used to match each of the routines of the hidden Markov model and get the probabilities. For example there are 7 routines in the behavior model; one routine from the 7 routines will be chosen as observation routine. Then the routine will be used to match the 7 routines in the model and get 7 different probability value sets. The best and worst sets will be chosen as a matching result. Each routine in the model will be chosen as the observation routine to match the model, so there should be other best and worst routines as a result. After all the 7 routines are chosen and matched with the model there should be 7 best and 7 worst value sets.

Figure 5.10 indicates the method that got the best and the worst matching results from the behavior model (daily activity routines). The left side of figure is the model with 7 daily activity routines. On top of the figure is the first routine which was chosen from the model. It was used to match all the routines in the model and got 7 matching results. The 7 matching results are shown in the middle of the figure. One best and one worst result were chosen from the 7 matching results. Then each of the routines in the model was chosen to match the 7 routines in the model. So there are 7 groups of the best and the worst results.

Figure 5.11 indicates the 7 groups of the best and the worst results which yield the top and the bottom boundary of the matching results. On the left side of the figure are the 7 different groups of the best and the worst matching results. In the middle of the figure are all the 14 matching results (7 were the best results and 7 were the worst results). On the right top side of the figure is the best matching result from the 7 best matching results (top boundary). On the right bottom side of the figure is the worst matching result from the 7 worst matching results (bottom boundary).

In the thesis the above introduced method was used to get the top and bottom boundaries. In the above figures the number of routines was 7. The number 7 was just an example. The number 7 could be changed, dependent on the count of the routines in the model.

Figure 5.12 shows the test result with the above introduced method. X-axis is the state count. There are 48 states (each state has time duration 30 minutes). Y-axis is the log-probability

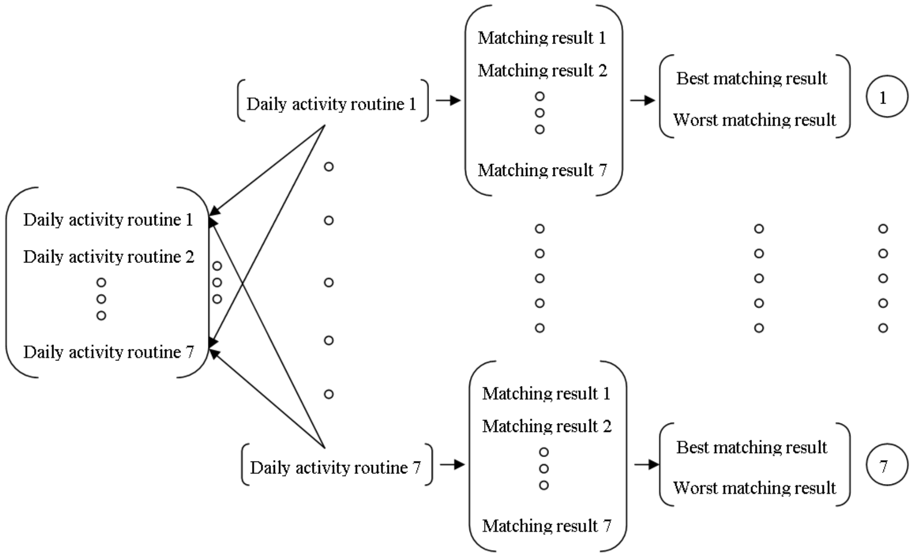


Figure 5.10: Getting the best and worst matching results from behavior model

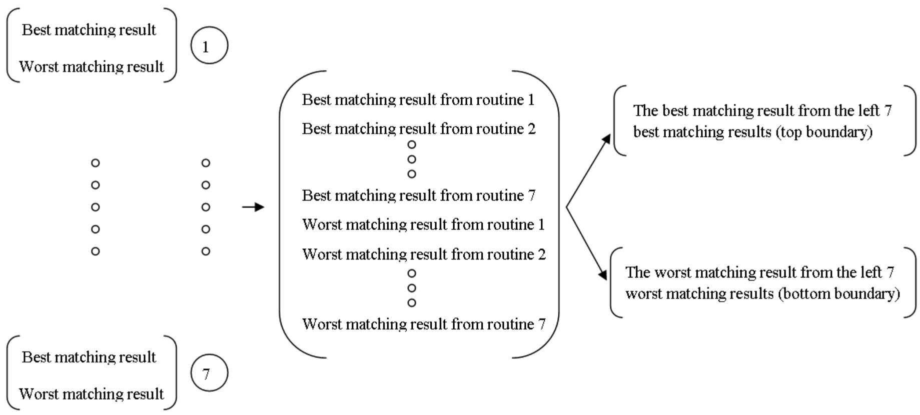


Figure 5.11: Getting the top and the bottom boundaries from the groups of the best and the worst results

(log-probability is the logarithm of the probability, because the probability has a value interval from zero to one, so the log-probability is smaller or equal to zero). The green lines with small green points are the best matching result for each daily routine matches to the model. The blue lines with small stars are the worst matching results for each daily routine matches to the model. With the best of the 7 green lines and the worst of the 7 blue lines, we can learn the matching boundaries of the model. The lines with red circles indicate the boundaries of the matching results.

If an observation routine matched with the hidden Markov model and the result is inside of the boundary that means the daily routine is normal. But if the test result is outside of the boundary that means the activity of the user is unusual. In the following some daily routines from other

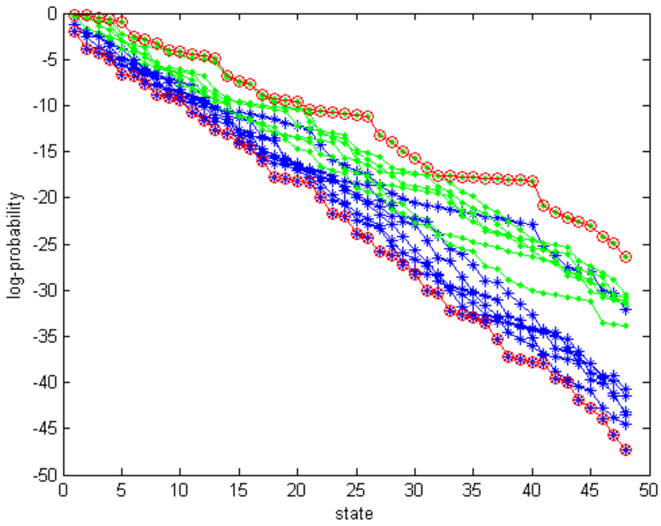


Figure 5.12: The log-probability of daily routine in living and bedroom with boundary values (red circles), best values (green lines), and worst values (blue stars)

days (different from the 7 days which were used for model) will be chosen to match with the hidden Markov model.

Figure 5.13 shows the matching result with another 7 daily routines outside of the behavior model. There is only one line outside of the boundary in some states and they are very near. So the routine can be treated as a normal activity routine. How to judge whether a routine whose states are outside of the boundary is normal or not will be discussed in the following case study section. Generally the decision is made by looking at how many consecutive states are outside of the boundary, and the value distance between the states and the boundary.

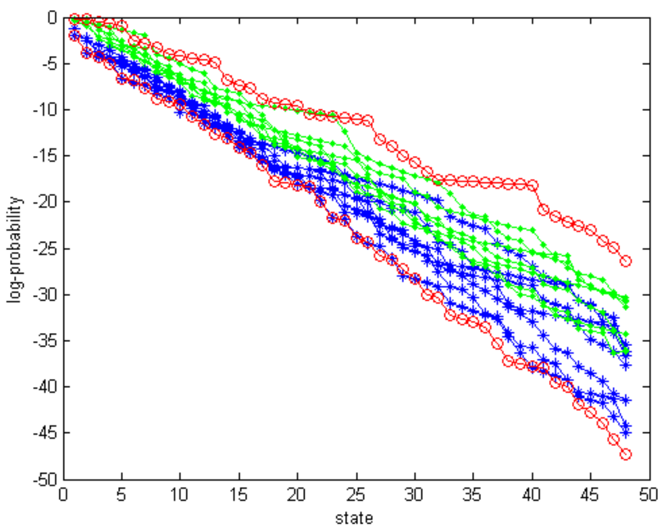


Figure 5.13: The log-probability of daily routine outside of behavior model in living and bedroom with boundary values (red circles), best values (green lines), and worst values (blue stars)

Figure 5.14 shows the test result with behavior model from sensor at the entrance. The best values keep to the 20th states nearly unchanged. The worst values keep changing linearly until to the 15th state. The matching result corresponds to the state values shown in figure 5.5. In the figure, no activity is detected by the motion detector at entrance from states 1 to 15. All the states have state value 0. The consecutive states have the same value that makes the matching value change linearly. The test states stay at the same model state and have the same self transition value. The self transition is smaller than 1 but it keeps at the same value at same state and it makes the test value change linearly. Then after the 20th state the test value becomes smaller. This is because in figure 5.5 there are some consecutive states after the 20th state for which the state value is changed. The test states match the states in the model and get different transition, self transition, and emission values. These changed values are used in the forward algorithm and make the transition probability change state by state.

If the observation routine has the same state values as the model routine then the observation routine matches the model and it reaches the last state of the model. If the observation routine has different state values from the model routine then three different situations will be possible. First the last observation routine cannot reach the last state of the model. That means the observation routine stays at some states before the last state of the model. Second the observation routine does not go to the last state but the last state of the model is already reached. The third situation is that the last state of the observation routine reaches the last state of the model but the test value is smaller than the best matched observation routine. This is because the best matched observation routine has more self transition and emission value than the worst matched observation routine. Based on the above analyzing the hidden Markov model can be used to test different observation routines and detect which ones are more matched to the model and which ones are not. Furthermore if an observation routine cannot reach the last state of the model that means the observation routine does not match the model.

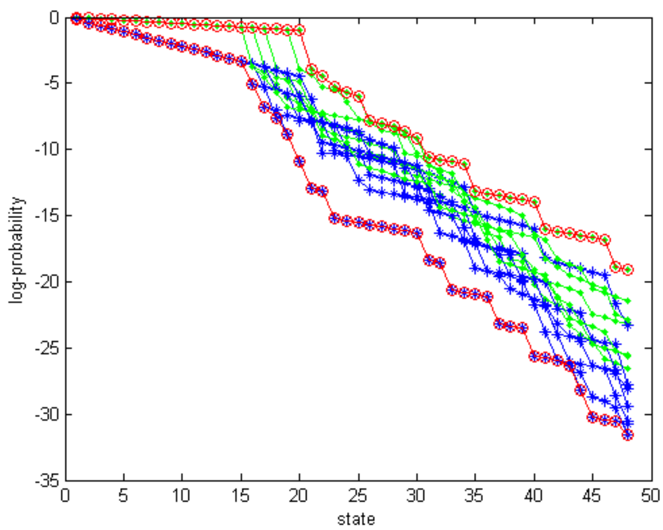


Figure 5.14: The log-probability of daily routine at entrance with boundary values (red circles), best values (green lines), and worst values (blue stars)

Figure 5.15 shows the test result with observation routines from 7 days that are different from the days used in the behavior model. Different routines have different state values and the different state values make the test result perhaps outside of the boundary of the model. On left side of figure 5.15 there is one best routine outside of the bottom boundary. That means one observation routine shows a large difference from the routines in the model. On the other side there is another routine with worst values outside of the boundary. Because each observation has two results so perhaps the two routines which are outside of the boundary have come from the same observation routine. The behavior model showed there is no activity at night and the test result shows large differences from the model from state 1 to about state 20. That means the observation routines have activities at night and the different state values (comparing to the behavior model) caused the test results which were outside of the boundary.

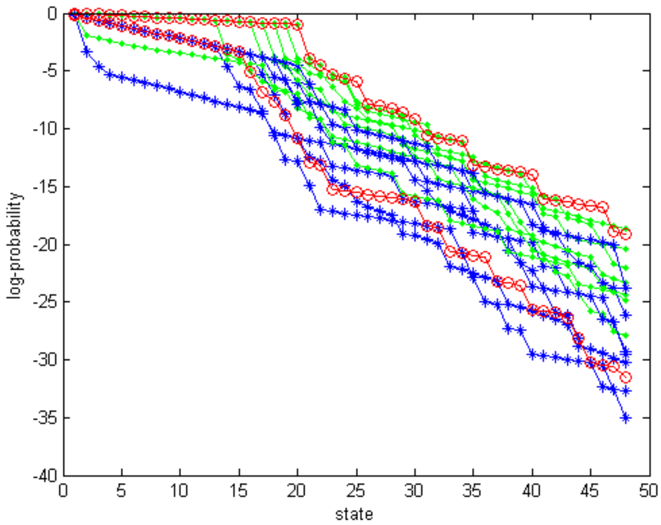


Figure 5.15: The log-probability of daily routine outside of behavior model at entrance with boundary values (red circles), best values (green lines), and worst values (blue stars)

Figure 5.16 shows the test result with behavior model from sensor in the corridor near the WC. The greatly different value between best and worst routines indicate that the activity of the user in the area has greatly different. The best value routine comes from the same routine when it is used as an observation routine and tested with the hidden Markov model. The worst value routine comes from a routine when it matches with another routine which has a large difference.

Figure 5.17 shows the test result with observation routines which came from some other days different from the days used for the behavior model. There are 4 routines outside of the boundary between state 1 to around state 20. That means at least 2 routines (7 routines in total were chosen as observation routines; each routine has one best and one worst matching result) have large differences from the model. In figure 5.4 from state 1 to state 15 there user activity in the area (corridor near WC) is seldom detected, but on the observation days there is user activity in that area at night. The differences in activity mean that the test values (from the above 4 routines) are outside of the boundary. The result proved again that it is possible, based on the hidden Markov model, to judge if another activity routine is normal or unusual.

Single sensor measurement has the following disadvantages: limited spatial coverage, sensor defection causes information losing of observed objects [Elm01, p. 4–5]. Because of these disadvantages, using a single motion detector to detect unusual activity has shortcomings: firstly, a single motion detector has a limited observation area. When unusual activity happens within another area, the activity will not be detected by the motion detector. In order to get complete obser-

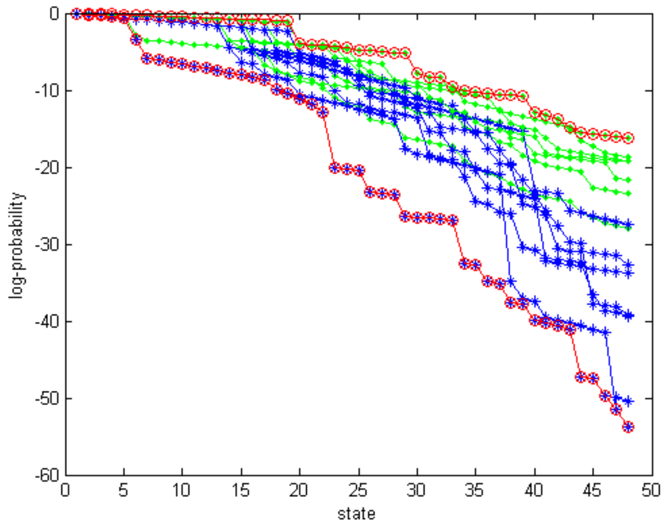


Figure 5.16: The log-probability of daily routine in corridor near WC with boundary values (red circles), best values (green lines), and worst values (blue stars)

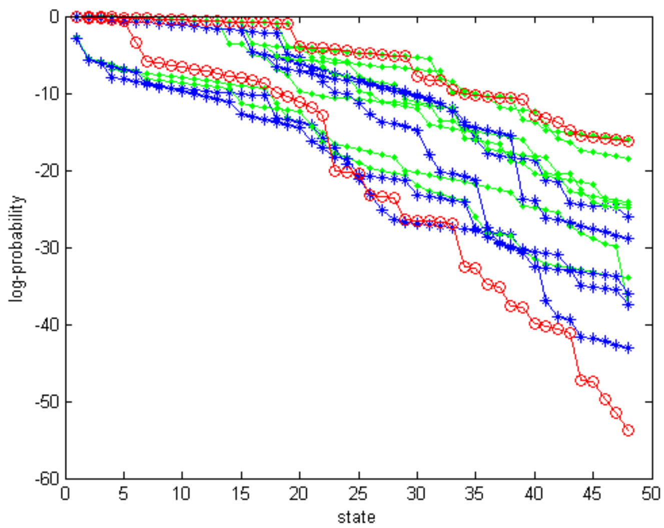


Figure 5.17: The log-probability of daily routine outside of behavior model in corridor near WC with boundary values (red circles), best values (green lines), and worst values (blue stars)

vation of the whole living environment, other motion detectors should be used to detect activity in the different areas and the results from these motion detectors fused together. Secondly, if the single sensor became defective then the learnt behavior model and the hidden Markov model will be wrong too. In an extreme case, if the used motion detector totally ceased to function there would be no signal to be used to learn the model. In order to overcome these problems the data from motion detectors in different locations will be fused together. The behavior model and hidden Markov model will be learnt using the fusion data. With this approach all of the living environment will be under observation. In the following, data from sensors in different locations will be fused together to learn the appropriate behavior model and hidden Markov model. The model will be used to test observed activity routines of the user.

5.3.2 User Daily Routine Analyses with Motion Detectors Data

In the previous sections single sensors were used to learn the behavior model and a hidden Markov model. Some observation routines from other days were chosen to test the model and to detect unusual activity routines of the observation routines. In this section the sensor data from these motion detectors in different locations will be fused together and used to learn the behavior model and a hidden Markov model. Then each fused state value routine will be used to match the model. Furthermore some observations from activity routines from some other days outside of the model will be used to match the hidden Markov model.

The fused state values are shown in figure 5.7. The fused state values were from all 5 motion detectors in the living environment. The parameters of the hidden Markov model will be obtained based on the behavior model. These parameters were used in the forward algorithm and obtained the probability of each state when an observation activity routine matched to the model. The approach was introduced in chapter 3.

