Sarah Fallmann

Human activity pattern recognition based on continuous data from a body worn sensor placed on the hand wrist using Hidden Markov models

Technische Universität Wien

Institut für Analysis und Scientific Computing
Head: Univ.Prof. Jens Markus Melenk, PhD.

1st Supervisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker
2nd Supervisor: Jonathan Steinhart, BS MA

Vienna, September 2015

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....08.09.2015.....
(date)

.....
(signature)

ACKNOWLEDGMENT

I would like to express my appreciation to AIT Austrian Institute of Technology for enabling this diploma thesis.

The completion of this thesis would not have been possible without the assistance of my supervisors Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker and Jonathan Steinhart, MA. Thank you for your support and suggestions.

Special thanks go to Dipl.-Ing. Dr. Johannes Kropf, who assisted and encouraged me in all matters concerning the diploma thesis.

Abstract

The aim of the thesis is to combine discrete and continuous data in an algorithm to detect complex activity patterns such as tooth brushing, food preparation or household work. These patterns are of highly complex nature, meaning they consist of many sub-activities.
Food preparation, for instance consists of sub-activities like 'take pans out of the cupboard','take food out of the refrigerator', 'cutting', 'cooking' and others. The complicated parts of this task are, that the food preparation not only differs in the order of the sub-activities, but also in the food which is prepared. Envision the preparation of a three-course menu in comparison to preparing a breakfast. This two activities differ a lot in needed duration and sub-activities. The human activity recognition system built in this thesis can handle these differences.

The InvenSense MotionFit^{TM} Software Development Kit (SDK) is used to record data with the MPU-9150 sensor. The MPU-9150 sensor is a nine-axis MotionTracking device, which is optimized for those kind of applications in this thesis and is normally used in mobile devices.[20] The wearable sensor MPU-9150 is able to send accelerometer and gyrometer data, which are later used to detect human activities in real environments.
The frequency of the annotated data is 50Hz in all experiments. The processed activities are 'Comb hair','Wash face', 'Wash hands', 'Tooth brushing', 'Make bed', 'Change clothes', 'Put roller blinds up/down', 'Prepare food', 'Eat' and 'Open/close window'. Inbetween this activities a 'NULL'-class is performed, which describes the preparation for the next activity or the closure of the previous activity. This raw data are preprocessed via a shifted window and different features. In this thesis the window length is set to 50, equal the sampling frequency and the shift is accomplished with an 50% overlap. The used features are mean, variance, correlation and fast Fourier transformation based features. The fast Fourier transformation bases features are spectral entropy and energy.

The pattern recognition is done in MATLAB using the PMTK3 toolbox from Murphy *et al.* [22] with real annotated data. The used classification algorithms are from supervised learning structure, meaning that they need labeled data for training. This circumstances are fulfilled for the data used in this thesis. The classification algorithms that are used during the experimentation are continuous Hidden Markov Model (cHMM) and k-nearest-neighbors (kNN) classification. The classification methods are described in detail and a comparison is done to discuss the differences between the results. In the end the cHMM leads to the more accurate outcomes in comparison to the kNN classifier.

Daily activity detection works well in the context of ambient assisted living. It can be concluded that, the cHMM is the most proper method and comes to the best solutions. In contrast the kNN classification is much worse, because of its simplifying assumption. Due to that the kNN classifier is not the best classifier to use, but for the simpleness, acceptable results can be expected.

After validation of the model the different features combinations are compared to each other to find the most suitable combination. Other experiments focus on different training and test sets, the best number of sensors, the impact of filters, the impact of activity division and the application of discrete and continuous data. The experiments show that the accelerometer data on their own lead to one of the best results, whereas the filters as well as activity division do not lead to a qualitative improvement.
The combination of discrete and continuous data improves the results a lot and leads to different activities with highest recall and precision. The precision for 'Wash hands' is with a value of 100% the best in the continuous data case and 'Tooth brushing' in the combined case, also with 100%. 'Eat' has one of the best recall values with 97.13% in the continuous case, whereas the improvement in the combined case can be seen on the recall value 100% for 'Wash hands', 'Make bed', 'Put blinds up/down', 'Prepare food', 'Eat' and 'Open/close window'. Overall the system leads to reasonable outputs even with a relative small dataset.

Zusammenfassung

Das Ziel dieser Arbeit ist kontinuierliche und diskrete Daten in einem Algorithmus zu vereinen um komplexe Aktivitäten-Muster zu erkennen, wie Zähne putzen, Zubereitung von Essen oder Hausarbeit. Diese Muster sind oft sehr komplex, da sie aus vielen Untermustern bestehen.
Das Zubereiten von Essen zum Beispiel, besteht aus Untermustern wie 'Töpfe aus dem Schrank nehmen', 'Essen aus dem Kühlschrank nehmen', 'Schneiden', 'Kochen' und andere. Der schwierige Teil dieser Aufgabe ist, dass die Essenszubereitung sich nicht nur in der Reihenfolge der Untermuster unterscheidet, sondern auch in den unterschiedlichen Speisen die zubereitet werden. Man stelle sich den Unterschied zwischen der Vorbereitung eines Drei-Gänge-Menüs zu einem Frühstück vor. Diese beiden Aktivitäten unterscheiden sich enorm in der Vorbereitungsdauer und den Untermustern. Das Erkennungssystem welches innerhalb dieser Arbeit konstruiert wurde und menschliche Aktivitäten erkennt, kann mit dieser Art von Unterschieden umgehen.

Das InvenSense MotionFit™ Software Development Kit (SDK) wird verwendet um die MPU-9150 Sensordaten aufzuzeichnen. Der MPU-9150 Sensor ist ein neun-achsiges MotionTracking Gerät, welches für diese Art von Anwendungen, wie in dieser Arbeit benötigt, optimiert wurde. Dieser tragbare Sensor kann Beschleunigungssensor- und Gyrometer-Daten senden, welche später herangezogen werden um menschliche Aktivitäten in realen Umgebungen zu erkennen.
Die Frequenz der annotierten Daten ist in allen Experimenten auf 50Hz gesetzt worden. Die aufgenommenen Aktivitäten sind 'Haare kämmen', 'Gesicht waschen', 'Hände waschen', 'Zähne putzen', 'Das Bett machen', 'Kleidung wechseln', 'Rollos rauf/ runter ziehen', 'Essen zubereiten', 'Essen' und 'Fenster schließen/ öffnen'. Zwischen diesen Aktivitäten wird eine 'NULL'-Klasse durchgeführt, welche die Vorbereitung bzw. Nachbereitung der nächsten bzw. vorigen Aktivität beschreibt. Diese rohen Daten werden mithilfe eines verschiebbaren Fensters und verschiedener Features vorverarbeitet. In dieser Arbeit beträgt die Fensterlänge 50, genauso wie die Frequenz beim Aufnehmen. Die Verschiebung des Fensters wird mit einer 50%-Überlappung durchgeführt. Die verwendeten Features sind Mittelwert, Varianz, Korrelation und die auf schnelle Fourier-Transformation beruhenden Features, spektrale Entropie und Energie.