In following figure 5.18 the state probability of each routine matched to the model is shown. In figure 5.18 the best value routines concentrate on the top boundary, even some of the worst value routines concentrate on the top boundary too. That means most of the routines in the behavior model from all 5 motion detectors are very similar. There is only one routine diverges to a large extent from the other routines in the model. The probability of the routine matched to the model is shown in the bottom boundary in figure 5.18.

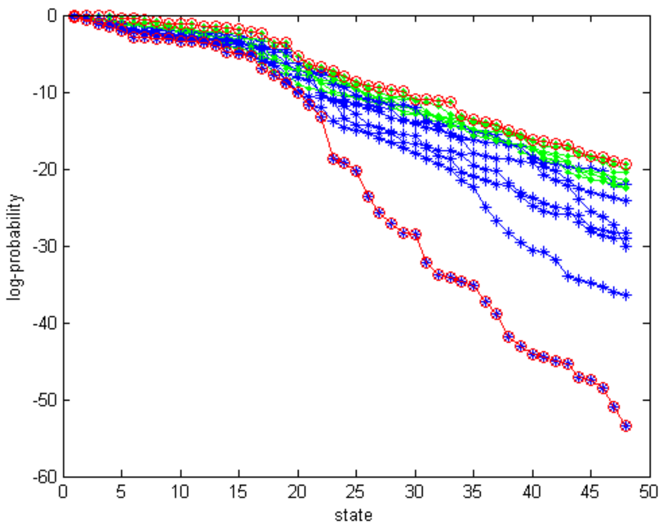


Figure 5.18: The log-probability of daily routine with all motion detectors with boundary values (red circles), best values (green lines), and worst values (blue stars)

Figure 5.19 indicates the matching results from 7 observation routines came from others days that were outside of the model. In the figure there are only 6 green routines and 6 blue routines which reached the last state of the model. This is because one observation routine did not match to the model and the probability was zero. In the figure the matching result with zero probability will not be shown.

On the other side in figure 5.19 there is only one routine outside of the boundary. That means one observation routine has more different with the model. Comparing the test result with the test result from single sensors, there are more observation routines which did not match to these models and the matching values outside of the boundaries (showed in previous section in figure 5.15 and in figure 5.17). In fact the used observation routines for a single sensor and for motion detectors all came from the same 7 days. The difference surely came from the models. A hypothesis is that the model with sensor fusion data has more ability to include different observation routines than the models which came from single sensor. In the next section all the sensors in living environment will be used to learn the model and to test observation routines which came from the same 7 days as above single sensor and motion detector used. If the hypothesis is right the test value should be closer to the boundary.

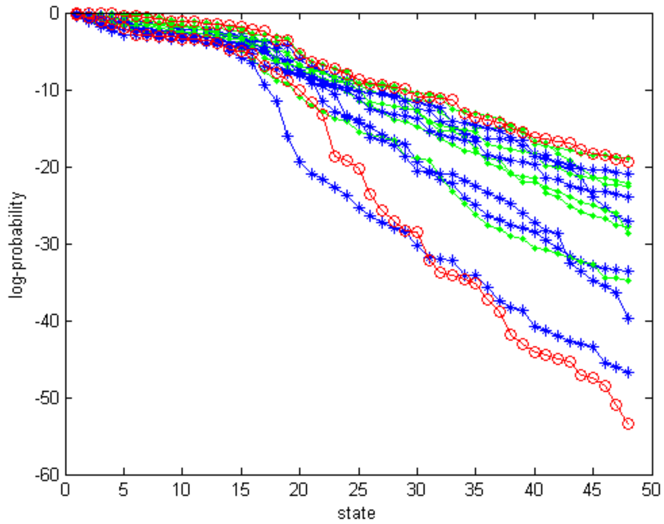


Figure 5.19: The log-probability of daily routine outside of behavior model with all motion detectors with boundary values (red circles), best values (green lines), and worst values (blue stars)

5.3.3 User Daily Routine Analyses with Multi-Sensors Data

In this section all the data from different types of sensors in living environment of the user will be fused together and used to learn the behavior model. A Hidden Markov model whose basic parameters came from the behavior model will be used to test observation routines.

Figure 5.20 shows the result that the observation routines were chosen from the learnt behavior model. The value differences between top and bottom boundaries in the figure are smaller than the differences from the most of single sensors (from about -25 to -50 in figure 5.12, and from about -15 to -55 in figure 5.16) and motion detectors (from about -20 to -55 in figure 5.18). That means with the fused sensor state data the differences between different routines in the model are reduced. This is because after sensor state data fusion the influences from each single sensor to the model are reduced.

Figure 5.21 shows the result that the observation routines came from different days than the days used to learn the model. The test result showed that all the values were nearly between the boundaries. The above hypothesis that the model with sensor fused data has more ability to

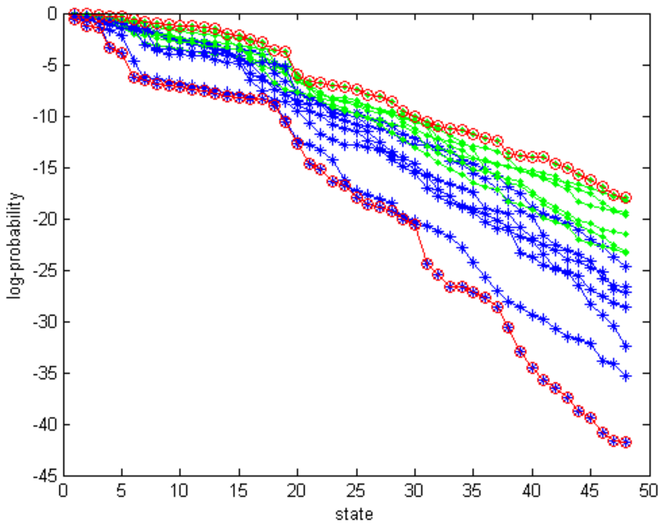


Figure 5.20: The log-probability of daily routine with all sensors with boundary values (red circles), best values (green lines), and worst values (blue stars)

include different observation routines than the models which came from single sensor are shown in the above figures. The reason is that, for example one single sensor has state value 0 and the observation routine has state value 1. The two state values are totally different. But in the fused data if there are 10 values in a state and only one value is different. So the difference between the model state and the observation state is only 0.1. This is the reason why the model which came from sensor fused state data has more inclusive ability than the model which came from single sensor state data.

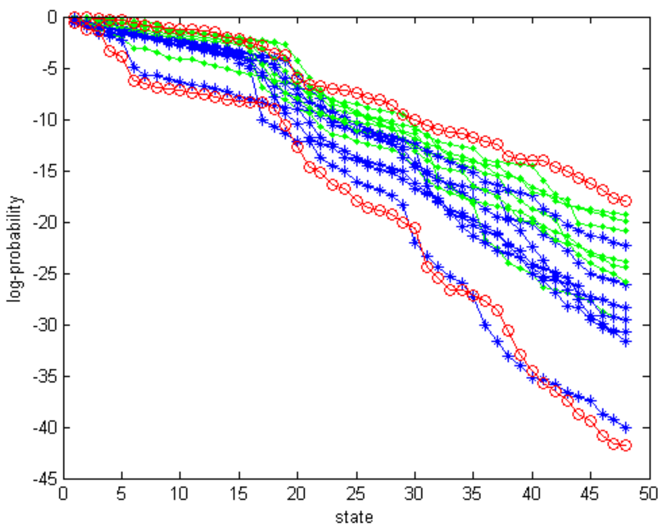


Figure 5.21: The log-probability of daily routine outside of behavior model with all sensors with boundary values (red circles), best values (green lines), and worst values (blue stars)

5.4 Automatic Scenario Detection with Hidden Markov Model

In the section a hidden Markov model (HMM) will be used to detect the daily routine in the behavior model which has the most probability matching to an observation activity routine. Firstly, automatic scenario detection for which the behavior model, hidden Markov model and the observation routine came from data from a single sensor will be introduced. Then automatic scenario detection for which all the models and observation routines came from fused data of multi-sensors will be introduced and discussed.

5.4.1 Automatic Scenario Detection with Single Motion Detector Data

Figure 5.22 shows 7 activity routines from a single sensor which was installed at the entrance. The 7 activity routines are from 7 days. One day from the 7 days was randomly chosen as the observation day. The activity routine from the observation day was used to test the model, which is shown in figure 5.23. The result shows that the second day in figure 5.22 has the highest probability matching to the observation day. The best matching day is shown with a green line in figure 5.22. In fact the randomly chosen day is the second day. That means the model has the ability to find the best matching routine to an observation routine. The same day has the best value when it matches itself.

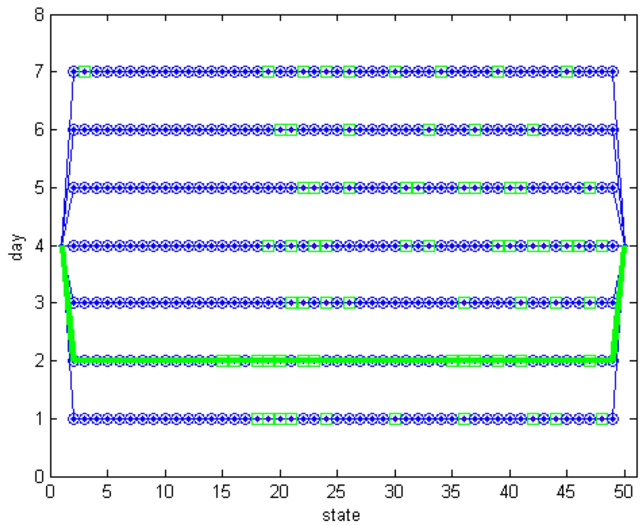


Figure 5.22: The best daily routine matches to observation routine from single sensor

Figure 5.23 was the behavior model of the single motion detector which was installed at the entrance. In the figure 5.22 if the consecutive states have the same state value they should be fused together. If the single state has different value from its neighbor the state should be fused with the neighbor states. If the fused consecutive states have similarity more than 0.5 they would be fused together.

Figure 5.24 showed the log-probability value that all the routines in figure 5.22 was tested by an observation activity routine. Because there are 7 routines in figure 5.22 so there are 7 matching results. The best value sequence is shown with a top red circle line. The best value is about -24 at

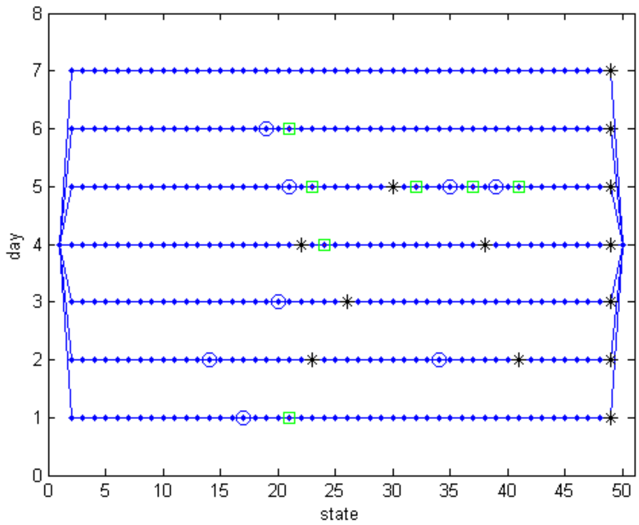


Figure 5.23: The behavior model of the single motion detector which was installed at the entrance

the 48th state. The second best value is about -26 and the worst value is about -32 (the bottom red circle line). That means the model has the ability to detect the best matching routine.

Furthermore, in figure 5.22 at the second routine (green line), there are some consecutive states with state values which change from 0 (with circle symbol) to 1 (with square symbol) or from 1 to 0. For example, about at state 15 the state value changed from 0 to 1, and at about state 25 the state value changed from 1 to 0; at about state 35 the state value changed from 0 to 1; at about state 40 the state value changed from 1 to 0; and at about state 45 the state value changed from 0 to 1. The state value changing may be lead by the self transition probability, transition probability, and emission probability changing between states (introduced in the behavior model chapter). In figure 5.24 at the top boundary red line we see the log-probability value changing at the same states. For example at about states 15, 25, 35, 40, and 45 there is more change of probability value than other states.

Now another routine outside of the model was chosen to match the 7 routines in the model. Figure 5.25 shows that the 5th day in the model matched the chosen routine best. In the 5th routine at about state 20, 25, 30, 40, and 45 there are changes in state value.

Figure 5.26 shows the matching results. The best value is about -22.5 and the second best value is about -23.2. Both values were nearly the same. The worst value was about -28. The value differences between these 7 routines were not very big. This was because the routines in the model have more similarity. In fact the routines came from the single sensor installed at the entrance. The differences between these routines were very small (from about -20 to -30 in figure 5.14). When an observation routine matched the 7 routines (with high similarity) the results changed in a small value interval (from -23.2 to -28). Furthermore in the figure at the top boundary red line there are more probability values changing than other states at about state 20, 25, 30, 40, and 45.

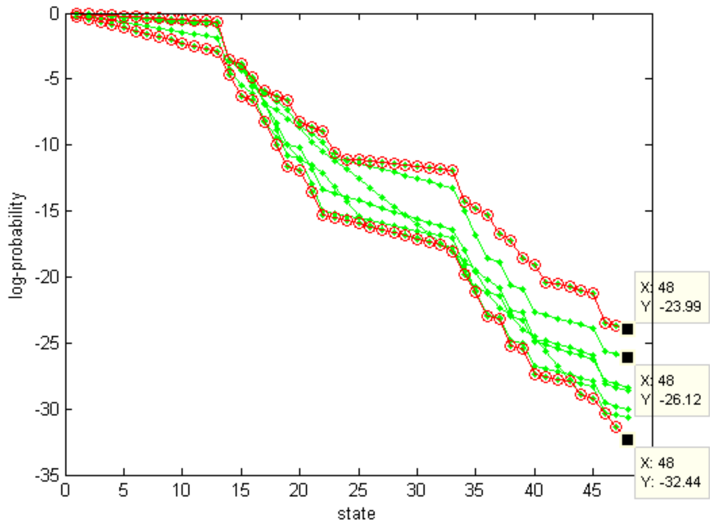


Figure 5.24: The log-probability of behavior model matches to observation routine from single sensor

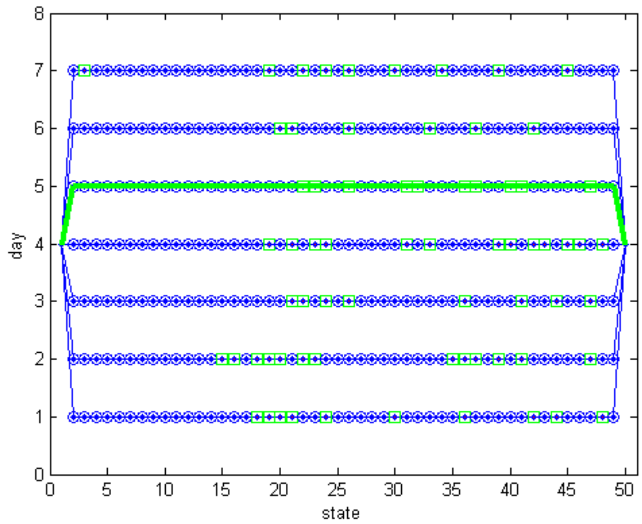


Figure 5.25: The best daily routine matches to observation routine outside of model from single sensor

5.4.2 Automatic Scenario Detection with Motion Detectors Data

Figure 5.27 shows a behavior model from motion detectors which were installed in all of the living environment. There are 5 motion detectors in total. An observation daily activity routine was randomly chosen from the above model and the model which is shown in figure 5.28 was tested with it. The result shows (with green line) that the first day in the model has the biggest probability matching to the observation routine. In fact the first day was the randomly chosen day. That means the hidden Markov model from the behavior model has the ability to detect the most matching daily activity routine.

Furthermore, in the figure there are some states with mixture values (symbol star). That means in the states some motion detectors have state value 1 and others have state value 0. In the first

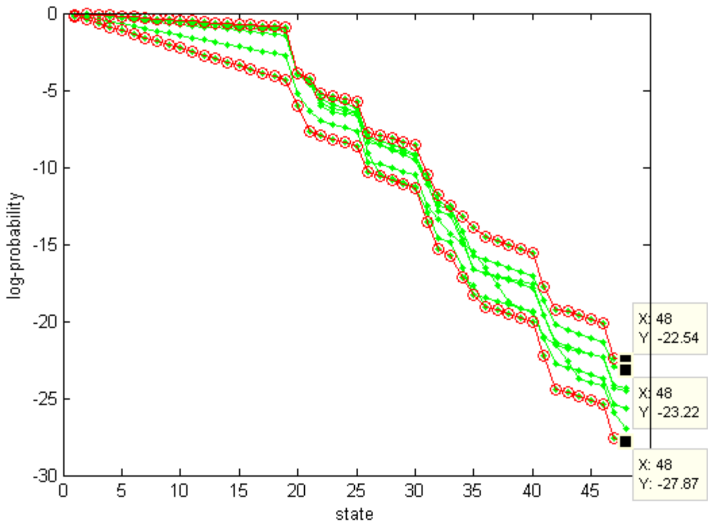


Figure 5.26: The log-probability of behavior model matches to observation routine outside of model from single sensor

routine of the figure between about state 15 and 20 there are state values changing from 0 to 1, at about state 40 the state value changes from 0 to 1, and at about state 45 the state value changes from 1 to 0. Just as introduced above, the states value changing caused the transition probability to change. The result is shown in figure 5.29.