Die Mustererkennung wird mit MATLAB und der von Murphy *et al.* zur Verfügung gestellten Toolbox PMTK3 [22] bewerkstelligt. Die verwendeten Klassifikations-Algorithmen sind supervised Lernmethoden, dies bedeutet, sie brauchen gelabelte Daten für das Training. Dieser Umstand wird von den Daten in dieser Arbeit erfüllt. Die Klassifikations-Algorithmen die während der Experimente verwendet wer-

den sind einerseits kontinuierliche Hidden Markov Modelle (cHMM) und andererseits k-nächste-Nachbarn (kNN) Klassifikatoren. Die Klassifikationsmethoden werden im Detail beschrieben und ein Vergleich wird durchgeführt um Unterschiede zu erörtern. Die Experimente zeigen schlussendlich, dass das cHMM zu besseren Resultaten führt, im Gegensatz zu dem kNN Klassifikator.

Die Erkennung von Alltagsaktivitäten funktioniert gut im Zusammenhang mit Ambient Assisted Living. Es kann gefolgert werden, dass cHMM die geeignetste Methode ist und zu den besten Ergebnissen führt. Verhältnismäßig ist der kNN Klassifikator viel schlechter aufgrund seiner einfachen Annahme. Deshalb ist der kNN Klassifikator nicht der beste Klassifikator, aber trotz seiner Einfachheit können annehmbare Ergebnisse erwartet werden.

Nach der Validierung des Models sind verschiedene Features Kombinationen verglichen worden um die geeignetste Kombination zu finden. Andere Experimente konzentrieren sich auf die Verwendung von verschiedenen Training- und Test-Sets, die beste Anzahl von Sensoren, den Einfluss von Filtern, den Einfluss der Teilung von Aktivitäten und die Anwendung von diskreten und kontinuierlichen Daten. Die Experimente zeigen, dass die Bewegungssensor-Daten alleine, die besten Resultate liefert, während Filter und Teilung von Aktivitäten keine qualitative Verbesserung bringen.
Die Kombination von diskreten und kontinuierlichen Daten verbessert die Resultate erheblich und führt zu verschiedenen Aktivitäten mit bester Genauigkeit und Trefferquote. Die Genauigkeit für 'Hände waschen' ist mit 100% die beste Aktivität für kontinuierliche Daten und 'Zähne putzen' 100% für den kombinierten Fall. 'Essen' hat eine der besten Trefferquoten mit 97.13% im kontinuierlichen Fall, wohingegen der kombinierte Fall eine Verbesserung nach sich zieht mit 100% für 'Hände waschen', 'Das Bett machen', 'Rollos rauf/ runter ziehen', 'Essen zubereiten', 'Essen' und 'Fenster öffnen/ schließen'. Im Allgemeinen liefert das System vertretbare Resultate sogar mit einem relativ kleinen Datensatz.

Contents

1. Introduction

This thesis deals with activity pattern recognition based on continuous sensor data using the MPU9150 sensor. Activity recognition is currently a subject of intensive research, because of it's importance in many different fields. The motivation of this work in specific is the growing generation of older adults, and the need to provide them a secure and appropriate living standard. The continuous sensor data are used to recognize activities with basic machine learning algorithms. This algorithm analysis comprises the continuous Hidden Markov Model and the k-nearest-neighbors (kNN) classification. These classification algorithms are described, compared against each other and their different parameters are analyzed.

1.1. Motivation

The motivation of the thesis presented here is mainly driven by the demographic changes and the need for ambient assisted technologies to support a longer independent living of the older adults.[29]
The focus lies on caring for older adults, which are independent enough to live on their own, but desire to feel more secure within their four walls. The security factor comprises, among other things, the sending of help in case of an accident, warning from an open window or switch off stove before leaving the house for a longer period, inform to take adequate clothing during an walk and knowing preferences. This factors are treated in Ambient Assisted Living (AAL) technologies and Human activity recognition can handle and meet the demand of the older adults with those kind of needs.[39]

AAL has a wide range of applications and research topics and becomes more and more important in society. The AAL topics includes 'AAL at home', 'AAL on-the-move' over 'AAL at community' to 'AAL at work'.[39] This topics are from high importance for older adults. Due to the demographic change the mentioned needs concern a higher amount of people and will even increase in the next decades.

The demographic changes also lead to more people suffering from Alzheimer's and Parkinson's disease. For example, the challenge of the increasing number of dementia patients can be approached by Ambient Assisted Living Technologies like activity recognition, as some tasks of care givers can be eliminated or can be performed easier. This includes, among other things, sensors which control kitchen appliances like stoves, and guarantee the appropriate usage due to activity recognition.

1.1.1. Demographic Changes

The population follows the trend of longer life expectancy leading to a greater demand of caring for older adults. The average life expectancy for women living in Austria was in 2006 82.7 years and for men 77.1 years. A sixty year old woman has 2006 a further life expectancy of 24.9 years and for a sixty year old man it amounts 21.0 years. The life expectancy of people in older adult age has been increased in the last decades.

This trend is an advantage for men and women, meaning that the further life expectancy for sixty-year-old increased by 6.2 years for men and by 6.1 years for women since 1970. Due to the higher life expectancy the statistical probability to reach an older age has increased.
The probability of a new born girl to reach age 60 lies at 94% and for newborn boys at 89%. This values will increase till 2030 to 97% and 95%, respectively. The probability of reaching the age of 85 has increased from 21% to 52% between 1970 to 2006. If the positive trend of the last decades continues in diluted form a newborn girl in 2030 will reach at least the age of 85 with 73%.[38]
The share of people over 65 years on the whole population in Austria has been $14,9\%$ in 1990 and $18,2\%$ in 2013. This trend would lead to $23,6\%$ of the world population being at least 65 year old in 2030. The life expectancy of a newborn girl changed from 78.9 years to 83.6 years over the course from 1970 to 2013 and will increase to 86.7 years till 2030. For a newborn boy $72.2, 78.5$ and 82.3, respectively.[6]
In figure 1 the population pyramid of Austria from $1952, 2014$ and 2075 can be seen and shows the amount of different ages on the whole population. The trend in the figures shows a prognoses of a greater population growth with a higher percentage of older adults.[7]

Figure 1: Population Pyramid 1952 - 2014 - 2075 [7]

The increasing by aging population leads to the following challenges for society:

- A further increase in diseases like Alzheimer's and Parkinson's disease.
- A further increase in health care costs, that is the reason why the current model will be strained.
- shortage of care givers, meaning less professionals, leading to a responsibility of the family members resulting in emotional distress and physical health problems.
- dependency, there will be a rise in individuals needing care.
- larger impact on society, meaning it is not possible for society to pay for older adults to live in assisted living. Employers will have losses due to working family care givers absenteeism.

These facts and challenges underline the necessity of more Ambient Assisted Living technologies, enable older adults to life longer independently in their own home.[34]

1.1.2. Ambient Assisted Living

Ambient Assisted Living (AAL) can be defined in the following way:
»Cultivating the development of innovative products, services and systems for ageing well at home, in the community and at work, accordingly improving quality of life, autonomy, participation in social life, skills and reducing the costs of health and social care.«[39]

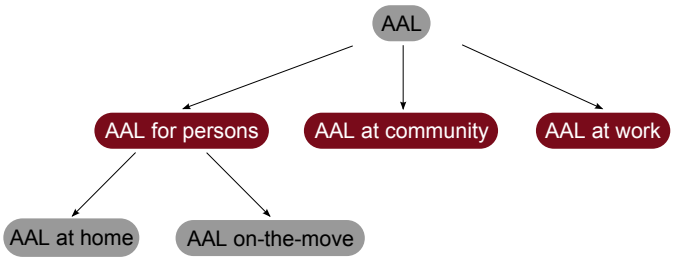

Figure 2: AAL classification

For AAL, a system is demanded which can be adapted to different living conditions like rehabilitation after a hospital treatment or degeneration and is useable in an easy way. The users of the system are primary older adults, but also care givers, who might use services from their own home.[39]
Another important aspect, which must be taken into account, is the variety of locations of assistance. For example the type of living locations[1], people on the move[2], visiting locations[3] or location of people[4].

[1] like family home, home for seniors, nursing home, etc.
[2] public transport, walking, car driving, etc.
[3] homes of families, workplaces, theater, etc.
[4] family, neighbors professional carers

The AAL support can be classified in

- '*Ageing well at home*': Technologies enabling a healthier, higher quality of life, with a high degree of independence and autonomy.
- '*Ageing well in the community*': Technologies to stay socially active, oriented toward social networking and access to public services, reducing social isolation.
- '*Ageing well at work*': Technologies enabling to remain productive for a longer period.

This gives three main topics in AAL: *AAL for persons* over *AAL at community* to *AAL at work*. *AAL for persons* consists of *AAL at home* and *AAL on-the-move*. The field of *AAL for persons* contains ideas and technologies for health, rehabilitation and care. This technologies makes it easier to cope with impairments, monitor personal activity, ensure the security of older adults and reduce social isolation. *AAL in the community* covers, among other things, the social inclusion, entertainment, leisure and mobility aspect. *AAL at work* includes all aspects of supporting working-life, access to the working space and assuring environmental working conditions. Beside the rules regulating the architecture of the working environment, standards and laws define parameters related to other environmental factors like temperature, light and change of air and position of work-tools.[39] An overview of the AAL classification can be seen in figure 2.

1.1.3. Ambient Assisted Technologies

Ambient Assisted Technologies shall be embedded[5], distributed throughout the environment, personalized, adaptive and anticipating users' desires. The applications in the AAL field operate within the framework of the technologies sensing, reasoning, acting, communication and interaction.[39]
Sensing comprises sensors on-body and on-applications or in-applications for monitoring persons and ensure safety and security within the own home. This can, for example, be done by a gas sensor. One challenge is the creation of low power and sustainable sensors to minimize energy consumption.[39]
In this field Smart Homes are from high importance, actuators and sensors are used to assist older adults in their homes. For example, researchers at Washington State University within the CASAS [33] project tried to provide a noninvasive environment system to assist users with dementia. Another group of researchers at Georgia Tech within the Aware Home project [1] tried to help older adults with a variety of sensors such as smart floor sensors as well as assisting robots.[34]
The other huge topic in the field of AAL is wearable sensors. Many different sensors are used in this context like accelerometer, gyrometer and others. In this thesis accelerometer and gyrometer are used to detect human activities. Health status can be detected with indicators like blood glucose, blood pressure, and cardiac activity. This indicators can be measured through wearable

[5] noninvasive or invisible devices

sensors using techniques such as infrared sensing and optical sensing.[34]

Reasoning comprises the collection of data and further processing and analyzing it. The processing of the data is important to make it possible to handle imperfect information and imperfect temporal information. The analysis also comprises the transformation of data into knowledge. This can be divided into sections of emergency situations, activity of daily living, motion, physiological behavior and mental and physical constitutions.[39]
Human activity recognition is an important part in reasoning, therefore patterns must be found describing for example jogging, running or walking. In mobile activity recognition, data from accelerometer and gyrometer are stored and preprocessed with filters for high frequencies, for instance. After the preprocessing step features like mean, variance, fast Fourier transform (FFT) coefficients and others are extracted over a specific window, compare [31] and section 5. The classification predicts the activity to the features.[34]
In ambient activity recognition a whole complex of sensors is used to detect activities with the help of supervised algorithms, compare section 3. Hidden Markov Models are very popular for ambient activity recognition.[34]
For anomaly detection clustering-based methods, statistical methods and information theoretic methods are used, among others.[11, 34]

Acting makes an automated control through actuators and feedback possible, comprising home automation systems. Household robotic appliances, mechatronic robotic devices and posture and movement support are some examples.[39] Robotics is within these topic of acting and can assist with activities like feeding, dressing, preparing food, house keeping, social communication and new learning. Dusty robots [44] help older adults by grabbing and finding dropped objects from floor, for instance.[34]

Communication is the connection of sensors and actuators to reasoning systems. It includes for example 'a person moving from home to a public space with a vehicle'.[39]
Interaction of people with a service or a system is another research topic, containing initiative of the user, design processes, modality and connectivity.[39]

In the last part of this section some concrete applications are described. In health and activity monitoring, daily activities are monitored to draw conclusions to health status and to lower the burden of care givers. For example CASAS [33] monitors activities of daily living to see consistency in daily routines. Tele-health devices to monitor vital signs are also important, AMON [5] is such a wearable health monitoring system, for instance. Fall detection methods which are another great field in AAL can be wearable device based, ambiance sensor based or camera based.[26] Wearable fall detection uses accelerometer and gyrometer data to conclude on position and motion.[25, 43] Ambient fall detections uses pressure sensors and vibration detection.[3] The vision based fall detection uses video features.[12] Wandering prevention for

dementia patients are provided for example by OutCare, which informs the care givers when older adults don't walk on their daily routes.[42] Activity reminders for older adults are also from great importance for care givers, like reminding to take medication. One system for this demand is provided by Schulze named MEMOS.[36, 34]

1.2. Scope of the thesis

Human Activity recognition based on accelerometer and gyrometer sensor data is an important task in the field of AAL technologies. The thesis focuses on the recognition of complex daily activities like tooth brushing, dinner preparation, changing clothes and others. The annotated data is recorded with the MPU-9150 sensor and the InvenSense MotionFit^{TM} Software Development Kit (SDK). Supervised classification algorithms, namely continuous Hidden Markov Model (HMM) and k-nearest neighbors (kNN) classification, are used to detect the different activities. The main also lies on the comparison of the different parameters in the algorithms and the performance of these algorithms.

The current attempts to detect human behavior and activity can be classified by the type of the sensors used (1) body worn sensors, (2) video cameras and (3) domotic sensor networks. This work concentrates on body worn sensors.[15] The data gained from body worn sensors consist of accelerometer and gyrometer data.
In the literature the importance of continuous Hidden Markov Models is always highlighted and the results in this study show the good performance of this model.