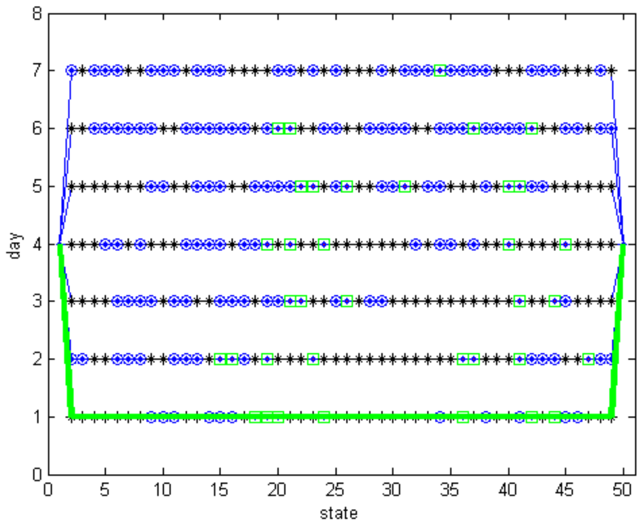


Figure 5.27: The best daily routine matches to observation routine from motion detection sensors

Figure 5.28 indicates the behavior model of the motion detectors which were installed in the living environment. In the figure, before state 15 in each routine there are merged states with mixture values. That means the user has activity at night.

Figure 5.29 shows the probability value that the observation activity routine matches to all the 7 routines in the behavior model. Because there are 7 routines in the model so there are 7 matching

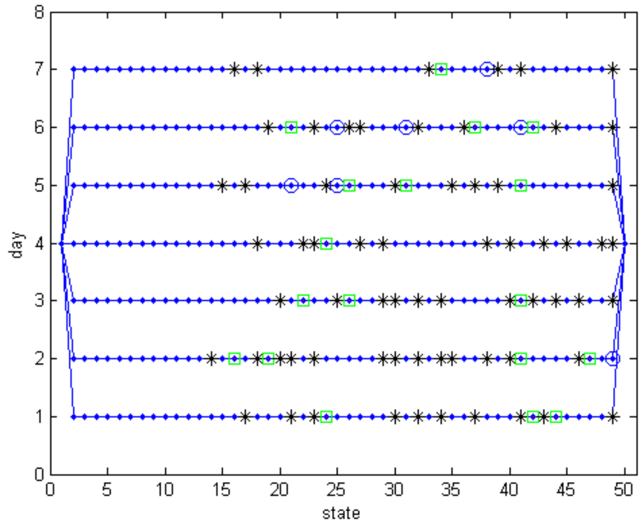


Figure 5.28: The behavior model of the motion detectors which were installed in the living environment

results. The best value sequence is shown with the top red circle line. It indicates the first routine in the model which matched to the observation routine best with value -20. The second best value is about -26 and the worst matching value is about -32 (the bottom red circle line). They have big differences compared to the best matching value. That means the model has the ability to distinguish different routines. Furthermore between state 15 and 20, at about state 40, and at about state 45 the probability values change more than for other states. This is because the state value changes in the first routine of figure 5.27.

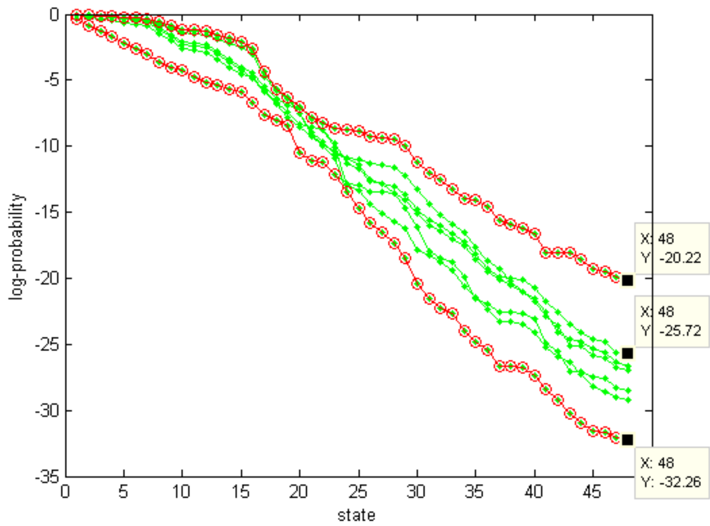


Figure 5.29: The log-probability of behavior model matches to observation routine from motion detection sensors

Figure 5.30 shows a test result for the observation routine which was chosen from an other day outside of the model. The 6th routine (with green line) in the model has the best similarity to the chosen routine. Furthermore in the 6th routine there are states values changing (from 1 to 0 or from 0 to 1) at about state 20, 37, and 40.

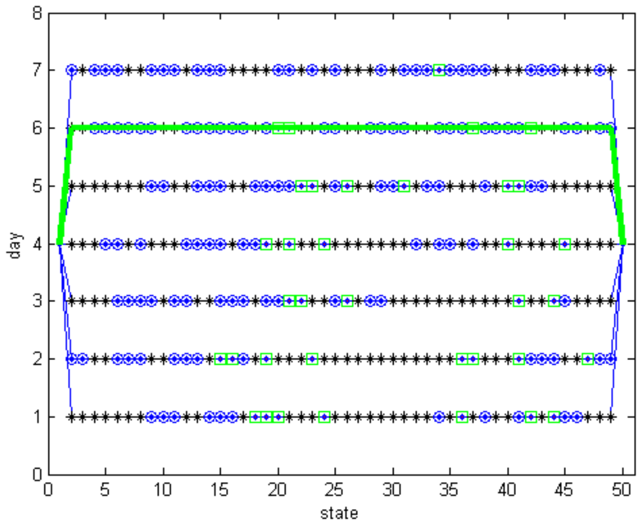


Figure 5.30: The best daily routine matches to observation routine from motion detection sensors which outside of model

Figure 5.31 showed the probability value that the observation activity routine matches to all the routines in the behavior model. In contrast to figure 5.29 the differences between these value sequences were not great (only one routine with matching value -34). This is because the chosen routine has similarity to 6 routines in the model. Only one routine in the model did not match well. If the chosen routine was very similar to one of the routines in the model there should be one value sequence is greatly different from other routines. Furthermore at states about 20, 37, and 40 there are changes in the probability values. The reason has been introduced above.

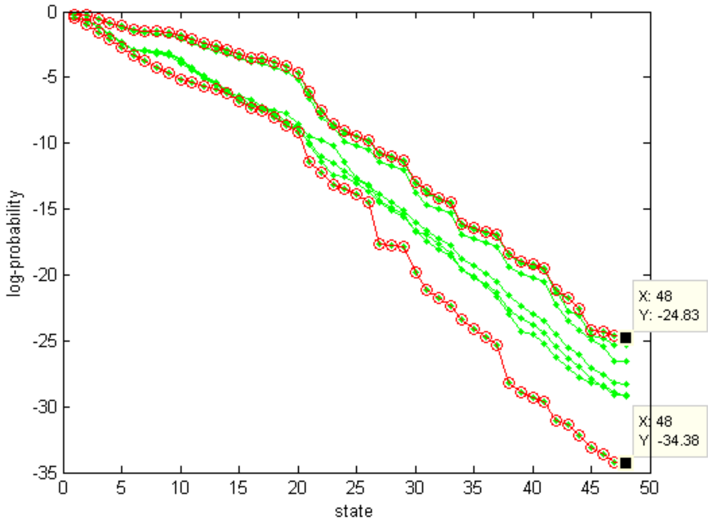


Figure 5.31: The log-probability of behavior model matches to observation routine from motion detection sensors which outside of model

5.4.3 Automatic Scenario Detection with Multi-Sensors Data

Figure 5.32 shows the best daily routine (with green line) matched to an observation routine which was chosen from the same figure. The behavior model which was shown in figure 5.9 came from all the sensors in the living environment. Together with the test results from all motion detectors and from single sensors it demonstrates that the hidden Markov model has the ability to detect the best matching routine which is unrelated to the model coming from single or multi-sensors. Furthermore in the third routine of the figure there are only 5 states with value 0; all other states have mixture values. Based on only the mixture state values, it is difficult to judge in which state the states value changed more.

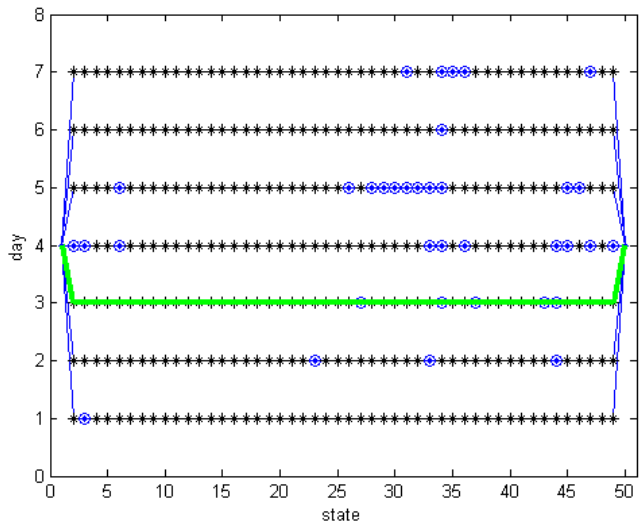


Figure 5.32: The best daily routine matches to observation routine from multi-sensors

Figure 5.33 indicated the matching result, that is how well the observation routine matches to the model. The best value sequence was from the third routine in the model. The best value is about -18. The best value sequence had great differences compared to other value sequences. The second best value is about -22. That means the model has the ability to detect the best matched routine. Furthermore at about states 15, 20, 37, and 42 there are more changes in probability values on the top red line. That means in the third routine of figure 5.32 the states value changed more at these states.

Figure 5.34 shows the best daily routine (with green line) matching to an observation routine which was chosen outside of the model. In the first routine there is only one state with value 0 and all other states have mixture values. It is difficult to judge which states values changed more.

Figure 5.35 shows the result of the best matching routine in the model in comparison to the observation routine. The best value is -20 and the value is smaller than the best value -18 from figure 5.33. That means that the routine came from the model has the biggest value than the routine came from outside of the model. The result indicated again that the hidden Markov model has the ability to detect the best matched routine. Furthermore in the top red line at states about 15 and 30 more probability values change than in other states. That means in the first routine of figure 5.34 the states values changed more at the same states.

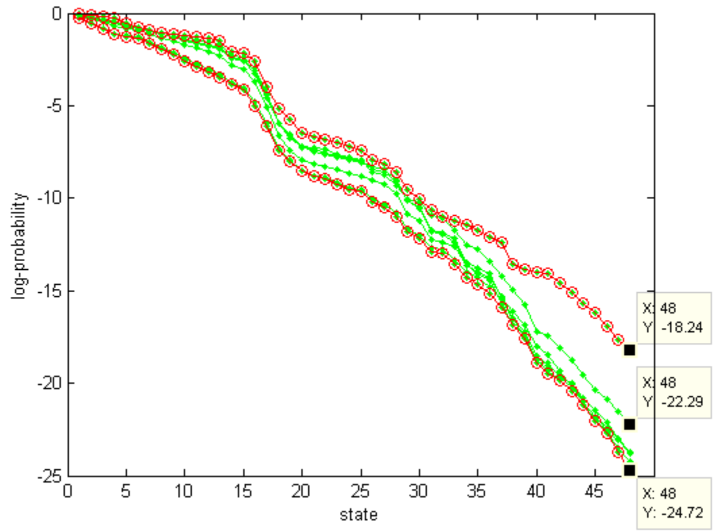


Figure 5.33: The log-probability of behavior model matches to observation routine from multi-sensors

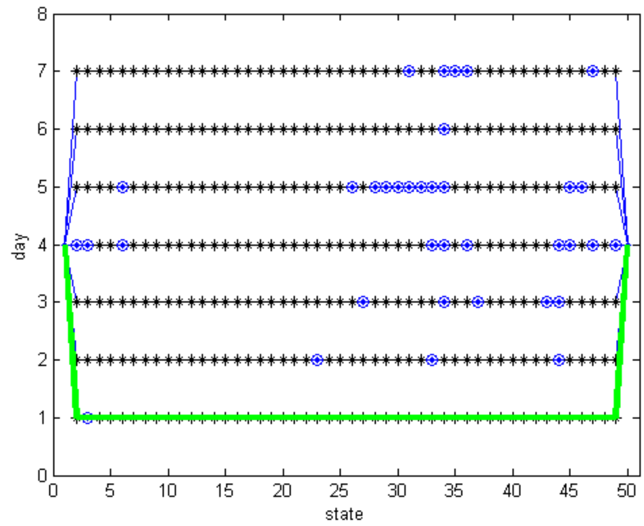


Figure 5.34: The best daily routine matches to observation routine from multi-sensors which outside of model

5.5 Automatic Unusual Scenario Detection with Hidden Markov Model

In the section the hidden Markov model will be used to detect unusual scenarios in the daily life of the user. Here the term "unusual scenario" means unusual daily activity which is different from the normal routine of the user as learnt in the model. For example, based on the behavior model the user should get up about 7:30 in the morning. If one day there is no user activity at 7:30 and there has still been no activity at 9:00, an alarm signal will be sent to the caregiver. Or if one night (when there should be no or very little user activity) a lot of activity is detected, this

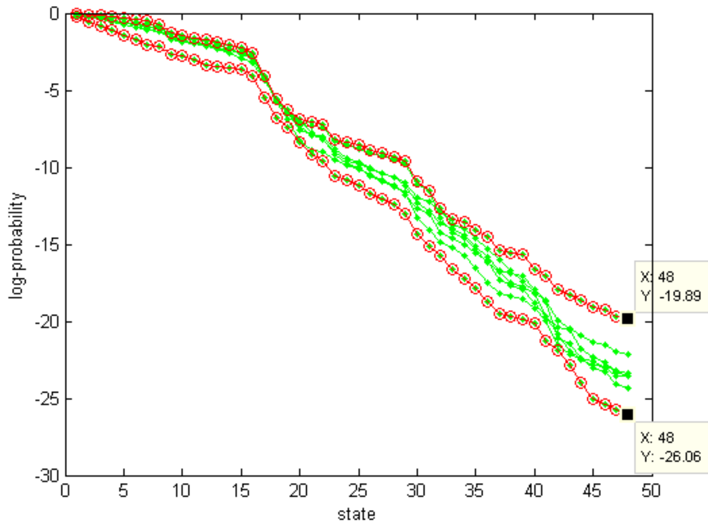


Figure 5.35: The log-probability of behavior model matches to observation routine from multi-sensors which outside of model

should be considered an unusual activity, so an alarm signal will be sent to the caregiver. In this section the single motion detector will be used to detect unusual activity at first. Then fusing of data from motion detectors in the living environment will be used to detect unusual activity of the user. Lastly all sensors in living environment will be used to detect unusual activity of the user. The result of the three approaches will be compared and discussed.

5.5.1 Automatic Unusual Scenario Detection with Single Motion Detector Data

A single motion detector installed in living room will be used to detect unusual activity of the user. For example a motion detector did not send any signal suddenly (state value stayed at 0), then based on the earlier learnt model and the probability boundaries of the model to judge if the situation unusual or not (the matching values outside of the boundaries or not).

Figure 5.36 shows the result that an observation routine with states value 0 all the time. The green lines in the figure are the probability of the matching result. From the 5th state 2 green lines are outside of the boundary. But the distance between the green line and top red boundary is not large. About after state 13 the distance increases more and more. The third green line is outside the boundary too. According to the matching result (3 lines are outside of the boundary) an alarm should be sent to caregiver at night, but it will be a wrong alarm. At night the user sleeps in the living room/bedroom area. This is normal activity. That means that if only the data from a single sensor is used it is possible to get a wrong result and a wrong alarm.

Figure 5.37 showed the result that an observation routine with states value 1 all the time. About after the 12th state the green lines are outside of the boundary and the distance increases more and more. In such a situation an alarm signal should be sent to the caregiver. But the problem is there are activities all through the night in the living room/bedroom area. The situation is unusual. An alarm sent to the caregiver in the morning (after state 12) is likely to be too late.

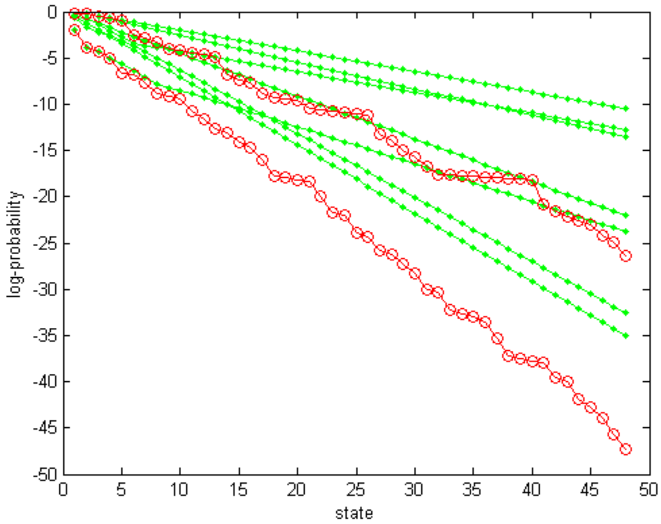


Figure 5.36: The log-probability of behavior model matches to observation routine from single sensor with value 0 in living room

The bottom green lines of the figure indicate the worst matched routines. They were inside of the model boundary, so they are interpreted as normal activity but in fact they were unusual. The result indicates again that, using only a single sensor, it is possible to get wrong results and wrong alarms. Furthermore there are only 4 green lines in the figure. That means the observation routine did not match the other 3 routines in the model. If the observation routine did not match the routine in the model the probability should be 0, so there will be no matching result to show in the figure.

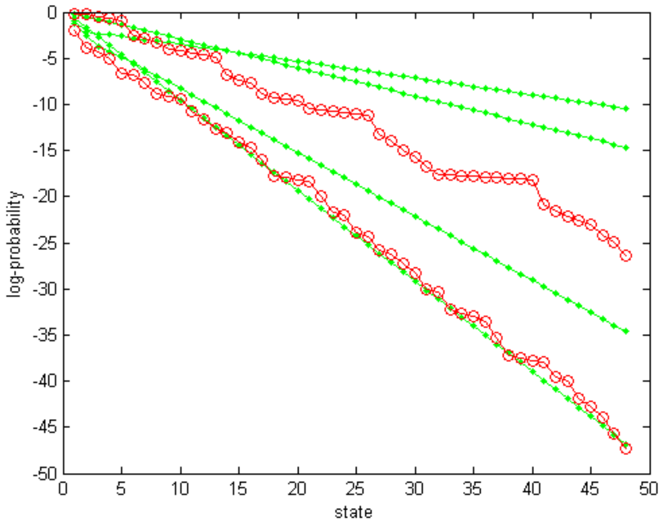


Figure 5.37: The log-probability of behavior model matches to observation routine from single sensor with value 1 in living room

Another single motion detector example came from the motion detector at the entrance. The result is shown in figure 5.38. The observation routine had all states value 0 all the time. The green lines on top of the figure were outside of the boundary about after state 20 (about 10:00 in the morning). The green lines on bottom of the figure were outside of the boundary about after state 23. That means the unusual activity was detected by the single sensor at the entrance.