1.3. Aim

The aim of this thesis is to compare and find out the best algorithm and parameter combinations to reach an appropriate result within the activity recognition process. Important in this regard are annotated data, size of dataset, but also the used sensors and combinations of continuous and discrete data.

2. Related Work

There are many research studies over human activity recognition in different settings. [8, 35, 18, 37, 23] Most of these works are based on acceleration data and tries to recognize daily activities like [8, 35, 18, 23].

Each study uses different sample frequencies for example Bao *et al.*[8] used a sample frequency of 76.25Hz, Ravi *et al.*[35] as well as Shoaib *et al.*[37] used 50Hz and Huynh[18] used different frequency for various experiments, for example raw acceleration data of 80Hz. To get a hint which frequency is accurate in the context of daily activity recognition, but still doesn't need too much memory, Khusainov *et al.*[23] compared different sampling rates and came to the conclusion that most of the body movements are contained in frequency below 20Hz. The classification accuracy also stays stable starting from 20Hz. Most of the accuracy changes stay between 10Hz and 20Hz.[23] The used sample frequency of 50Hz in this thesis is more than enough to get appropriate results, because of the explained reasons above.

In [8] mean, energy, frequency-domain entropy, and correlation features were extracted from the data and analyzed via decision table, instance-based learning, decision tree (C4.5), and Naive-Bayes classifiers included in Waikato Environment for Knowledge Analysis (WEKA)[14]. They analyzed twenty activities like eating, bicycling, reading, walking or running.

In [35] mean, standard deviation, energy and correlation are used to analyze the data with base-level classifiers like decision tables, decision trees (C4.5), k-nearest neighbors, SVM and Naive Bayes and meta-level classifiers like Boosting, Bagging, Plurality Voting, Stacking with Ordinary-Decision trees (ODTs) and Stacking with Meta-Decision trees (MDTs) available for WEKA. With these classifiers activities like standing, walking, running, climbing up/down stairs, sit-ups, vacuuming and brushing teeth are recognized without noise filtering of the data.

In [18] more complex activities are analyzed like shopping and doing housework. The features to classify the activities were mean, variance, energy, spectral entropy and discrete FFT coefficients and classifiers like Naive-Bayes are used.

In [37] a combination of accelerometer, gyrometer and magnetometer data from a smart-phone sensor is recorded and later six different activities with seven classifiers are analyzed using WEKA. Walk downstairs, running, sitting, standing, walk upstairs and walking are analyzed with classifiers like Naive-Bayes, Support Vector Machines, k-Nearest-Neighbors. Shoaib *et al.*[37] show that the combination of accelerometer and gyrometer completes the system and gives better results during physical activity recognition. The feature calculation is kept as simple as possible and only two time domain features are considered. They handle four dimensions

x, y, z and the magnitude $\sqrt{x^2 + y^2 + z^2}$ and compute mean and standard deviation.
On top of the work from Shoaib *et al.*[37] the frequency domain features are included within this research study and the aim is to find out the performance including more complex activities.

In [4] a feature dataset is provided which is suitable for evaluating the model in this thesis. They recorded eight activities from a group of 30 persons with a sampling rate from 50Hz, which equals the sampling rate of the recording in this thesis. The six activities consist of standing, sitting, laying down, walking, walking downstairs and upstairs The volunteers were instructed to perform all activities twice wearing a smartphone on their waste, that recorded the accelerometer and gyrometer data.
They calculated a bundle of 561 features and experimented mainly with a multi-class Support Vector Machine, showing an overall accuracy of 96% for test data consisting of 2947 patterns.[4] This results could not be reached with the model in this thesis, as the recorded data set is much smaller. The experiments in this study are based on feature data from [4] and only deals with a small part of the whole features.

This thesis is mainly based on the work from Bulling *et al.* [10]. They used body-worn accelerometer and gyrometer, to detect hand gestures which where commonly used in daily living activities.
They recorded 12 activities: opening a window, closing a window, watering a plant, turning book pages, drinking from a bottle, cutting with a knife, chopping with a knife, stirring in a bowl, forehand, backhand and smash. Additionally, no specific activities are performed inbetween so called 'NULL'-class. They used data from two persons with three sensors placed on their arms in different heights. They are placed on the top of the right hand, the outer side of the right lower and upper arm. The data comes from a three-axis accelerometer and a two-axis gyrometer, both recording annotated motion data at a sampling rate of 32Hz.[10]
The best results in [10] came from the combination of features mean, variance, zero crossing rate (ZCR) and mean crossing rate (MCR), this output fits quite well with the experience in this thesis. The combination of all positions upper arm, lower arm and hand leads to the most exact results.
In this thesis only one position is used with one sensor. This is why the results for the hand sensors are compared. One of the most appropriate algorithm methods is the HMM [10], also this work comes to this conclusion, but in this thesis only the comparison to one other model is performed.

3. Models

In this work supervised learning methods are used to analyze and classify daily activities using data recorded from wearable sensors. The applied supervised models are a continuous Hidden Markov Model (cHMM) and a k-nearest neighbors (kNN) classifier.
Annotated training data are used to learn the supervised models. The approach is to analyze the data with cHMM and kNN classifiers. By comparison unsupervised learning methods deploy a different procedure to handle data.

In supervised learning methods the inputs to be learned are labeled and the output is defined. This makes supervised learning a very accurate method if a sufficient large dataset is provided. The training data are used to learn the model, which means deriving of the model parameter values. Afterward the model should produce correct outputs for any new input.[13]
In comparison, unsupervised learning methods get input data for training, which is not labeled. This unsupervised methods intend to build statistical representations of the data. The representations can be correlations, clusters and similarities between data over statistical probabilities, for instance. Those can be used for decision making or prediction of future inputs.[13]
Supervised and unsupervised learning methods work with non-labeled data during application of the trained model. Therefore the difference between these models focuses on the learning of the model, meaning the training part. There is a distinction between generative and discriminative models within the learning methods.

3.1. Generative vs. Discriminative Models

The aim behind classification is to assign each input vector x to one of different categories, each of these categories are labels $\{y_i | i \in C\}$, where C is the set of categories. The input data x are called 'attributes' and the output data y the 'labels'. The joint probability distribution $P(x, y)$ gives a summary over the situation. To determinate the conditional probability $P(y \mid x)$ from a training dataset is the problem which must be solved and further gives a complete probabilistic description.[9]
In learning algorithms generative and discriminative models have to be distinguished. Generative models learn the joint probability $P(x, y)$ and the discriminative model learn the conditional probability $P(y \mid x)$.[28]

3.1.1. Generative Models

In generative models the problem of determining the density $P(y \mid x)$ for each label y is solved individually and the prior class probabilities $P(y)$ is inferred. The posterior label probabilities

$P(y \mid x)$ can be found by Bayes theorem

$$P(y|x) = \frac{P(x,y)}{P(x)} = \frac{P(x|y)P(y)}{P(x)} \tag{1}$$

for each label y. Decision theory helps to find the correct label to each input x with the conditional probability. Generative models model the distribution of input and output data employing implicit or explicit models.[9]

3.1.2. Discriminative Models

For discriminative models the approach is different, first the posterior label probability $P(y|x)$ is learned and with decision theory new x are assigned to one of the labels. Discriminative models model the conditional probabilities directly. An approach to calculate the conditional probability is to use the functional form of the model and to determine its parameters directly by means of using maximum likelihood. The aim is to maximize a likelihood function defined through the conditional distribution $P(y|x)$, representing discriminative training.[9]

3.1.3. Difference between generative and discriminative models

As already pointed out generative models learn the joint probability $P(x, y)$ and the discriminative model learns the conditional probability $P(y \mid x)$. In generative models the joint probability distribution is used to calculate the conditional probability via Bayes theorem. Hence the discriminative model is more straightforward than the generative model.

The generative model has a higher asymptotic error than the discriminative model, but approaches it faster. In some cases with large datasets the generative model is more accurate due to reaching the asymptotic error beforehand. In other cases the degenerative model is the right choice as it reaches a lower asymptotic error than the generative model.[28] Another advantage of the discriminative approach is that there are fewer parameters to be determined.[9]

3.2. Hidden Markov Model

In this section the construction of a Hidden Markov Model (HMM), which is a generative model, is discussed. This section contains the definition of Markov Chains, the extension to a HMM and modifications to get the finally used continuous HMM. In the thesis the continuous HMM (cHMM) provided by the PMTK3 toolbox from Murphy *et al.* [22] is used to analyze human activity recognition.

3.2.1. Discrete Markov Processes

Definition Markov Chain

A set of N states $S_1, \ldots, S_N$ is defined as a first order Markov Chain, if it holds the Markov Property for all time instants, meaning the present state depends only on the past state. Higher order Markov Chains increase the dependency towards a longer history.[41][32]

The state at any given time can be described in a Markov Chain as one state out of a set of states S_i. At each time instant a change of the state is performed, depending on the probabilities associated with each state. The time instants correlating with state changes are described as $t = \{1, 2, \ldots\}$. The state at a specific time t is depicted as x_t. A full probabilistic description in the case of an one-order Markov Property requires knowledge of the present and the past state, compare equation 2.[32]

$$P[x_t = S_j | x_{t-1} = S_i, x_{t-2} = S_k, \ldots] = P[x_t = S_j | x_{t-1} = S_i] \tag{2}$$

Important quantities in the Markov Chain theory are the state transition probabilities a_{ij}, which are conditional probabilities with specific properties, accomplishing equations 3-5.[32][41]

$$a_{ij} = P[x_t = S_j | x_{t-1} = S_i] \tag{3}$$

$$a_{ij} \geq 0 \tag{4}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{5}$$

Example: Markov Model

The stochastic process explained in the section above is called an observable Markov model. The output of the process is a set of states at each point in time, where each state depicts an observable event. An example of a Markov model with 3 states is presented in figure 3.

This Markov model could, for example, represent a model of the weather. For instance, once a day the weather is recorded as rainy (State 1), sunny (State 2) or cloudy (State 3) and out of

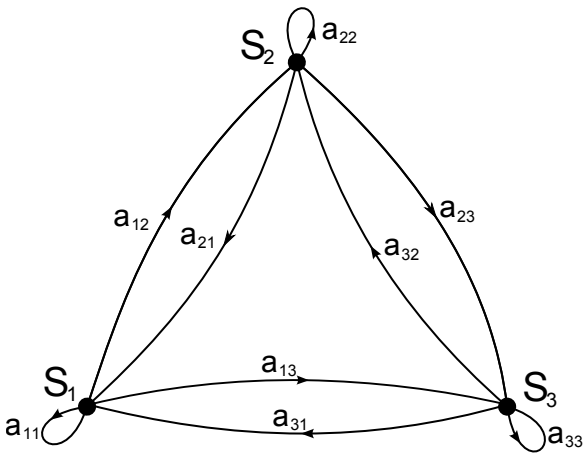

Figure 3: A Markov Model with 3 states S_1, S_2, S_3 and state transitions a_{ij}.[32]

this dataset a transition matrix can be built. This matrix explains the transitions from one state to the next state during one time instant. In this example the transitions probabilities are given by the following matrix:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.3 & 0.6 & 0.1 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

Assume the initial weather is sunny on day 1, meaning $\pi_2 = P[x_1 = S_2] = 1$, then the observation sequence $O = \{S_2, S_3, S_2, S_1, S_2, S_2, S_3, S_1\}$, namely sunny(initial state), cloudy, sunny, rainy, sunny, sunny, cloudy, rainy, can be expressed as

$$\begin{aligned} P(O|\text{Model}) &= P[S_2, S_3, S_2, S_1, S_2, S_2, S_3, S_1|\text{Model}] \\ &= P[S_2] \cdot P[S_3|S_2] \cdot P[S_2|S_3] \cdot P[S_1|S_2] \cdot P[S_2|S_1] \cdot P[S_2|S_2] \cdot P[S_3|S_2] \cdot P[S_1|S_3] \\ &= \pi_2 \cdot a_{23} \cdot a_{32} \cdot a_{21} \cdot a_{12} \cdot a_{22} \cdot a_{23} \cdot a_{31} \\ &= 1 \cdot (0.1)(0.2)(0.3)(0.3)(0.6)(0.1)(0.2) \\ &= 2.16 \times 10^{-5} \end{aligned}$$

Assume the model is in a known state, what is then the probability that it stays in exactly that state for d days. This problem can be answered by the probability of the observation sequence $O = \{\underbrace{S_i, S_i \ldots, S_i}_{d\times}, S_j\}$, where $S_j \neq S_i$, which is

$$P(O|\text{Model}, x_1 = S_i) = a_{ii}^{d-1}(1 - a_{ii}) = p_i(d).$$

The quantity $p_i(d)$ is called the probability density function with a duration d in state i. This duration density is characteristic of the state duration in a Markov chain. Now it is possible

to calculate the expected duration of staying in one specific state, with the property that this specific state is the starting state.

$$\begin{aligned}\overline{d}_i &= \sum_{d=1}^{\infty} d \cdot p_i(d) \\ &= \sum_{d=1}^{\infty} d \cdot (a_{ii})^{d-1}(1 - a_{ii}) \\ &= \frac{1}{1 - a_{ii}}\end{aligned}$$

Therefore the expected number of consecutive rainy days is 1.67, sunny days is 2.5 and cloudy days is 2.5.[32]

3.2.2. Extension to a Hidden Markov Model

In the section above only states which are observable events where considered. Now this model is extended to include the case where the observation is a probabilistic function of the state. This means a Hidden Markov Model (HMM) is a double embedded stochastic process, which has an underlying stochastic process that is hidden, but can be observed over another bundle of stochastic processes that produces the sequence of observations. For a clear understanding two examples are explained.[32]

Example: Coin Toss Model

Assume there are two rooms, in one room stands an observer in the other room an experimenter. The observer doesn't know what exactly happens in the other room, except from the task of the experimenter. This experimenter performs a coin tossing experiment, using a fixed number of coins. The observer has no information about the exact experimental setup and circumstances, but only over the result of the coin flips. According to that a sequence of 'hidden' coin toss experiments is conducted, which leads to an observation sequence consisting of a series of head and tails. For example the observation sequence can look like $O = O_1 O_2 \dots O_T = KKTTKTKTTTTKKK \dots K$, where K displays the experimental outcome 'head' and T displays the experimental outcome 'tail'.[32]