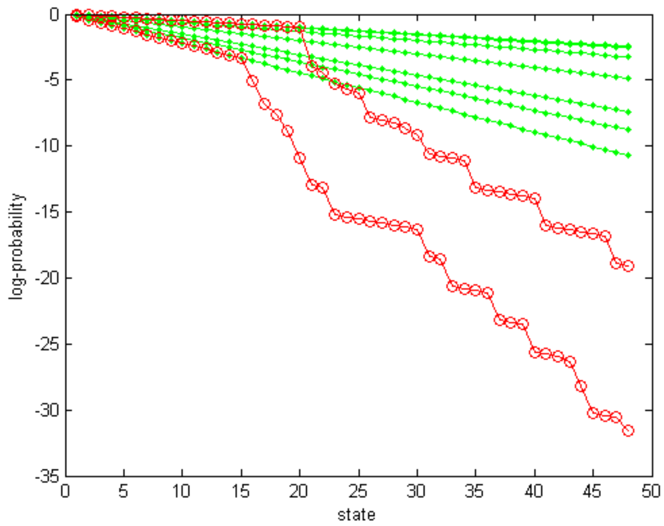


Figure 5.38: The log-probability of behavior model matches to observation routine from single sensor with value 0 at entrance

Figure 5.39 shows the result from the same motion detector at entrance but the observation routine had all states value 1. That means there is user activity all the time in this area. It is unusual activity and the green lines were outside of the bottom boundary at the first state. The above results indicate that, with the model from single motion detector at the entrance, the unusual activity would be detected. But the observation routine matched only 4 routines in the model.

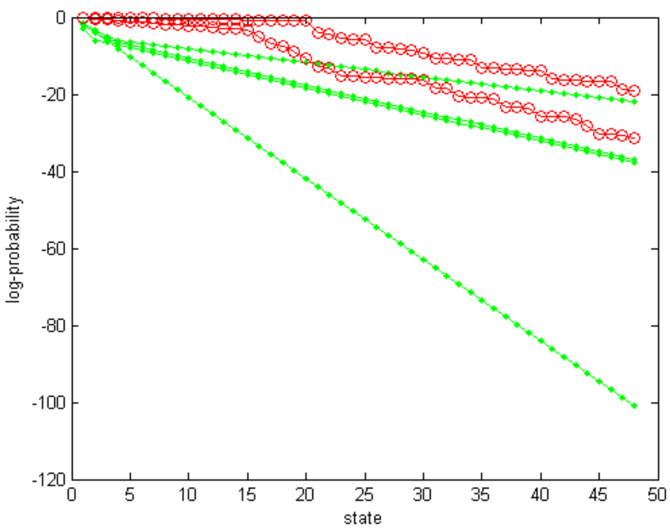


Figure 5.39: The log-probability of behavior model matches to observation routine from single sensor with value 1 at entrance

Figure 5.40 showed the result from a motion detector in the corridor and the observation routine had all states value 0. About after state 20 the green lines were outside of the red boundary. Based on the top green lines an alarm should be sent at about 10:00 in the morning.

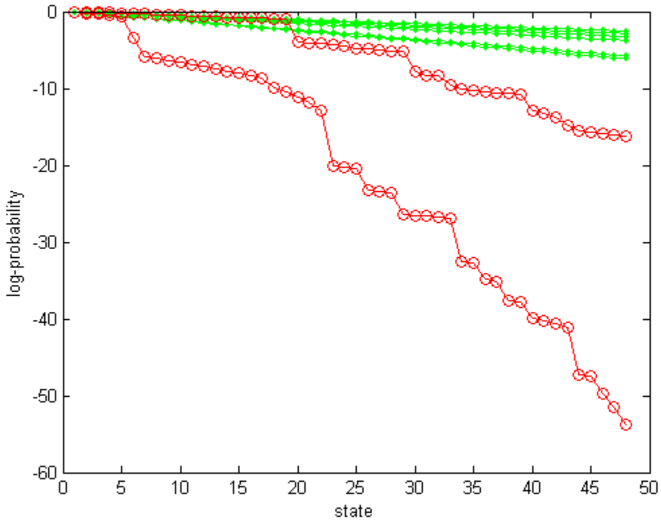


Figure 5.40: The log-probability of behavior model matches to observation routine from single sensor with value 0 at corridor

Figure 5.41 shows the result from the the same motion detector at corridor, but the observation routine had all states value 1. The green lines were outside of the bottom boundary from the first state. But the observation routine matched only 2 routines in the model.

The above results indicate that the observation state with value 1 (or 0) stayed at the first state with state value 1 (or 0) in the model and cannot go outside of the same state. The other situation is that the observation state with value 1 (or 0) stayed at the first mixture state. If the consecutive state in the model with value 1 (or 0) then the observation state can go outside of the mixture state, but then it stays at the state with value 1 (or 0). Because it stays at the same state the self transition will keep the same value, it leads to the probability value changing linearly.

5.5.2 Automatic Unusual Scenario Detection with Motion Detectors Data

In the section the fused data from all motion detectors in the living environment will be used to match the model and to find the unusual activity. At first all the sensor states value will be keep to 0. That means a wrong situation in which the user has no activity at any time. Figure 5.42 shows the result from all motion detectors in the living environment and the observation routine had all states value 0. The top and bottom green lines are very near and they went outside of the top boundary about after state 20 (about 10:00) in the morning. That meant the unusual situation was detected by the sensors.

Figure 5.43 shows the result from all motion detectors in the living environment and the observation routine had all states value 1. The green lines went outside of the bottom boundary at the

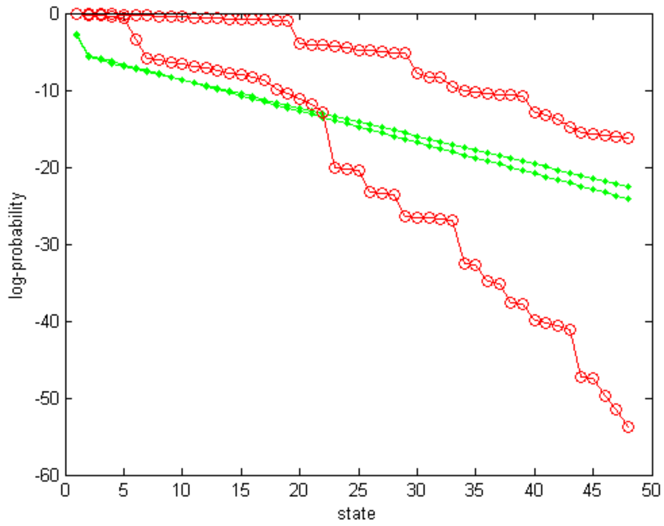


Figure 5.41: The log-probability of behavior model matches to observation routine from single sensor with value 1 at corridor

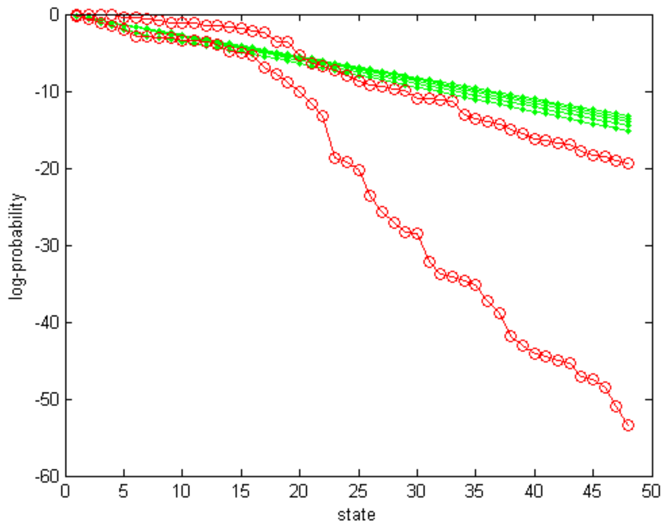


Figure 5.42: The log-probability of behavior model matche to observation routine from motion detectors with value 0

first state. That meant the unusual situation detected was by the sensors at night. The results showed in figure 5.42 and figure 5.43 indicate that with the unusual activity would be detected with the model from all motion detectors.

5.5.3 Automatic Unusual Scenario Detection with Multi-Sensors Data

In the section the fused data from all sensors in the living environment will be used to test the model and to find unusual activity. At first all the sensors states values will be keep to 0. That means a wrong situation that the user has not any activity at all during the time period measured.

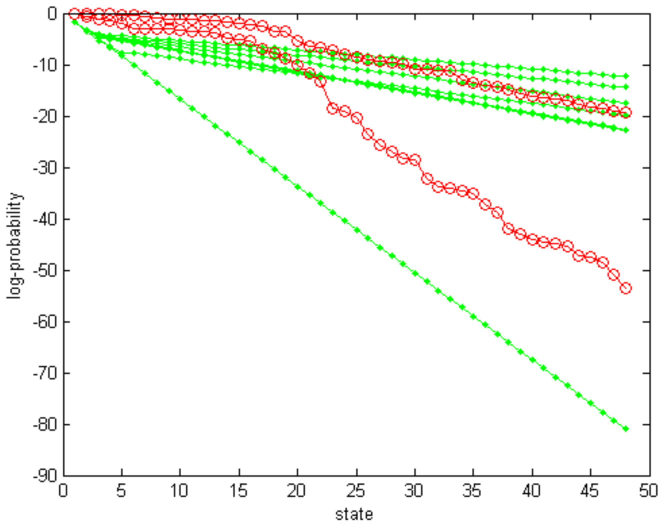


Figure 5.43: The log-probability of behavior model matches to observation routine from motion detectors with value 1

Figure 5.44 shows the result from all sensors in the living environment and the observation routine had state value 0 throughout. The 2 top green lines went outside of the top boundary about state 20 (about 10:00) in the morning. That means the unusual situation was detected by the sensors. The bottom green line went outside of the bottom boundary about after state 12 but went back into the boundary again about after state 20. That means that, based on the top green line, the unusual situation is detected with all 0 states optimal.

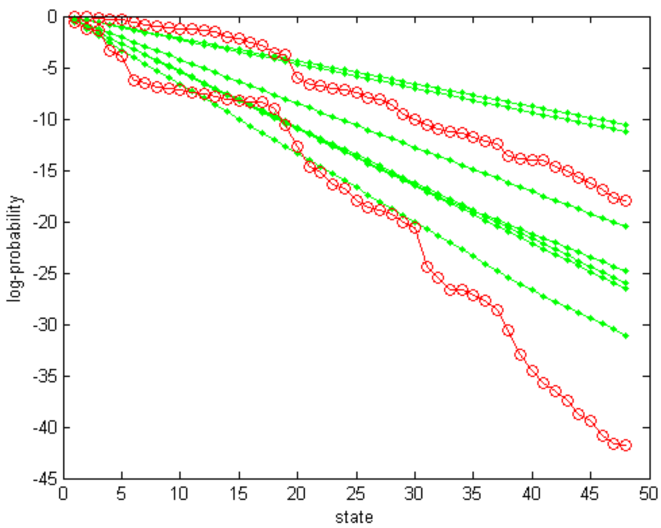


Figure 5.44: The log-probability of behavior model matches to observation routine from multi-sensors with value 0

Figure 5.45 shows the result from all sensors in the living environment. The observation routine had all states value 1. The top green lines were very near to the bottom boundary and a few states were outside of the boundary. The bottom green lines went outside of the bottom boundary at first state and went back into the boundary about after 20th and 30th states. That means the bottom green lines could be used to detect the unusual situation with all 1 states optimal.

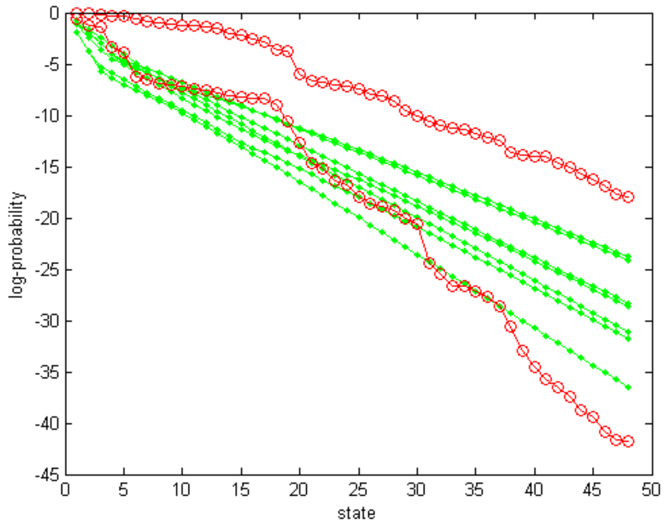


Figure 5.45: The log-probability of behavior model matches to observation routine from multi-sensors with value 1

All the above results indicate that using a single sensor to detect unusual activity or situations was not reliable. A wrong alarm will sometimes be sent, and an unusual situation cannot always be detected as early as possible. With fusion data from multi-sensors the results were better. Using the top line to detect the situation of no activity and the bottom line to detect the situation with constant activity gave optimal results. But in real daily life this kind of situation seldom happens. In the following case study section the real routines will be used to match the model in order to detect the unusual routines.

5.6 Comparing the Result between Single Sensors and Multi-Sensors

In this section the automatic scenario detection resulting from single sensors and multi-sensors will be compared. The goal to do such a kind of comparing is to compare the end result with different methods to test if the result was comparable. At first the result between single sensors and motion detectors will be compared, and then the result between single sensors and multi-sensors will be compared.

5.6.1 The Result between Single Sensors and Motion Detectors

In the following the matching results between using single sensors to learn the model and using motion detectors to learn the model will be compared. Here the single sensor was the individual sensor from the 5 motion detectors. In the above sections the learnt models were from 7 daily activity routines. In the following sections the models will be learnt from 14 daily routines. This kind of changing has no special reasons, just to show the method could be used in different time durations. In the real situation the model should be learnt from one or two months worth of data (in order to include more different types of activity routines as introduced in above sections). But in the thesis it was not necessary to show a figure with 60 routines (the figure should be bigger than one page or each state in the daily routines would be very small and could not be watched clearly). Here activity routines from all 5 motion detectors in 14 days were used to learn the behavior model and get the value boundaries. An observation sequence was chosen outside of the time interval of the model and it will be used to match the model. The best and worst routines in the model will be found using the probability of the matching result.

How to get the model value boundaries and the matching results had been introduced in figure 5.10 and figure 5.11. In the following how to get the matching results from single sensors will be introduced in figure 5.46.

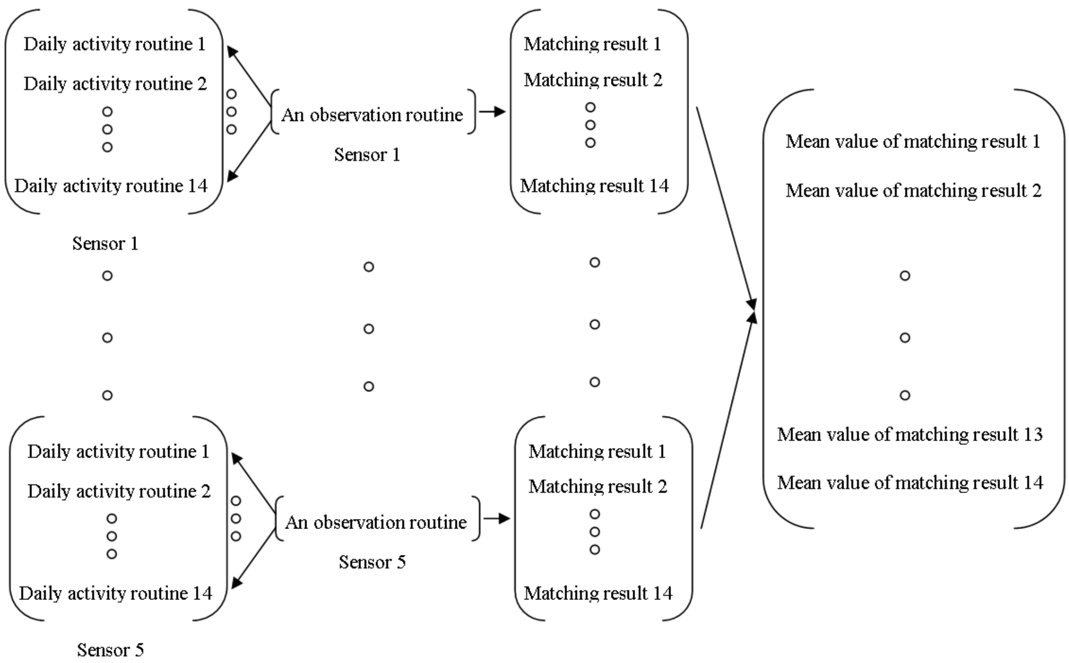


Figure 5.46: The method getting the matching results from single sensors

Figure 5.46 shows each single sensor learnt model with the activity routines from the 14 days. An observation sequence was chosen from the same day as the observation sequence from the 5 motion detectors used. The observation sequence will be used to match the model. Then the results from the 5 single sensors will be merged together to get the result. Finally the results from the single sensors and from 5 motion detectors would be compared. If the results overlap each other or have a small deviation that means the used approaches can be comparable.

Figure 5.47 shows the log-probability the observation sequence matched with the model which was from the 5 motion detectors. The red lines are the value boundaries from the model. The green lines are the matching results when the observation sequence matched with the model. The best value is about -25 and the worst value is about -34. They are both in the boundary.