According to this scenario a HMM is constructed to explain the observation sequence. First the decision has to be made, what the states in the model correspond to, and afterwards the number of states in the model have to be defined.
If only a single coin is tossed, then the model could be explained trough a 2-state model, where the states correspond to 'head' and 'tail'. This is an observable model and the only unknown parameter which must be adjusted is the bias, namely the probability of saying heads.
The scenario can also be explained by a 2-state model, where each state corresponds to a different coin. Each state has probabilities for head and tail as well as for transitions from one state to

the next state. The selection of transition probabilities is done by a set of independent coin tosses, but can also be done trough some other probabilistic event. This model has 4 unknown parameters.
Another explanation can be given by a 3-state model, each state standing for one coin, where choosing among the three is based on some probabilistic event. This model has 9 unknown parameters.
To choose the best fitting model among the three, the greater degree of freedom seems to be an adequate characteristic. This assumption is theoretically true, but has some limitations. Assuming only one coin is used and the 3-state model is utilized. This choice is an underspecified system, as the actual physical events do not correspond to the model which is used.

Example: Urn and Ball Model

Assume that there are N transparent glass urns each filled with a lot of M distinct colored balls. The process of obtaining observations is done in the following way:

A independent person in the room chooses an initial urn, according to some random process, and draws a ball randomly. The color from a random ball in the urn is recorded as the observation and put back in the same urn. Another different urn is chosen, according to the random selection process associated with the present urn, and the step with the ball is repeated. The entire process generates a finite observation sequence, describing the colors of the drawn balls from the urns. This sequence is the observable output from a HMM.

The simplest HMM that describes the urn and ball model is one in which each state correspond to a specific urn, and where a color probability is defined for each state. The urn choice is dedicated by the state transition matrix of the HMM.[32]

3.2.3. Fundamentals of a Hidden Markov Model

A Hidden Markov Model(HMM) can be described by the following parameters:

1. N, number of states within the model
2. M, number of different observation symbols per state
3. A, transition matrix
4. B, the observation matrix
5. π, the initial distribution vector

The number of states in the model N, are hidden, but for many practical applications there is some physical significance related to states or set of states. Compare the urn and ball model, where the states correspond to the urns. In these models the states are connected to each

other in a way that each state can be reached by any other state. But also other interconnections are possible. The states are described as $S = \{S_1, S_2, \ldots, S_N\}$ and the state at time t as x_t.

The number of different observation symbols per state M is synonymous to the discrete alphabet size. The different observation symbols correspond to the physical output of the system which is to be modeled. For the urn and ball model the observation symbols were the different colors. The individual symbols are described as $V = \{v_1, v_2, \ldots, v_M\}$

The transition matrix A is displayed as

$$A = \{a_{ij}\} = P[x_{t+1} = S_j | x_t = S_i].$$

In an ergodic model each a_{ij} fulfills $a_{ij} > 0$, because a transition from each state to all other states is possible. In contrast models with other interconnections allow cases where $a_{ij} = 0$ for one or more entries in the transition matrix.

The observation matrix B is described as follows

$$B = \{b_j(k)\}$$

where $b_j(k)$ is

$$b_j(k) = P[v_k \text{ at } t | x_t = S_j], \quad 1 \leq j \leq N, \quad 1 \leq k \leq M.$$

The initial distribution vector π is described as follows

$$\pi = \{\pi_i\}$$

where π_i is

$$\pi_i = P[x_1 = S_i], \quad 1 \leq i \leq N$$

The HMM, described with the 5-tuple (N, M, A, B, π), can be used to generate an observation sequence

$$O = O_1 O_2 \ldots O_T$$

where each O_t is a symbol from V and T is the number of observations in the sequence. This concept follows the procedure:

1. First an initial state $x_1 = S_i$ is chosen, according to the initial distribution vector π.
2. Then t is set to 1.

3. According to the symbol probability distribution $b_i(k)$ in state S_i, $O_t = v_k$ is selected.
4. Afterward the transition step to a new state is executed $x_{t+1} = S_j$ pursuant to the state transition probability distribution a_{ij} for the state S_i.
5. t is set to $t + 1$ and the steps start by 3 again, as long as $t < T$. Otherwise the procedure terminates.

This procedure is used as a generator of observations and as a model for how a given observation sequence was generated by an appropriate HMM.[32]

A complete description of the characteristics of a HMM requires specification of the two parameter N and M, of observation symbols and of the probability measures A, B and π. This can be written in a compact form.[32]

$$\lambda = (A, B, \pi)$$

3.2.4. Continuous Hidden Markov Model

Most often HMM with discrete observations are discussed, but there exist cases where this approach does not lead to the right solution. This is the case when the observations are continuous vectors. Therefore the definition must be expanded and some restrictions must be made.

The observation symbol probability distribution is represented as a finite mixture of the form

$$b_j(O) = \sum_{m=1}^{M} c_{jm}\, \mathcal{N}[O, \mu_{jm}, U_{jm}], \quad 1 \leq j \leq N \tag{6}$$

where O is the vector being modeled, c_{jm} is the mixture coefficient for the mth mixture in state j and $\mathcal{N}$ is a log-concave or elliptically symmetric density, most often Gaussian, with mean vector μ_{jm} and covariance matrix U_{jm}. Each c_{jm} fulfills the following conditions

$$\sum_{m=1}^{M} c_{jm} = 1,\ 1 \leq j \leq N \tag{7}$$

$$c_{jm} \geq 0,\ 1 \leq j \leq N, \tag{8}$$

$$1 \leq m \leq M \tag{9}$$

so that the probabilistic density function is adequately normalized,

$$\int_{-\infty}^{\infty} b_j(x)\, dx = 1,\ 1 \leq j \leq N \tag{10}$$

The probability density function 6 can be used to approximate any finite, continuous density function, arbitrarily close. Therefore it can be applied to a wide area.[32] For more details about continuous Hidden Markov Models compare [32].