Figure 5.48 indicated the best and worst matched routines in the behavior model using the

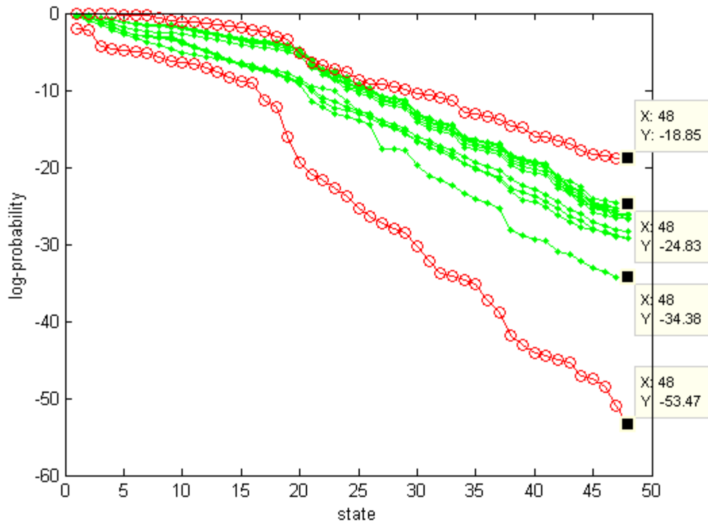


Figure 5.47: The log-probability of observation routine matches to the behavior model from motion detectors and the observation routine was chosen outside of the time interval of the model

probability value from the 14 days. The green line was the best matched routine and the red line was the worst matched routine to the observation sequence.

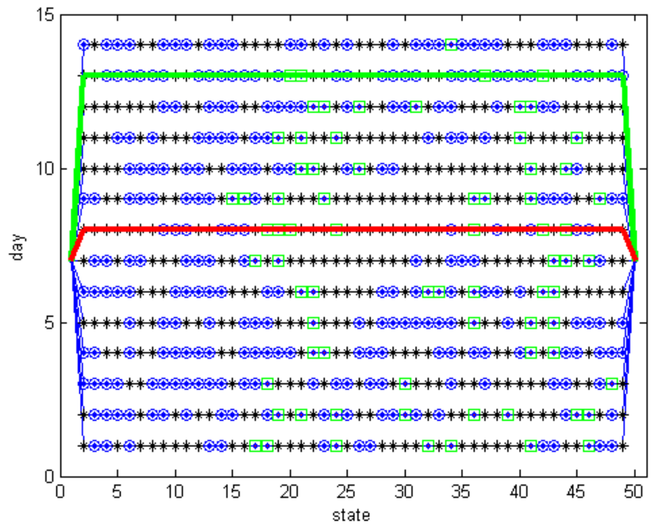


Figure 5.48: The best and worst matching routines by an observation routine from motion detectors

Figure 5.49 was basically the same as figure 5.47. However the graph has additional blue lines. The blue lines show the information from single motion detectors. The value differences between them are -23 from best blue line value and -25 from best green line value, -30 from worst blue line value and -34 from worst green line value. Many of the lines overlapped. That means the test results from the two methods compare favourably.

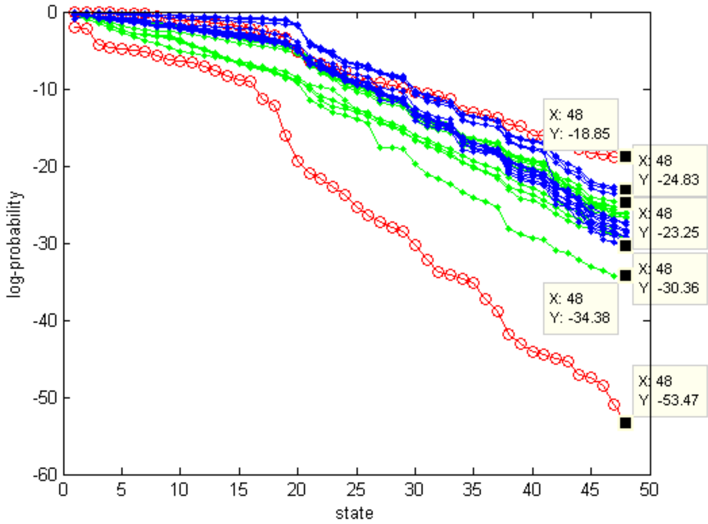


Figure 5.49: The log-probability of observation routine matched to the behavior model from motion detectors. The observation routine was from single motion detectors and was chosen outside of the time interval of the model

5.6.2 The Result between Single Sensors and Multi-Sensors

The matching result between single sensors to the model and motion detectors to the model have been compared above. Now the same result from single sensors to model were used to compare with the result from the multi-sensors to the model (the data from the multi sensors). The blue lines in figure 5.50 are the same as in figure 5.49. But the green lines and red lines were from all multi-sensors. The value differences between them were about -22 from the best green line value and about -23 from the best blue line value, about -28 from the worst green line value and about -30 from the worst blue line value. Many of the lines overlapped in places. This indicates again that the test results from the used method can be comparable.

In the following the matching result between single sensors and the model and multi-sensors and the model will be compared. Here the single sensor was an individual sensor chosen from the multi-sensors. Figure 5.51 shows the probability that the multi-sensor model (learnt from data collected over 14 days) and an observation sequence outside of the model time interval matched with the multi-sensor model. The best value was about -22 and the worst value was about -28. All the green lines were between the red boundary lines.

Figure 5.52 indicated the best and worst matched routines in the behavior model reference to the probability values from the 14 days. The green line was the best matched routine and the red line was the worst matched routine to an observation sequence which was chosen outside of the model.

Figure 5.53 shows the compared result between single sensors and multi-sensors. The green lines and red boundary lines are the results from the multi-sensors, blue lines were from single sensors. Each single sensor was one of the multi-sensor group. The result showed that the green lines and blue lines did not overlap. The differences were quite large. The worst blue lines value was about -18, but this was bigger than the best green lines value of -22. The result was different to the result from above. Was it the method used wrong or was something else wrong?

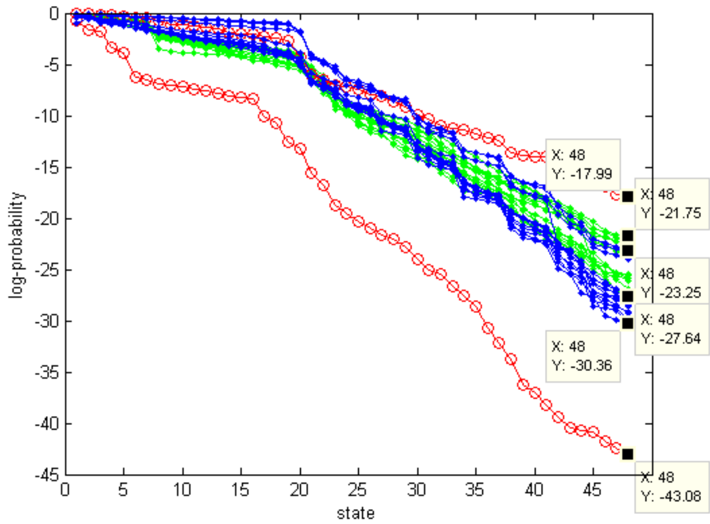


Figure 5.50: The log-probability of observation routine matched to the behavior model from multi-sensors. The observation routine was from single motion detectors and was chosen outside of the time interval of the model

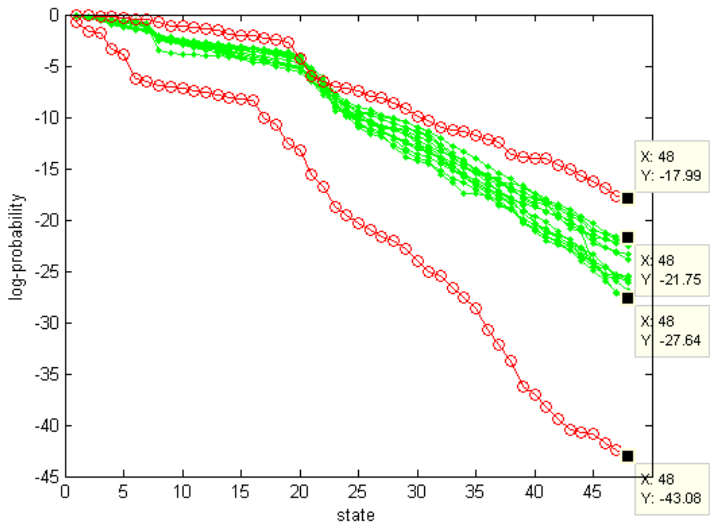


Figure 5.51: The log-probability of observation routine matched to the behavior model from multi-sensors. The observation routine was from multi-sensors and was chosen outside of the time interval of the model

Through analyzing the sensor state routines it became clear that there were two types of routine. The first type of routine had more similarities and the consecutive states had mostly the same value. These activity routines reflect the activity of the user. The other kind of sensor state routine had less similarity and most of the routines had the same state value in all the state routines. The result was that all the state routines could be merged into one or two merged states. In the following these single sensors were separated into two groups.

In figure 5.54 the first kind of sensors were used to compare with the multi-sensor model. The

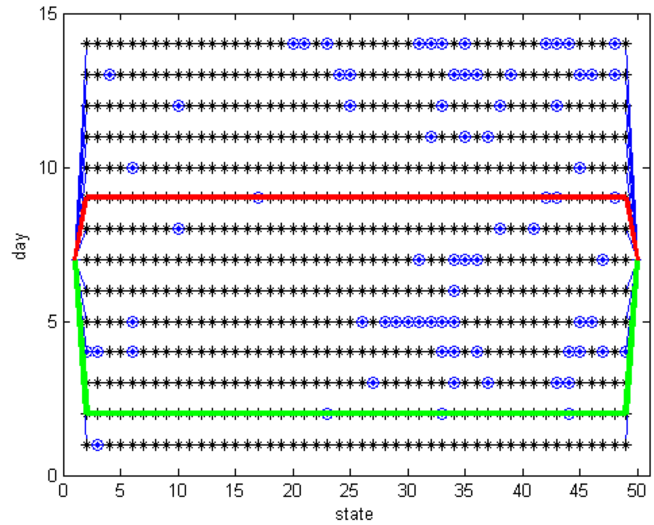


Figure 5.52: The best and worst matching routine by an observation routine outside of model time interval from multi-sensors

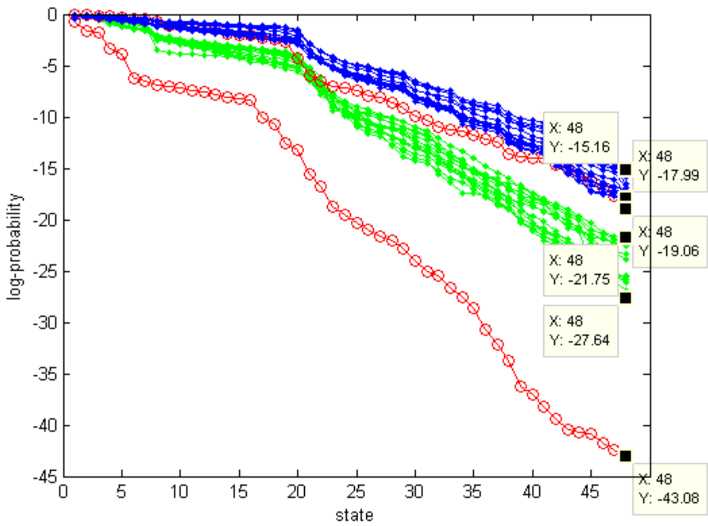


Figure 5.53: The log-probability of observation routine matched to the behavior model from multi-sensors. The observation routine was from single sensors and was chosen outside of the time interval of the model

blue lines were the results from these single sensors. The results show a large degree of overlap between the green and blue lines. The best value from green lines was about -22 and the best value from blue lines was about -24. The worst value from green lines was about -28 and the worst value from blue lines was about -30. These overlapping lines indicated the chosen sensors were well matched to the multi-sensor model.

In figure 5.55 the second kind of sensors were used to compare with the multi-sensor model. The blue lines represent the results from these single sensors. The results show that the cluster of green lines and the cluster of blue lines deviate greatly from each other. The best value from the

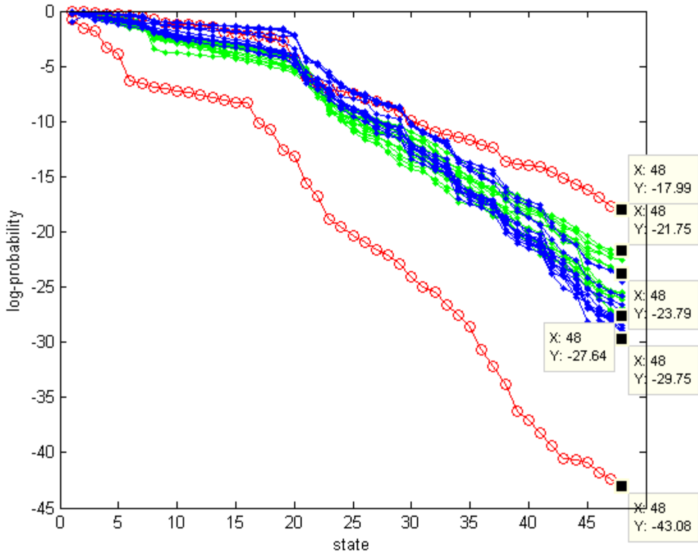


Figure 5.54: The log-probability of observation routine matches to the behavior model from multi-sensors. The observation routine was from the first of the chosen single sensors and it was chosen outside of the time interval of the model

green lines was about -22 and the best value from the blue lines was about -2. The worst value from green lines was about -28 and the worst value from blue lines was about -7. These green and blue lines were totally separated. The smallest value from blue lines is much bigger than the best value from green lines. The bigger values from blue lines were because of the merged state from these sensor routines. There were fewer states and all of the observation states had nearly the same value. The observation states stayed at the same merged state and so the probability value is always bigger.

Furthermore, when data from the two different types of sensors were merged together, the result was that all values deviated from the true values. Because the second types of sensors had much bigger values so the overall results from single sensors were bigger than for the multi-sensors. Figure 5.53 indicates the result. There the values of the blue lines were bigger than the green lines. The above 3 figures indicate that the matching results (green lines) from the multi-sensor model have smaller deviations than the matching result from the single sensor model (blue lines).

5.7 Case Study

In above sections the built model and the value boundary were used to find the best and worst matched routines in the model for an observation sequence. In the following, 3 special sequences were chosen in order to test if the model can detect the difference.

Figure 5.56 shows the best and worst matched routines to an observation sequence. The green one was the best matched routine to the observation sequence and the red one was the worst matched to the observation sequence. Here the best and worst routines came from the values when an observation was tested against the model.

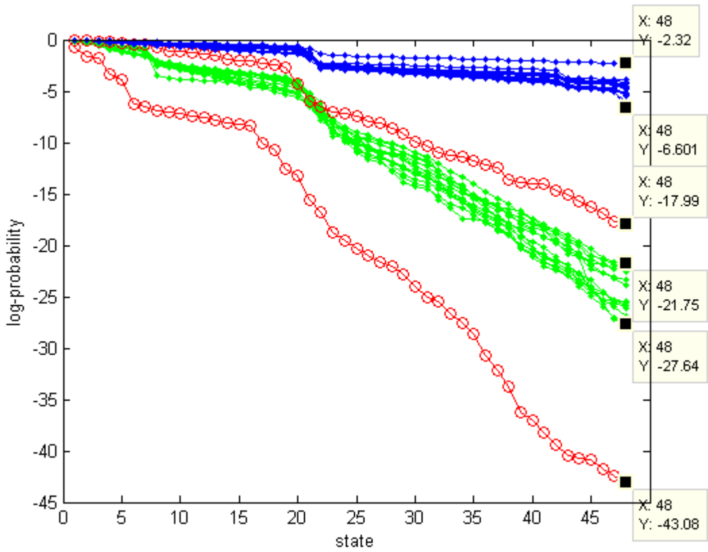


Figure 5.55: The log-probability of observation routine matched to the behavior model from multi-sensors. The observation routine was from the second of the chosen single sensors and was chosen outside of the time interval of the model

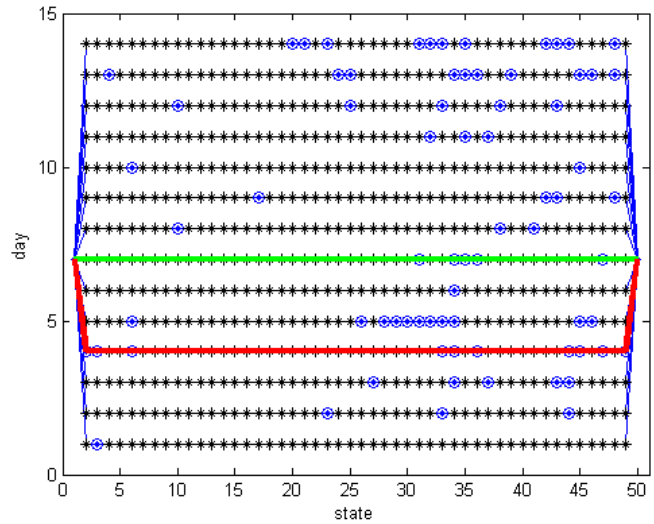


Figure 5.56: The best and worst matching routines by an observation routine, which was chosen outside of the time interval of the behavior model

Figure 5.57 indicates the value result when an observation routine from another day and it was used to match the model. The red lines were the boundary from the model. The green lines were the test result when an observation sequence was tested against the model. If the green lines went outside of the boundary that means the observation routine did not match some routines in the model. This indicates unusual activity.