3.3. K-nearest neighbors classification

Nearest-neighbor classifiers follow a discriminative model and are based on the principle, that test tuple and training tuples, which are similar to each other, are compared. The training tuples are sets of n attributes, where each tuple displays a point in a n-dimensional space. Therefore all training tuples are represented as a n-dimensional pattern space. Assume the k-nearest neighbors (kNN) classifier gets an unknown test tuple, then the classifier searches for the k nearest tuples in the pattern space, which are the most closest to this test tuple. This k searched training tuples are called k nearest neighbors of the unknown test tuple.[16]
The vague term 'closeness' mentioned in the paragraph above is defined in terms of a distance metric. The Euclidean distance between two points $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{Y} = (y_1, y_2, \ldots, y_n)$ is defined as [16]

$$dist(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{11}$$

The value of each attribute is normalized before using equation 11, to prevent outweighing of attributes with smaller range at the begin by attributes with larger range. For example, min-max normalization can be used to transform a value m of an attribute A into a value $m' \in [0, 1]$ by calculating

$$m' = \frac{m - \min_A}{\max_A - \min_A},$$

where $\min_A$ and $\max_A$ describe the maximum and minimum value of A.[16]

The unknown test tuples are classified as the most common class in the k-nearest neighborhood. In the special case of $k = 1$, the test tuple gets the label of the training tuple which is the closest in the pattern space. Also numeric prediction is possible with a kNN classifier, as it returns then, the average value of the real labels in the k-nearest neighborhood.[16]

A missing value of an attribute A in tuple $\mathbf{X}$ or $\mathbf{Y}$ can be handled with the maximum possible difference. Because of normalization, the range of the attributes lie between 0 and 1.
A more difficult problem is the right choice of the parameter k. In this study the experimental approach is used. The classifier starts with $k = 1$, and with the test set, the error rate is calculated. Due to iterations, by incrementing k by one, the model with the lowest error rate is found and chosen. Typically k increases with increasing training set. If the size of the training set reaches infinity and $k = 1$, the error can't be bigger than twice the Bayes error rate, which depicts the theoretical minimum. If both size of trainings set and k reaches infinity, the error rate approaches the Bayes error rate.[16]
kNN classifiers are based on distance comparisons, and in practice assign equal weight to each attribute. That is the reason why they often result in low accuracy, due to noisy irrelevant

attributes. Also the Euclidean distance metric is a drawback, which can be dealt with by using other measurements like Manhattan distance.[16]
Another drawback of the kNN classifier is the commonly slow classification of test tuples. If T is a dataset of $|T|$ tuples and $k = 1$, then $O(|T|)$ comparisons are required to classify the test tuple. The comparisons can be reduced to $O(\log(|T|))$ by sorting and arranging the tuples into search trees. Parallel implementation reduces the runtime even to a constant $O(1)$, dependent on $|T|$.

In this thesis the most easiest kNN classifier provided by the PMTK3 toolbox from Murphy *et al.* [22] is used.

4. Sensor

4.1. Description

The InvenSense MotionFit^{TM} Software Development Kit (SDK) is used to record data. It enables sensor solutions designed for fitness, health and sports applications and consists of Hardware and Software components. It is able to track 10 degree-of-freedom, with the MPU-9150 (consisting of gyrometer, accelerometer and compass), a pressure sensor, a microcontroller and a Bluetooth radio module. The MPU-9150 is a nine-axis MotionTracking device optimized to fulfill the purposes for this sort of wearable sensor applications in this thesis.[20]

InvenseSense MPU-9150 is a single-chip Micro-Electro-Mechanical system (MEMS). The MEMS with nine-axis integrated circuit composes a three-dimensional accelerometer, three-dimensional gyrometer and a compass.
A nine-axis Motion Fusion inside the MPU is performed through an integrated Processor. A temperature sensor for temperature compensation for gyro bias is integrated in the MPU-9150 as well as a compass sensor, which detects terrestrial magnetism in all three axes by unifying magnetic sensors.[20] The SDK structure can be seen in figure 4.

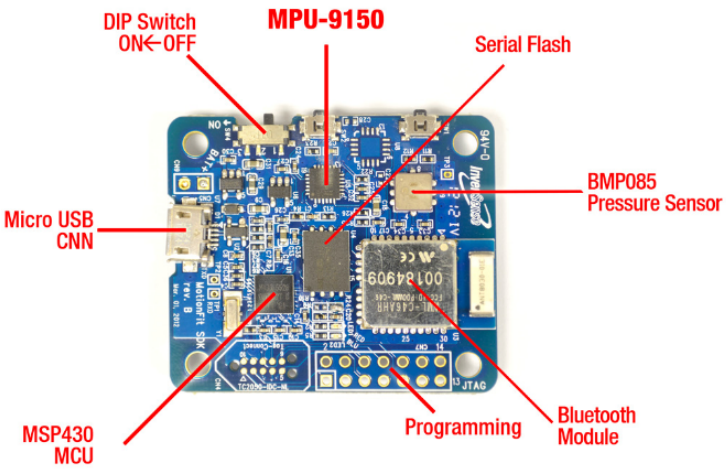

Figure 4: MPU 9150 MotionFit^{TM} Board [19]

Another important aspect which must be accounted for in the experiments is the orientation. The diagrams in figure 5 show the orientation of the axes of sensitivity and polarity of rotation. The pin 1 is symbolized by a black point on the upper side of the picture.[21]

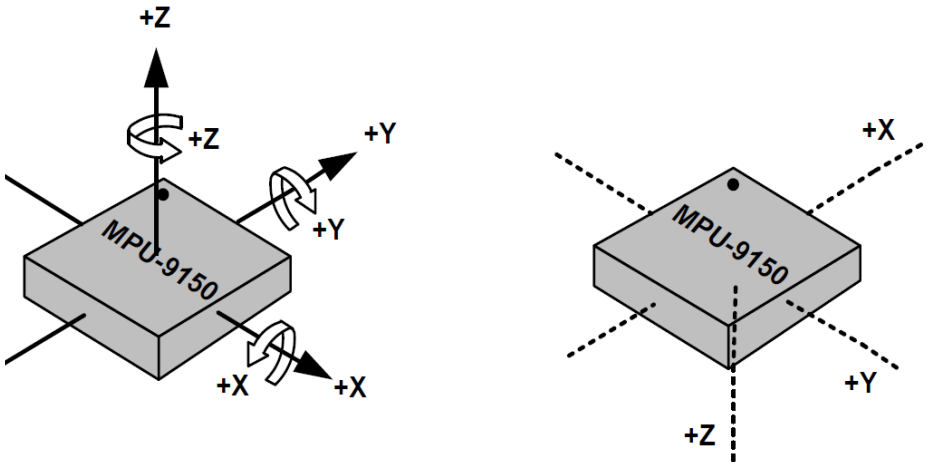

Figure 5: Left: Orientation of Axes of Sensitivity and Polarity of Rotation for Accelerometer and gyrometer.[21] Right: Orientation of axes of sensitivity for Compass.[21]

The MSP430 Microcontroller is from Texas Instruments (TI), supporting Code composer Studio (CCS) tool chain for software development. The device has a powerful 16-bit reduced instruction set computing (RISC), a Central Processing Unit (CPU), 16-bit registers and constant generators that lead to maximum code efficiency.[20]

The BMP085 Processor sensor is from Bosch and is based on piezo resistive technology for electromagnetic compatibility (EMC) robustness, high accuracy, linearity and long term stability. Another advantage is the low attitude noise of 0.25m at fat conversion.[20]

The Radio Module has a reaching area of up to 10m distance and low power consumption modes, in particular 50mA in active mode and 30μA in deep sleep mode.

Other components are the Serial Flash, Rechargeable Battery and the Micro USB connector. The 100mAh rechargeable battery provides more than 24 hours of activity logging and 4 hours of streaming time. The Micro USB controller is designed for charging the battery and wired connectivity.

In this thesis the accelerometer and gyrometer data are relevant and used for analysis. Therefore the basic details for accelerometer and gyrometer are described in the sections below.