In figure 5.57 there are some green lines outside of the red boundary. It seems that the observation sequence does not match the model well. In figure 5.57 after state 11 (each state has time duration 30 minutes, so state 11 is about 5:30 in the morning) there are some green lines outside of the bottom red boundary. This indicates unusual user activity. Furthermore after state 21 (about 10:30 in the morning) there are some green lines outside of the top red boundary. This again indicates unusual user activity. Based on these two unusual activities, an alarm should be sent to the caregiver. In the real data, no signal was sent to the controller before state 25 (about 12:30 in the afternoon). If no user activity was detected in the living environment during the whole morning, it is likely that the user was ill or that some other unusual situation has happened. Following the model, an aware signal should be sent to the user at 5:30 (after state 11). Furthermore an alarm signal should be sent to the caregiver if there has not been any answer from the user after 10:30 (after state 21).

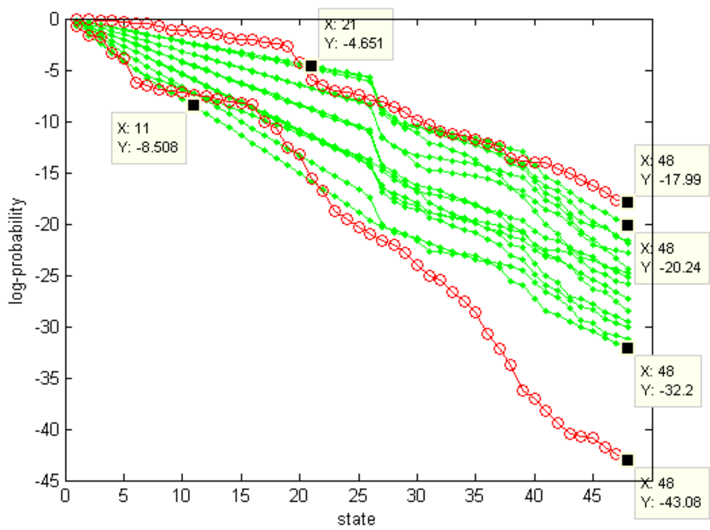


Figure 5.57: The log-probability of multi-sensors behavior model matches to an observation routine, which was chosen outside of the time interval of the model

Figure 5.58 shows the best and worst matched routines to an observation sequence with more activity. The blue one was the best matched routine to the observation sequence and the red one was the worst matched to the observation sequence. Here the best and worst routines came from the values when an observation sequence was tested by the model.

Figure 5.59 indicates the result when an observation routine with more activity from another day was used to match the model. In figure 5.59 there were some green lines outside of the red boundary. This occurred from state 32 to state 38. That meant there had been unusual activity in the time interval of these states in the living environment of the user. The daily book which was written by the user indicated that in this time interval there were visitors from 15:00 to 20:00. The activities of the visitors provide a reason why the green lines are outside of the red boundary.

Figure 5.60 shows the best and worst matched routines to an observation sequence with more activity at night by the user. The blue one was the routine best matched to the observation sequence and the red one was the worst matched to the observation sequence. Here the best and worst routines came from the values when an observation sequence was tested against the model.

Figure 5.61 indicates the result when an observation routine from another day with more activity at night and it was used to match the model. In figure 5.61 there are some green lines outside of the red boundary from state 1 to state 5. Furthermore, one of the green lines continues outside

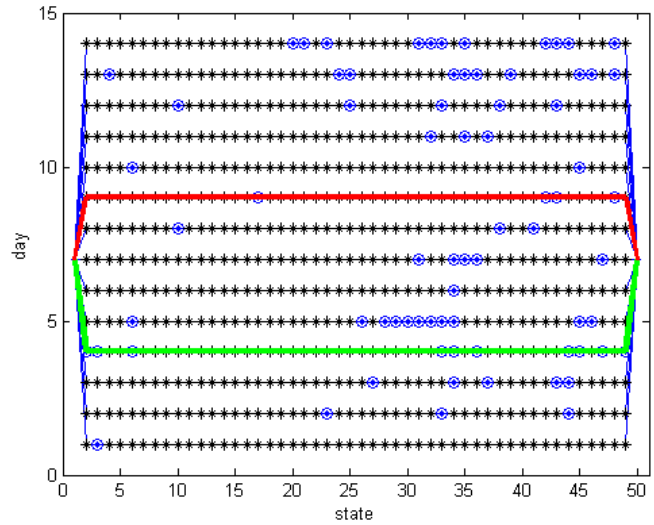


Figure 5.58: The best and worst matching routines by an observation routine with more activity, which was chosen outside of the time interval of the behavior model

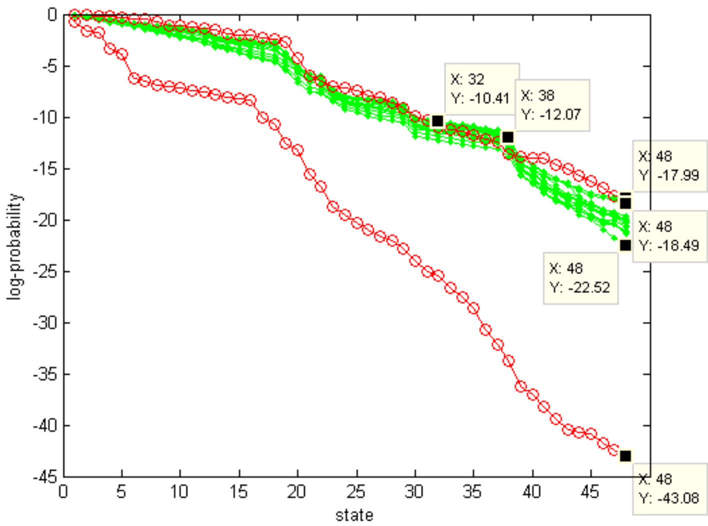


Figure 5.59: The log-probability of multi-sensors behavior model matched to an observation routine with more activity, which was chosen outside of the time interval of the model

the boundary up to state 20. The conclusion is that there was some unusual activity in the living environment of the user in the time interval of these states. The daily book written by the user indicated that at night the user went to the WC 5 times. This is different from the other days in the daily book, when the user went to the WC at night only 1 or 2 times. So going to the WC 5 times represents unusual activity for this user. In the daily book the user had written that she had had a bad night on this occasion. Furthermore the user got up at 6:00 in the morning. Rising at this time was also unusual for the user. In figure 5.61, from about state 2 to state 18 there were again some lines outside of the red boundary, indicating unusual user activity. These activities at night and in the morning are the reason that the green lines are outside of the red

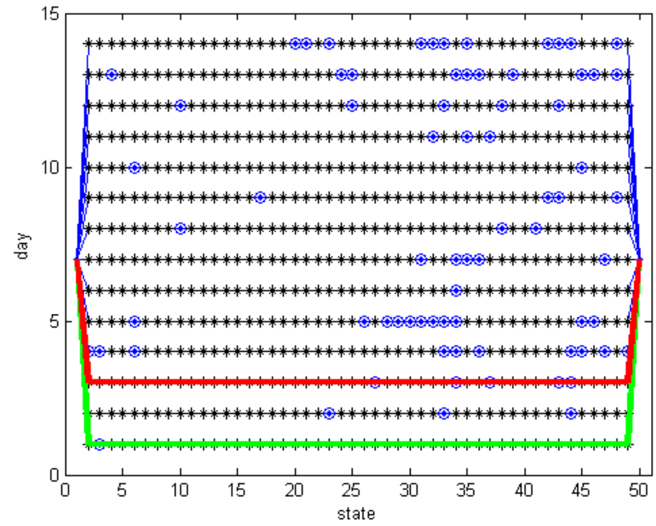


Figure 5.60: The best and worst matching routines by an observation routine with more activity at night, chosen outside of the time interval of the behavior model

boundary during this period.

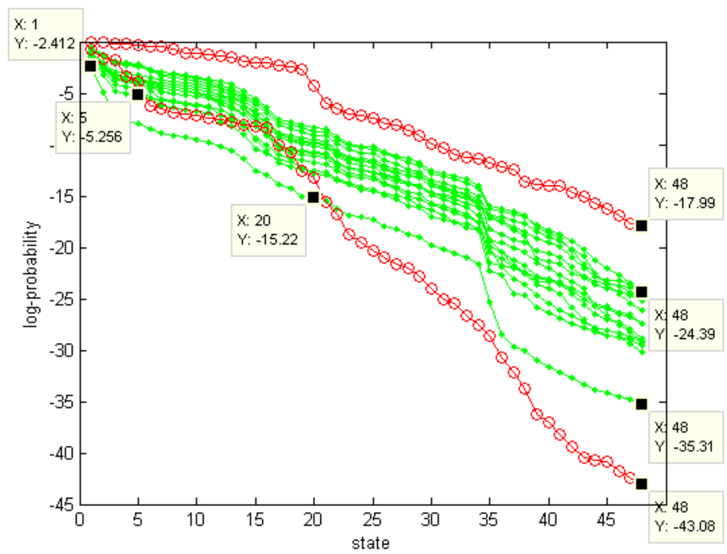


Figure 5.61: The log-probability of multi-sensors behavior model matched to an observation routine with more activity at night, which was chosen outside of the time interval of the model

5.8 Result and Discussion

Based on the different behavior models, automatic scenario detection was realized in this chapter. For regular behavior, the split-merge algorithm and Gaussian mixture model were applied to learn the behavior model, and the cluster mean and standard deviation values were used to detect unusual behavior. For random behavior (daily routines of the user), the hidden Markov model and forward algorithm were used to realize automatic scenario detection and detect unusual behavior.

Data from individual motion detectors, fusion data from multiple motion detectors, and fusion data from different types of sensors were used to learn the behavior model, then combined with a hidden Markov model and forward algorithm to realize automatic scenario detection. The behavior model from single sensors was learnt and compared with observation sequences outside of the time interval of the model. Then the behavior models from fusion sensor data were learnt and compared with observation sequences outside of the time interval of the model. Behavior models from the single sensors and from fused multi-sensors were used to detect the best matching routine in the behavior model. Furthermore, unusual scenarios (with all state values 0 or 1 from single sensors and multi-sensors) were detected using the hidden Markov model and forward algorithm.

Generally, the hidden Markov model from single sensors was limited to detecting unusual scenarios and, depending on the chosen sensor, the results differed. The performance of multi-sensors was better than that of single sensors. The test results showed that various unusual scenarios were detected by appropriate multi-sensors (situations with unusual activities at night were immediately detected by multi-sensors, situations with no activity at night were detected in the morning). The conclusion was that the test results with fusion data from multi-sensors were better than the results from single sensors.

In order to compare the automatic scenario detection results from single sensors and multi-sensors, an observation activity sequence was checked against the models from single sensors and multi-sensors. The test results from all single sensors were merged together and compared with the results from the multi-sensors. The results overlapped each other, which means the employed approaches were comparable when an observation sequence was used to compare the models.

Three examples were given in order to explain how the model has the ability to detect unusual activity by the user. In one case there was no activity by the user from night until noon, and the model detected this unusual activity. In the second example the user had visitors who created unusually high amounts of activity. In the last example the user displayed more activity at night, due to being unable to sleep, and going to the WC more often than usual. All of the above unusual situations were detected by the model. Furthermore, the user kept a log in which they noted what happened every day, and the unusual situations detected above could be correlated with events noted in the daily log.

The entirety of the data from the test period of about 4 months was checked against the model learnt from multi-sensors, and the results indicated that all unusual situations detected by the model could be found in the daily log. One of these unusual situations was when the user had visitors in the house. Because the model was learnt while the user was alone, the extra activity caused by other persons in the living environment was correctly treated as an unusual situation. In order to reduce false alarms in such situations, an awareness signal should initially be sent to the user. If the user does not react to the awareness signal after 30 minutes (because in some situations the matching result strayed outside the red boundaries but returned to within the boundaries again after 1 or 2 states, 30 or 60 minutes, meaning the situation was probably normal) an alarm signal should be sent to the caregiver.

On the other hand, there were times when the user had a visitor at home but this situation was not detected by the model. This was because the visitor and user stayed close to each other, so that the sensor system treated the two people as one person. In future work other methods should be used to try to distinguish different persons in the living environment.

6 Conclusion and Outlook

The aim of this work is automatic scenario detection for ambient assisted living. The user is an elderly person who lives alone their own home, and the challenge of the work is to detect and analyze the behavior of the person in their daily life which is subject to variations. How to analyze this dynamic behavior and learn a suitable behavior model are the main issues in the thesis. Another challenge is the question of how to apply the behavior model to the realization of automatic scenario detection.

In order to elucidate the context of the problem, the initial problem statement and reasoning were introduced in Chapter 1. The aims of the thesis were discussed to show the focus of the work, and the scientific methods used were introduced. In Chapter 2 the state of the art in ambient assisted living and scenario recognition were presented. Chapter 3 showed how two differing types of behavior were classified and how, depending on the type of behavior, different algorithms were utilized. The split-merge algorithm and the Gaussian mixture model were used to deal with regular behavior, while the hidden Markov model was exploited for random behavior. In order to make the behavior model complete, sensor fusion was applied and data from different types of sensors fused together to cover the entirety of the living environment. The content is introduced in Chapter 4. In Chapter 5, sensor fusion and behavior model were joined together in order to realize automatic scenario detection, and case study included as examples.

The conclusions of this thesis are derived from Chapters 3, 4, and 5. In the following paragraphs these conclusions will be presented and perspectives for future work will be identified.

6.1 Conclusion of the Thesis

There are 4 main points in the conclusion of this thesis: 1. Different types of behavior were analyzed using different algorithms. 2. State data was used for sensor fusion. 3. Sensor fusion was connected with the hidden Markov model. 4. Automatic scenario detection was realized based on different behavior models. In the following, these points will be introduced and discussed.

6.1.1 Different Behaviors were Analyzed by Different Algorithms

Within the thesis, two different types of behavior by the user are classified. One type of behavior happens regularly in daily life - for example, the user taking medicine tablets every day at nearly the same time in the morning, at noon or in the evening. This type of behavior can be clearly detected, for example by a door contactor installed on the door of the medicine box. When the user opens the medicine box to take medicine tablets, a signal will be sent to a server and the action will be unambiguously detected. This kind of behavior happens daily when the user takes tablets several times (for example around 7:00 in the morning, around 12:00 at noon, and around 19:00 in the evening). With a time duration of, for example, one month, there will thus be some instances of this action (for example around 7:00) gathered together. These instances occurring around 7:00 are assembled into a Gaussian model. Equally, the instances occurring around 12:00 are assembled into another Gaussian model, and the instances occurring around 19:00 into a third Gaussian model. The three resulting Gaussian models are compiled into a Gaussian mixture model. A split-merge algorithm can deal with this Gaussian mixture model and determine the time mean value and deviation with which the behavior occurred. The reason for the deviations from the split-merge algorithm is that the behavior of the user is regular on a daily basis, but not at precisely identical times. For example, a user should take tablets in the morning at 7 o'clock, but in the reality he perhaps takes tablets at 7:10 in the morning, or at 6:50. This kind of time variation causes a deviation in the learning result.

As a result of this, the behavior of the user will be reflected in the learning accuracy. For example, the behavior of one user might for example be time triggered, meaning he takes tables at exactly 7:00 in the morning, 12:00 noon, and 19:00 in the evening. In this case the learning result will be of a very high accuracy. But if a user is more irregular and takes tablets not precisely according to the schedule, the learning result will have a larger deviation and lower accuracy. In certain situations the learning result will be simply wrong when two different time groups are merged together, or one time group splits into two time groups. In fact this is not due to an error in the algorithm - the algorithm just reflects the data. The source of the deviation is the irregular behavior of the person.

The other type of behavior classified in the thesis is random behavior. This kind of behavior happens irregularly in daily life - but it happens sometimes, and detecting it is more complex than detecting regular behavior. An example could be the user going to the WC at night: perhaps one night he goes only once at about 2:00, but another night he goes 3 times - at 0:30, 3:00, and 5:50. Another example: a user might usually sleep a short time in their bedroom after lunch, but at the weekend might instead perform some cleaning in his living room, or go out for a walk. These kinds of activities make the daily routine of the user change from time to time, which also means that the daily behavior scenario changes from day to day. Furthermore the consecutive behaviors only directly relate to the previous behaviors. This characteristic corresponds to the Markov model, so the Markov model was used in the thesis. But because these activities cannot be directly observed - only their parameters can be obtained - the hidden Markov model was utilized.

In the thesis the hidden Markov model was exploited to analyze the daily activity routine of the user, to learn the scenario model of the user and apply the learnt model to detect unusual behavior in daily life. But the main challenge in the thesis is how to learn the behavior model using the hidden Markov model. Related state of the art work attempts to learn the hidden Markov model with data from a single sensor. A disadvantage of this approach is that a single sensor has a limited detection area in the living environment. For instance, a model of a person's behavior in their living room cannot be built with the data from a single sensor which is installed in their bedroom. To improve on this approach, sensor fusion was introduced with many different types of sensors installed in different locations within a living environment. With the help of sensor fusion, the hidden Markov model should completely reflect the behaviors of the user in the entire living environment.

6.1.2 State Data was used for Sensor Fusion

As shown in Chapter 3, the hidden Markov model works with the state and the state data in time interval (for example 30 minutes). During each time interval, different amounts of raw sensor data will be gathered, and this raw data needs to be translated to state data. In the thesis a filter algorithm was used to translate the raw data into state data. If the time summary of the raw sensor data was larger than the predefined threshold value for the time interval, the state value was set to 1. In this way all raw sensor data was translated into state data. But this method is only valid for the data from single sensor. In the thesis a fusion method was introduced to fuse different kinds of sensor state data together.

If in a time interval all the sensors' state data had the same value (0 or 1), the fusion state value was set to the same value (0 or 1). But if the sensors' state values were diverse, the fusion sensor state value was set to a vector. Thus another problem emerged: merging of the state vector in order to reduce the state amount. For single state data this is not a problem: if consecutive state data are the same (all 0 or 1), they can be merged together to the same value. Even when the consecutive state values alternate (0 1 0 1 0 1), they can be merged together in this way.

A compatibility matrix was utilized to resolve the problem presented by multiple sets of state data. In effect, a compatibility matrix is used to determine the correlations between two variable sets, and the correlation value indicates the relationship between the sets. In the thesis, a compatibility matrix was used to merge the consecutive sets of sensor state data. The correlation value was treated as the parameter which decides whether two states could be merged or not. By doing this, the problem of sensor fusion with mixed state data was resolved.

6.1.3 Sensor Fusion was Connected with the Hidden Markov Model

In earlier work the hidden Markov model received state data from single sensor, and the learnt behavior model therefore represented the single sensor's state data. However, if another sensor is installed in another location in the living environment, the learnt behavior model will be different, for the two models will have different structures. In the thesis, many sensors were installed in different locations in the living environment, and each sensor learned a model. How to merge these multiple models posed another challenge. Much time was spent reflecting on this problem, and the ultimate decision was that it would be better to fuse the sensor state data first than to merge models with different structures.

Furthermore, a new behavior model was be learnt due to the fact that the states in the model came from sensor fusion state data. The fusion state is not a single value but a state set. The consecutive sets in each daily routine were once again merged using a compatibility matrix, but the daily routines from different days were not merged. Thus the learnt daily model was concise and clear, with no crossed lines or complicated structure (as occurred in models with single sensor state data which merged the daily routines from different days).

6.1.4 Automatic Scenario Detection was Realized Based on Different Behavior models

The regular behavior model was built using Gaussian mixture models and the split-merge algorithm. The mean value and standard deviation value of each group was used to judge if a detected behavior should be classified as “normal” or not. If a detected behavior had not happened in the defined time interval, the system should send a prompt signal to the user or caregiver.

A Hidden Markov model was used to judge the daily activity routine. If a scenario outside of the model occurred, an alarm signal was triggered. But earlier work [Bru07, p. 137] tried to determine whether the final daily state conformed to the final state of the model to judge whether the scenario was normal or not. Because the last state is at about 24:00 at night (i.e. 0:00 the next day), this model could not determine unusual behavior occurring in the morning quickly enough to be effective.

In this thesis another method was introduced to detect any unusual behavior as quickly as possible. A value interval was predefined for each state based on the hidden Markov model and the training data. If the state value of some daily activity lies outside of this interval, the system is to send an alarm to the caregiver. In this way any unusual behavior should be detected and reacted to earlier.

6.1.5 Short Summary of the Achievements of the Thesis

The achievements of the thesis are: analysis and classification of the different activities of the user and, based on these activities, application of different algorithms; translation of the single sensor raw data to state data and fusion of state data from different types of sensors; merging of consecutive states with a compatibility matrix; overcoming of the fusion problem inherent in the different structures from various single sensors; and the learning of a new concise and clear model to present the fused sensor state data. Automatic scenario detection was realized for different behaviors using adapted algorithms. The method was furthermore enhanced using a hidden Markov model in order to detect unusual behavior as quickly as possible.

6.2 Outlook onto Future Work

The conclusion of the thesis was presented and discussed in the preceding sections. However, more work could be done to improve the presented system in ambient assisted living (AAL). For example, person tracking without the use of cameras could be investigated. Within the existing method, behavior detection from the user should be more detailed and the behavior model of the user should be learnt over a longer time period. These points will be discussed in the following.

6.2.1 Person Tracking Without Camera

Due to privacy concerns, no cameras were used as sensors in this work. However, using only data from motion detectors, door contactors, and accelerometers, it was sometimes difficult to know where exactly the person was within the living environment, in particular when the person moved from one location to another location. This is basically due to the fact that the types of sensors used have no clear detection boundary. For example, a motion detector was installed on a door leading to the living room. When the door was shut, the sensor covered a detection area in the living room - but when the door was open, the detection area was increased to regions outside of the living room. This changing of the sensor's detection area made it difficult to detect where the person actually was.

With different types of sensors it was possible to realize person tracking - for example, motion detectors were installed with overlapping detection areas. If any behavior was detected by both sensors, the behavior of the user must have been happening within the region of overlap. If activity was detected by a motion detector, a door contactor, and another motion detector in another location, it meant the user had moved from one location to another location through the door. With sensor fusion it was possible to track the user within the entire living environment.

6.2.2 Behavior Detection with More Detail

In the thesis the behavior model of the user was learnt and, based on the model, unusual user behavior could be detected. Here the behavior model was concerned with the daily activity routine of the user. By analyzing the data from motion detectors, door contactors, and accelerometers, activity by the user could be detected. But this behavior detection was very general, and it was difficult to detect it with more detail. For example, user behavior in the kitchen could be detected using motion detectors and accelerometers. Details of the behavior, for instance whether the user was eating their dinner, cooking or washing up, were however unknown. It was not even determinable which area of the kitchen the user was in. The reason was that the sensor used could not provide more detailed information. If there were a pressure mat on the floor, the area of activity in the kitchen could be detected in more detail. If the user stood on the pressure mat, would obviously be in the area where the pressure mat was. This example illustrates that if the behavior of the user were supposed to be detected in greater detail, there should be more different types of sensors installed in the living environment. Through sensor fusion the behavior information from different types of sensors could be fused together, and the fusion data indicated the detected behavior with more precision.

6.2.3 Behavior Model with Longer Time Period and More Situations

In the thesis, test data was recorded during a time period of about 4 months. In spite of the daily life habits of many elderly people being relatively stable, there would always be something happening outside of the regular routine. If any "normal" behavior occurred which was not included in the model, there would be a false alarm. This applies especially to those elderly people who do not have a stable daily lifestyle. It will therefore be necessary to record the behavior habits of the user over a longer period of time.

On the other hand, habitual activities in a user's daily life might change with the seasons of the year. In winter the user might, for example, sleep a little longer, and in summer perhaps a little shorter. This difference would require changes in the behavior model. If summer behavior was based only on the winter model there would be regular false alarms. Also, if the user held a birthday party or engaged in other rare but nonetheless "normal" activities, the user's behavior would likely temporarily change as a result, again causing false alarms.

The above examples indicate that for a model to include more different situations, it would be better for the test period to be longer, or that the tests should occur during different seasons of the year. Perhaps one month should be in spring, and one month each in summer, autumn and winter. Data from these different periods could then be compared to each other in order to detect differing and common behavior.

6.2.4 Outlook onto Ambient Assisted Living

Many elderly people these days have problems with modern technology, e.g. communicating via the internet or using a mobile phone. Therefore many products for the elderly are designed as easy to use as possible. But in the future, this situation might change. More and more people aged around 40 or 50 years old these days seldom have problems with modern technology. This means that in the future, products with more functions and complicated applications could be used more easily by aging people. Furthermore, with the help of modern biology and technology, products may be developed which assist the elderly in their hearing, vision, and motive capabilities.

With the help of technology, an intelligent environment could be developed. This thesis with its study of how the behavior of the elderly can be observed and analyzed, and how in the case of dangerous situations, a caregiver can be sent an alarm signal in order for the elderly person to receive help immediately, might represent a first step in this direction. Furthermore, with the help of modern technology, elderly people might get help even when they go outside of their home. This, however, would mean these people would have to be observed all the time and everywhere, and therefore how to preserve their privacy would be another important topic of research.

With the help of the modern technology the elderly people might one day be “safe” all the time as well as being able to live longer. But perhaps another problem will emerge from this: how to deal with the feelings of loneliness many elderly people suffer from. The longer an elderly person lives alone, the more likely he or she will feel lonely. Perhaps a virtual reality could be constructed within the living environment so that elderly people could feel like they were living together with their family and friends. Robots might be used to infuse the virtual living environment with more “reality”. However, it is difficult to say whether technology can take the place of social interaction with real people. Perhaps with the help of technology, elderly people could also be brought together with like-minded others more easily.

Generally speaking, there is much to do in the future withregard to research into the ambient assisted living domain. Based on developing technology, more new ideas and new methods should constantly emerge. But the primary goal remains to help elderly people enjoy their twilight years happily and peacefully with the assistance of technology.

The author of this thesis has devoted his knowledge and energy towards this very goal throughout the last 3 years, and in closing would like to express his utmost respect to elderly and gradually aging people everywhere, and extend his heartfelt best wishes to them.

Literature

[Ait99] Aitkin, M: A General Maximum Likelihood Analysis of Variance Components in Generalized Linear Models. In: *Biometrics* 55 (1999), p. 117–128 40

[Alp04] Alpaydin, Ethem: *Introduction to Machine Learning*. Cambridge, MA : The MIT Press, 2004. – 423 p 26

[AOJ06] Al-Omary, Alauddin Y. ; Jamil, Mohammad S.: A new approach of clustering based machine-learning algorithm. In: *Knowledge-Based Systems* 19 (2006), p. 248–258 27

[BB06] Bulla, Jan ; Bulla, Ingo: Stylized facts of financial time series and hidden semi-Markov models. In: *Computational Statistics & Data Analysis* 51 (2006), p. 2192–2209 31, 32

[BFJZ93] Benner, K.M. ; Feather, M.S. ; Johnson, W.L. ; Zorman, L.A.: Utilizing scenarios in the software development process. In: *Information System Development Process* (1993), p. 117–134 33

[BFPLC10] Bajo, Javier ; Fraile, Juan A. ; Pérez-Lancho, Belén ; Corchado, Juan M.: The THOMAS architecture in Home Care scenarios: A case study. In: *Expert Systems with Applications* 37 (2010), p. 3986–3999 36, 37

[Bil02] Bilmes, Jeff: What HMMs Can do. In: *UWEE Technical Report* UWEETR-2002-0003 (2002), p. 32 32, 49

[BMGR10] Bouchaala, Lobna ; Masmoudi, Afif ; Gargouri, Faiez ; Rebai, Ahmed: Improving algorithms for structure learning in Bayesian Networks using a new implicit score. In: *Expert Systems with Applications* Bd. 37, 2010, p. 5470–5475 55

[BMW10] Braun, Daniel A. ; Mehring, Carsten ; Wolpert, Daniel M.: Structure learning in action. In: *Behavioural Brain Research* Bd. 206, 2010, p. 157–165 55

[Bru07] Bruckner, Dietmar: *Probabilistic Models in Building Automation: Recognizing Scenarios with Statistical Methods*, Vienna University of Technology, PhD thesis, 2007. – 137 p 4, 8, 10, 143

[BSL07] Bruckner, D. ; Sallans, B. ; Lang, R.: Behavior Learning via State Chains from Motion Detector Sensors. In: *Bionetics* (2007), p. 8 49, 55

[BSR06] Bruckner, D. ; Sallans, B. ; Russ, G.: Probability Construction of Semantic Symbols in Building Automation Systems. In: *Proceedings of 2006 IEEE International Conference of Industrial Informatics INDIN* S. 6 (2006), p. 132–137 49, 55

[Bur04] Burgine, M.: Age of People and Aging Problem. In: *Proceedings of the 26th Annual International Conference of the IEEE EMBS*. San Francisco, CA, USA, 2004, p. 655–658 1

[CAJ09] Cook, Diane J. ; Augusto, Juan C. ; Jakkula, Vikramaditya R.: Ambient intelligence: Technologies, applications, and opportunities. In: *Pervasive and Mobile Computing* 5 (2009), p. 277–298 19, 20

[CEMW07] Chi, Hoi-Ming ; Ersoy, Okan K. ; Moskowitz, Herbert ; Ward, Jim: Modeling and optimizing a vendor managed replenishment system using machine learning and genetic algorithms. In: *European Journal of Operational Research* 180 (2007), p. 174–193 28

[CTL08] Chu, Chun-Jung ; Tseng, Vincent S. ; Liang, Tyne: An efficient algorithm for mining temporal high utility itemsets from data streams. In: *The Journal of Systems and Software* 81 (2008), p. 1105–1117 24, 26

[CW09] Chen, Shuyan ; Wang, Wei: Decision tree learning for freeway automatic incident detection. In: *Expert Systems with Applications* 36 (2009), p. 4101–4105 34, 35

[CWF04] Cheng, Shih-Sian ; Wang, Hsin-Min ; Fu, Hsin-Chia: A Model-Selection-Based Self-Splitting Gaussian Mixture Learing with Application to Speaker Identification. In: *EURASIP Journal on Applied Signal Processing* 17 (2004), p. 2626–2639 29, 40

[DRF98] Dojat, Michel ; Ramaux, Nicolas ; Fontaine, Dominique: Scenario recognition for temporal reasoning in medical domains. In: *Artificial Intelligence in Medicine* 14 (1998), p. 139–155 33, 34, 35, 36

[DS00] Dietrich, Dietmar ; Sauter, Thilo: Evolution potentials for fieldbus systems. In: *3rd IEEE International Workshop on Factory Communication Systems*, 2000, p. 343–350 17

[DSVS99] Dssouli, R. ; Somé, S. ; Vaucher, J. ; Salah, A.: A service creation environment based on scenarios. In: *Information and Software Technology* 41 (1999), p. 697–713 33, 34, 36

[DVD+02] Demongeot, Jacques ; Virone, Gilles ; Duchêne, Florence ; Benchetrit, Gila ; Hervé, Thierry ; Noury, Norbert ; Rialle, Vincent: Multi-sensors acquisition, data fusion, knowledge mining and alarm triggering in health smart homes for elderly people. In: *C. R. Biologies* 325 (2002), p. 673–682 20

[Elm01] Elmenreich, Wilfried: An Introdution to Sensor Fusion. In: *Research Report* 47 (2001), p. 4–5 11, 60, 61, 109

[FAP08] Foroughi, Homa ; Aski, Baharak S. ; Pourreza, Hamidreza: Intelligent Video Surveillance for Monitoring Fall Detection of Elderly in Home Environments. In: *proceedings of 11th International Conference on Computer and Information Technology (ICCIT 2008)*. Khulna, Bangladesh, 2008, p. 219–224 4, 5

[FBJ06] Flanagan, J R. ; Bowman, Miles C. ; Johansson, Roland S.: Control strategies in object manipulation tasks. In: *Neurobiology* 16 (2006), p. 650–659 23

[FPSS96] Fayyad, Usama ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic: From Data Mining to Knowledge Discovery in Databases. In: *American Association for Artificial Intelligence* (1996), p. 37–54 24

[FVPW07] Friedewald, Michael ; Vildjiounaite, Elena ; Punie, Yves ; Wright, David: Privacy, identity and security in ambient intelligence: A scenario analysis. In: *Telematics and Informatics* 24 (2007), p. 15–29 36, 37

[GCL06] Genon-Catalot, Valentine ; Laredo, Catherine: Leroux's method for general hidden Markov models. In: *Stochastic Processes and their Applications* 116 (2006), p. 222–243 31, 32

[GG99] Goebel, Michael ; Gruenwald, Le: A survey of data mining and knowledge discovery software tools. In: *SIGKDD Explorations* 1, Issue 1 (1999), p. 20–33 24

[GJH01] Galata, A. ; Johnson, N. ; Hogg, D.: Learning Variable-Length Markov Models of Behavior. In: *Computer Vision and Image Understanding* 81 (2001), p. 398–413 49

[GNV+08] GALÁN, F. ; NUTTIN, M. ; VANHOOYDONCK, D. ; LEW, E. ; FERREZ, P.W. ; PHILIPS, J. ; DEL R. MILLÁN, J.: Continuous Brain-Actuated Control of an Intelligent Wheelchair by Human EEG. In: *4th Intl. Brain-Computer Interface Workshop and Training Course*, 2008, p. 315–320 24

[HGFR06] HAMILTON, Howard J. ; GENG, Liqiang ; FINDLATER, Leah ; RANDALL, Dee J.: Efficient spatio-temporal data mining with GenSpace graphs. In: *Journal of Applied Logic* 4 (2006), p. 192–214 24, 26

[HKWE08] HEERINK, Marcel ; KROESE, Ben ; WIELINGA, Bob ; EVERS, Vanessa: Enjoyment, Intention to Use And Actual Use of a Conversational Robot by Elderly People. In: *HRI'08*. Amsterdam, Netherlands, 2008, p. 113–119 4, 5

[HMEZ+05] HELAL, Sumi ; MANN, William ; EL-ZABADANI, Hicham ; KING, Jeffrey ; KADDOURA, Youssef ; JANSEN, Erwin: The Gator Tech Smart House: A programmable pervasive space. In: *Computer* (2005), p. 64–74 24

[HRD10] HASSINE, Jameleddine ; RILLING, Juergen ; DSSOULI, Rachida: An evaluation of timed scenario notations. In: *The Journal of Systems and Software* 83 (2010), p. 326–350 33, 34

[HSM07] HUNTER, Audrey ; SAYERS, Heather ; MCDAID, Liam: An Evolvable Computer Interface for Elderly Users. In: *Workshop "Supporting Human Memory with Interactive Systems", HCI Conference*. Lancaster, UK, 2007, p. 29–32 4, 5

[Hsu04] HSU, William H.: Genetic wrappers for feature selection in decision tree induction and variable ordering in Bayesian network structure learning. In: *Information Sciences* Bd. 163, 2004, p. 103–122 55

[KD05] KOK, Stanley ; DOMINGOS, Pedro: Learning the Structure of Markov Logic Networks. In: *Proceedings of the 22 nd International Conference on Machine Learning*, 2005, p. 441–448 55

[LBVD09] LANG, Roland ; BRUCKNER, Dietmar ; VELIK, Rosemarie ; DEUTSCH, Tobias: Scenario Recognition in Modern Building Automation. In: *International Journal of Intelligent Systems and Technologies* (2009), p. 9, 4–1–5 24, 34, 35, 36

[LC09] LI, Haifeng ; CHEN, Hong: Mining non-derivable frequent itemsets over data stream. In: *Data & Knowledge Engineering* 68 (2009), p. 481–498 24, 25

[Ler92] LEROUX, Brian G.: Maximum-likelihood estimation for hidden Markov models. In: *Stochastic Processes and their Applications* 40 (1992), p. 127–143 31

[Li05] LI, Yujian: Hidden Markov models with states depending on observations. In: *Pattern Recognition Letters* 26 (2005), p. 977–984 31

[Li09] LI, Hua-Fu: Interactive mining of top-K frequent closed itemsets from data streams. In: *Expert Systems with Applications* 36 (2009), p. 10779–10788 24, 25

[LKT08] LIU, Li ; KANTARCIOGLU, Murat ; THURAISINGHAM, Bhavani: The applicability of the perturbation based privacy preserving data mining for real-world data. In: *Data & Knowledge Engineering* 65 (2008), p. 5–21 24, 25

[LWV07] LÜHR, Sebastian ; WEST, Geoff ; VENKATESH, Svetha: Recognition of emergent human behaviour in a smart home: A data mining approach. In: *Pervasive and Mobile Computing* 3 (2007), p. 95–116 36

[MA98] DE MANTARAS, Ramon L. ; ARMENGOL, Eva: Machine learning from examples: Inductive and lazy methods. In: *Data & Knowledge Engineering* 25 (1998), p. 99–123 28

[Mar09] MARSLAND, Stephen: Using habituation in machine learning. In: *Neurobiology of*

Learning and Memory 92 (2009), p. 260–266 27

[MH05] Ma, Jinwen ; He, Qicai: A Dynamic Merge-or-Split Learning Algorithm on Gaussian Mixture for Automated Model Selection. In: *Springer-Verlag Berlin Heidelberg* IDEAL (2005), p. 203–210 29, 40

[MMCG09] Magkos, Emmanouil ; Maragoudakis, Manolis ; Chrissikopoulos, Vassilis ; Gritzalis, Stefanos: Accurate and large-scale privacy-preserving data mining using the election paradigm. In: *Data & Knowledge Engineering* 68 (2009), p. 1224–1236 25

[MT93] Meyn, S.P. ; Tweedie, R.L.: Markov Chains and Stochastic Stability. In: *Springer-Verlag* (1993), p. 558 32, 49, 52

[MVWG07] Moncrieff, Simon ; Venkatesh, Svetha ; West, Geoff ; Greenhill, Stewart: Multi-modal emotive computing in a smart house environment. In: *Pervasive and Mobile Computing* 3 (2007), p. 74–94 20

[MZB07] Mozina, Martin ; Zabkar, Jure ; Bratko, Ivan: Argument based machine learning. In: *Artificial Intelligence* 171 (2007), p. 922–937 28

[Nil98] Nilsson, Nils J.: *Introduction to Machine Learning*. Stanford University, 1998. – 179 p 26

[NM06] Nascimento, Jacinto C. ; Marques, Jorgs S.: Performance Evaluation of Object Detection Algorithms for Video Surveillance. In: *IEEE Transactions on Multimedia* 8, NO, 4 (2006), p. 761–774 29

[PCH08] Picus, C. ; Cambrini, L. ; Herzner, W.: Boltzmann Machine Topology Learning for Distributed Sensor Networks Using Loopy Belief Propagation Inference. In: *2008 Seventh International Conference on Machine Learning and Applications* (2008), p. 347 65

[PDHP07] Pratl, Gerhard ; Dietrich, Dietmar ; Hancks, Gerhard P. ; Penzhorn, Walter T.: A New Model for Autonomous, Networked Control System. In: *IEEE Transaction on Industrial Informatics* 3 (2007), p. 21–32 17

[PLD05] Pratl, Gerhard ; Lorenz, Brigitte ; Dietrich, Dietmar: The Artificial Recognition System (ARS): New Concept for Building Automation. In: *6th IFAC International Conference on Fiedbus System and their Application*, 2005, p. 48–55 14, 17

[PP05] Pratl, Gerhard ; Palensky, Peter: Project ARS - The next step towards an intelligent environment. In: *Proceedings of the IEEE International Workshop on Intelligent Environments*, 2005, p. 55–62 14

[Pra06] Pratl, Gerhard: *Processing and Symbolization of Ambient Sensor Data*, Vienna University of Technology, PhD thesis, 2006. – 129 p 17

[Rab89] Rabiner, Lawrence R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognitio. In: *proceedings of the IEEE* Vol. 77, NO. 2 (1989), p. 257–286 32, 33, 49, 52

[RG97] Richardson, Sylvia ; Green, Peter J.: On Bayesian Analysis of Mixtures with an Unknown Number of Components. In: *J. R. Statist. Soc. B* 59, No. 4 (1997), p. 731–792 29, 40

[RKR+08] Reynolds, S.M. ; Käll, L. ; Riffle, M.E. ; Bilmes, J.A. ; Noble, W.S.: Transmembrane Topology and Signal peptide Prediction Using Dynamic Bayesian Networks. In: *PLoS Computational Biology* 4, Issue 11 (2008), p. 14 49

[ROTZ08] Rong, Hai-Jun ; Ong, Yew-Soon ; Tan, Ah-Hwee ; Zhu, Zexuan: A fast pruned-extreme learning machine for classification problem. In: *Neurocomputing* 72 (2008), p. 359–366 28

[SGU09] Şaykol, Ediz ; Güdükbay, Uğur ; Özgür Ulusoy: Scenario-based query processing for video-surveillance archives. In: *Engineering Applications of Artificial Intelligence* (2009), p. 15 33, 34, 36, 37

[SHJ96] Smyth, Padhraic ; Heckerman, David ; Jordan, Michael I.: Probabilistic Independence Networks for Hidden Markov Probability Models. (1996), p. 37 49

[SHP08] Swears, Eran ; Hoogs, Anthony ; Perera, A. G. A.: HMM-Based Clustering for Learning Motion Patterns in Surveillance Video. (2008), p. 1–2 29

[SK95] Sin, Bongkee ; Kim, Jin H.: Nonstationary hidden Markov model. In: *Signal Processing* 46 (1995), p. 31–46 31, 32

[SO93] Stolcke, A. ; Omohundro, S.: Hidden Markov Model Induction by Bayesian Model Merging. In: *Neural Information Processing System* 5 (1993), p. 8 49

[TCC09] Tai, Allen H. ; Ching, Wai-Ki ; Chan, L. Y.: Detection of machine failure: Hidden Markov Model approach. In: *Computers & Industrial Engineering* 57 (2009), p. 608–619 32

[TIL04] Tapia, Emmanuel M. ; Intille, Stephen S. ; Larson, Kent: Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In: *Proceedings of the Second International Conference on Pervasive Computing*, 2004, p. 158–175 3, 4

[UNGH98] Ueda, Naonori ; Nakano, Ryohei ; Ghahramani, Zoubin ; Hinton, Geoffery E.: Split and merge EM algorithm for improving Gaussian mixture density estimates. In: *0-7803-5060-X/98, IEEE* (1998), p. 274–283 29

[UNGH00] Ueda, N. ; Nakano, R. ; Ghahramani, Z. ; Hinton, G.: SMEM algorithm for mixture models. In: *Neural Computation* 12 (2000), p. 2109–2128 40

[VL02] Vlassis, N. ; Likas, A.: A Greedy EM Algorithm for Gaussian Mixture Learning. In: *Neural Processing Letters* 15 (2002), p. 77–87 40

[WLH06] Wang, Yida ; Lim, Ee-Peng ; Hwang, San-Yih: Efficient mining of group patterns from user movement data. In: *Data & Knowledge Engineering* 57 (2006), p. 240–282 24, 25

[WSB+09] Weimar, U. ; Simpson, R. ; Barsan, N. ; Heine, T. ; Simmendinger, W. ; Malfatti, M. ; Margesin, B. ; Gonzo, L. ; Grassi, M. ; Lombardi, A. ; Malcovati, P. ; Leone, A. ; Diraco, G. ; Siciliano, P. ; v. Sicard, O. ; Pohle, R. ; Fleischer, M. ; Redaelli, A. ; Giacosi, A. ; Bonassi, C.: Microsystem Technology for Ambient Assisted Living (AAL). In: *Procedia Chemistry* 1 (2009), p. 710–713 20, 21

[XFJ06] Xiong, Guanglei ; Feng, Chao ; Ji, Liang: Dynamical Gaussian mixture model for tracking elliptical living objects. In: *Pattern Recognition Letters* 27 (2006), p. 838–842 29

[YB09a] Yin, Guoqing ; Bruckner, Dietmar: Fast Learning Algorithm for Gaussian Models to Analyze Video Objects with Parameter Size. In: *Proceedings of the IEEE ETFA*. Mallorca, Spain, 2009, p. 4 4, 7

[YB09b] Yin, Guoqing ; Bruckner, Dietmar: Gaussian Mixture Models and Split-Merge Algorithm for Parameter Analysis of Tracked Video Objects. In: *the 35th Annual Conference of the IEEE Industrial Electronics Society*. Porto, Portugal, 2009, p. 4155–4158 4, 7, 40

[YB09c] Yin, Guoqing ; Bruckner, Dietmar: Gaussian Models and Fast Learning Algorithm for Persistence Analysis of Tracked Video Objects. In: *Proceedings of the IEEE HSI*. Catania, Italy, 2009, p. 60–63 4, 7

[YB10a] Yin, Guoqing ; Bruckner, Dietmar: Daily Activity Learning from Motion Detector Data for Ambient Assisted Living. In: *Proceedings of the 3rd International*

Conference on Human System Interaction (IEEE HSI), Rzeszow, Poland, 2010, p. 6 21

[YB10b] Yin, Guoqing ; Bruckner, Dietmar: Data Analyzing and Daily Activity Learning with Hidden Markov Model. In: *Proceedings of the The 2010 International Conference on Computer Application and System Modeling, Tai Yuan, China*, 2010, p. 3–7 21

[YB10c] Yin, Guoqing ; Bruckner, Dietmar: Split-Merge Algorithm and Gaussian Mixture Models for AAL. In: *Proceedings of the 2010 19th IEEE International Symposium on Industrial Electronics (ISIE) (invited), Bari, Italy*, 2010, p. 4 21, 30, 41

[YB11a] Yin, Guoqing ; Bruckner, Dietmar: Activity Analyzing with Multisensor Data Correlation. In: *Proceedings of the 2011 International Conference on Automation and Robotics (ICAR 2011), Dubai, UAE*, 2011, p. 529–534 21, 22

[YB11b] Yin, Guoqing ; Bruckner, Dietmar: Build User Daily Activity Model and Model Structure Changing. In: *Proceedings of IEEE AFRICON 2011, Zambia, Africa*, 2011, p. 5 21

[YB11c] Yin, Guoqing ; Bruckner, Dietmar: Daily Activity Model for Ambient Assisted Living. In: *Technological Innovation for Sustainability, Caparica, Lisbon, Portugal*, 2011, p. 197–204 21

[YBZ09] Yin, Guoqing ; Bruckner, Dietmar ; Zucker, Gerhard: Statistical Modeling of Video Object's Behavior for Improved Object Tracking in Visual Surveillance. In: *Proceedings of the 2009 IEEE International Conference on Africon*. Nairobi, Kenya, 2009, p. 6 4, 7, 27

[YK03] Yu, Shun-Zheng ; Kobayashi, Hisashi: A hidden semi-Markov model with missing data and multiple observation sequences for mobility tracking. In: *Signal Processing* 83 (2003), p. 235–250 32

[YRA98] Yang, Boo-Ho ; Rhee, Sokwoo ; Asada, Haruhiko H.: A Twenty-Four Hour Tele-Nursing System Using a Ring Sensor. In: *Proc. of 1998 Int. Conf. on Robotics and Automation*. Leuven, Belgium, 1998, p. 387–392 4, 5

[ZCSC03] Zhang, Zhihua ; Chen, Chibiao ; Sun, Jian ; Chan, Kap L.: EM algorithms for Gaussian mixture with split-and-merge operation. In: *Pattern Recognition* 36 (2003), p. 1973–1983 29, 40

[ZL09] Zhu, Songhao ; Liu, Yuncai: Automatic scene detection for advanced story retrieval. In: *Expert Systems with Applications* 36 (2009), p. 5976–5986 34, 35, 36

Internet References

[AAL] Ambient Assisted Living. – http://www.aal-europe.eu/about-us/aal-association

[ACR] Austrian Country Report. – http://www.aal-europe.eu/Published/reports-etc/CRAustria.pdf

[AEU] Ambient Assisted Living Europe. – http://www.aal-europe.eu/about-us

[AWH] Project Aware Home. – http://awarehome.imtc.gatech.edu/

[BAH] Company Barrett Hand Technology Inc. – http://www.barrett.com/robot/index.htm

[FZI] Project FZI Living LAB. – http://aal.fzi.de/

[HAM] Project In-HAM vzw. – www.in-ham.be

[HIS] Project Inhaus. – http://www.inhaus.fraunhofer.de/

[HLB] Project HomeLab. – http://www.newscenter.philips.com/main/standard/about/news/press/archive/2002/article-2156.wpd

[ICT] Institute of Computer Technology. – https://www.ict.tuwien.ac.at/home

[LTM] Project Living Tomorrow. – www.livingtomorrow.com

[RTC] real time control technology with human electroencephalography. – http://www.sciencedaily.com/releases/2009/06/090629101848.htm

[SSP] Project SSP (senior social platform). – http://ssp.ceit.at/

All internet references accessed in December, 2011.

Curriculum Vitae

Dipl.-Ing. Guo Qing Yin

Personal Information:

Birth	1.10.1972, Hebei, China
Marital Status	Single
Nationality	China

Education:

1992-1995	Beijing Institute of Civil Engineering and Architecture
2000-2001	German courses in Technical University Kaiserslautern
2001-2005	Technical University Kaiserslautern
2006-2012	Ph.D student in Technical University Vienna

Professional Experience:

2007-2012	Project assistant in Technical University Vienna European project SENSE: video data analysis, modeling, image processing. Based on the video data parameters (frame number, ID, position, height and width of the objects) to build objects size (duration, direction, velocity) models in scenes. According these models to detect unusual behavior (e.g. people "lurking" in an area) and situations (e.g. baggage left unattended). These models used for video surveillance system which installed in an airport. Austria FFG project ATTEND: Ambient Assisted Living, sensor data analysis, behavior model of elderly people. Based on the data from non-intrusive sensors (motion detector, door contactor, accelerometer, without camera and microphone) which installed in the living environment of the elderly people to build daily behavior model of the user. Based on the models in case of unusual activities happened the system will send alarm signal to caregiver.

Publications:

1. Yin, Guoqing ; Bruckner, Dietmar: Activity Analysis with Hidden Markov Model for Ambient Assisted Living. In: *Proceedings of the International Conference on Applied and Theoretical Information Systems Research (ATISR 2012), Taipei, Taiwan*, 2012, p. 1–12

2. Yin, Guoqing ; Bruckner, Dietmar: Activity Analyzing with Multisensor Data Correlation. In: *Proceedings of the 2011 International Conference on Automation and Robotics (ICAR 2011), Dubai, UAE*, 2011, p. 529–534

3. Yin, Guoqing ; Bruckner, Dietmar: Build User Daily Activity Model and Model Structure Changing. In: *Proceedings of IEEE AFRICON 2011, Zambia, Africa*, 2011, p. 5

4. Yin, Guoqing ; Bruckner, Dietmar: Daily Activity Model for Ambient Assisted Living. In: *Technological Innovation for Sustainability, Caparica, Lisbon, Portugal*, 2011, p. 197–204

5. Yin, Guoqing ; Bruckner, Dietmar: Daily Activity Learning from Motion Detector Data for Ambient Assisted Living. In: *Proceedings of the 3rd International Conference on Human System Interaction (IEEE HSI), Rzeszow, Poland*, 2010, p. 6

6. Yin, Guoqing ; Bruckner, Dietmar: Data Analyzing and Daily Activity Learning with Hidden Markov Model. In: *Proceedings of the 2010 International Conference on Computer Application and System Modeling, Tai Yuan, China*, 2010, p. 3–7

7. Yin, Guoqing ; Bruckner, Dietmar: Split-Merge Algorithm and Gaussian Mixture Models for AAL. In: *Proceedings of the 2010 19th IEEE International Symposium on Industrial Electronics (ISIE) (invited), Bari, Italy*, 2010, p. 4

8. Yin, Guoqing ; Bruckner, Dietmar: Fast Learning Algorithm for Gaussian Models to Analyze Video Objects with Parameter Size. In: *Proceedings of the IEEE ETFA*. Mallorca, Spain, 2009, p. 4

9. Yin, Guoqing ; Bruckner, Dietmar: Gaussian Mixture Models and Split-Merge Algorithm for Parameter Analysis of Tracked Video Objects. In: *the 35th Annual Conference of the IEEE Industrial Electronics Society*. Porto, Portugal, 2009, p. 4155–4158

10. Yin, Guoqing ; Bruckner, Dietmar: Gaussian Models and Fast Learning Algorithm for Persistence Analysis of Tracked Video Objects. In: *Proceedings of the IEEE HSI*. Catania, Italy, 2009, p. 60–63

11. Yin, Guoqing ; Bruckner, Dietmar ; Zucker, Gerhard: Statistical Modeling of Video Object's Behavior for Improved Object Tracking in Visual Surveillance. In: *Proceedings of the 2009 IEEE International Conference on Africon*. Nairobi, Kenya, 2009, p. 6

12. Yin, Guoqing ; Bruckner, Dietmar: Sensor Fusion and Hidden Markov Model for Project ATTEND, Poster: Ambient Assisted Living (AAL) Forum 2011, Lecce, 26.09.2011–28.09.2011.

13. Yin, Guoqing ; Bruckner, Dietmar: Hidden Markov Model and Split-Merge Algorithm for Project ATTEND, Poster: Ambient Assisted Living (AAL) Joint Programme 2010, Odense, 15.09.2010–17.09.2010.

14. Yin, Guoqing ; Bruckner, Dietmar: Data Analyzing with Gaussian Mixture Models and Split-Merge Algorithm for AAL, Poster: Conference on Ambient Assisted Living, Technology and Innovation for Ageing Well, AALIANCE, Malaga; 11.03.2010–12.03.2010.

15. Bruckner Dietmar ; Yin Guoqing: Project ATTEND — AdapTive scenario recogniTion for Emergency and Need Detection, Poster: Ambient Assisted Living (AAL) Forum, Vienna, 29.09.2009–01.10.2009.