4.2. Accelerometer

4.2.1. Accelerometer basics

Newton's second law of motion helps to describe the translational acceleration of a body. The acceleration is described with the following equation

$$F = ma. \tag{12}$$

Equation 12 describes that a force F on an object having mass m causes acceleration a of the object with respect to the system of inertia.
It is not practical to describe the acceleration of the whole object. Therefore only for a small part of the object the force is measured. This part is called seismic mass and consists in the primary element of an acceleration sensor, compare figure 6.
The proof mass of an accelerometer tends to resist changes in movement caused by acceleration along its sensitive axis. The tension of the spring balances the force acting on the mass. A further extension of the spring leads to a measure of the applied force proportional to acceleration. The total force can be described as

$$F = ma = mf + mg \tag{13}$$

where g denotes the gravitational constant and f is acceleration produced by forces except the gravitational field. An accelerometer is insensitive to gravitational acceleration so the output is proportional to the non-gravitational force to which the sensor is subjected.[27]

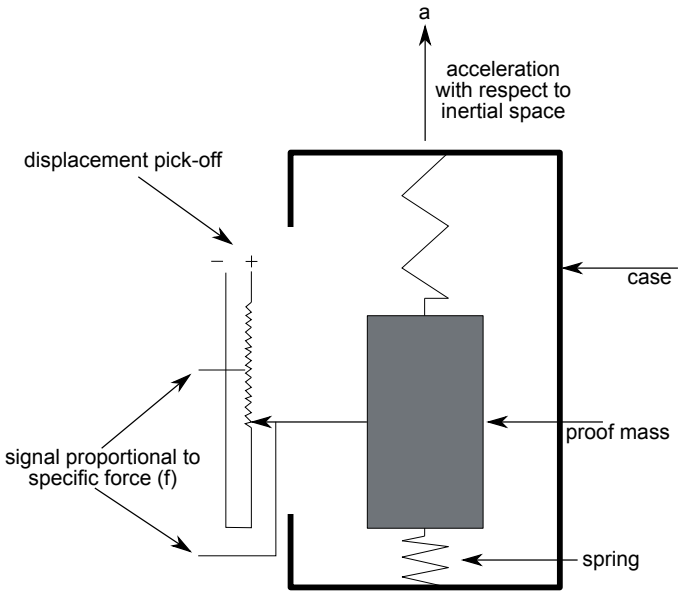

Figure 6: A simple accelerometer [27]

4.2.2. Accelerometer Features

The triple-axis MEMS accelerometer which is used in this thesis and is integrated into the MPU-9150 has the following features [21]:

- Digital output 3-axis accelerometer with a programmable full scale range of ±2g,±4g, ±8g as well as ±16g
- An integrated 16-bit Analog-to-Digital-Converters (ADCs) makes simultaneous sampling of accelerometers possible, while no external multiplexer is required.
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

4.3. Gyrometer

4.3.1. Gyrometer basics

The gyrometer consists of a mass connected to a frame through springs. This frame is connected to a fixed outer frame with springs. These two frames have Coriolis sense fingers pointing to each other, compare figure7.

The mass moves constantly sinusoidal along the inner springs. Rotation leads to Coriolis acceleration in the mass, resulting in a movement towards the direction of the outer springs. If the mass moves away from the axis of rotation, the mass is pushed perpendicular in one direction. If the mass is driven back in the direction of the axis of rotation, the mass is pushed in the opposite direction. This is caused by the Coriolis force, which has an impact on the mass.

When the mass is moved by the Coriolis force, a change in the capacitance can be measured, because of the smaller distance between the sensing fingers. If the mass is moved in the opposite direction, other sensing fingers have a smaller distance inbetween. In that way the sensor can measure the magnitude and the direction of the angular velocity within the system.[27]

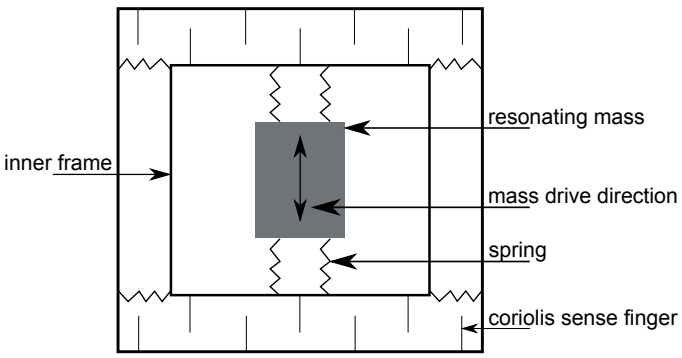

Figure 7: A simple gyrometer [27]

4.3.2. Gyrometer Features

The triple-axis MEMS gyrometer which is used in this thesis and is integrated into the MPU-9150 has the following features [21]:

- Digital-output X,Y,Z-axis angular rate sensors (gyrometers), with a programmable full-scale range of $\pm 250, \pm 500, \pm 1000$ and ± 2000 °/sec
- External sync signal connected to the FSYNC pin supports images, video and GPS synchronization
- Integrated 16-bit ADCs makes simultaneous sampling of gyrometers possible
- The need of user calibration is reduced by enhanced bias and sensitivity temperature stability
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Factory calibrated sensitivity scale factor
- User self-test

5. Activity recognition

The activity recognition consists of various steps. First the sensor (for details compare section 4) is placed, next the raw data are recorded. In the following step preprocessing is done, which means data segmentation and filters are applied. During feature extraction, features are calculated. Finally, the training and classification is done with a continuous HMM model (cHMM) or a k-nearest neighbors (kNN) classification. The activity recognition chain is displayed in figure 8. The following sections will explain these steps in more detail.

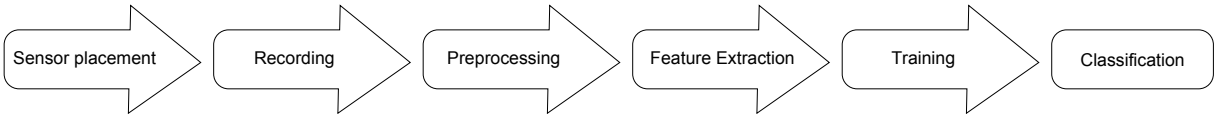

Figure 8: Activity recognition chain

5.1. Data

5.1.1. Placement

The MPU-9150 sensor is always placed on the left hand, with which the daily activities are mostly executed. There is only one left-handed person recorded, which ensures that normalizing of the data is unnecessary. For instance, if there is the need to compare left-handed and right-handed persons normalization is inevitable. The sensor is always located at the left hand wrist, with the DIP switch (compare figure 4) on the left, pointing towards the fingertips.

5.1.2. Recording

A three-dimensional accelerometer and gyrometer is used to record activities of daily living in different time units. For further details about the sensor compare section 4.

In table 1 common daily activities [30] are displayed, which are recorded during this study and accomplished by one person. An example of a typical sampling frequency is GPS with 5Hz, whereas acceleration is sampled at frequencies of 25Hz or more.[10] In this thesis a sampling frequency of 50Hz is chosen. The data are gathered in a $59m^2$ flat from one left-handed person.

The data are recorded in many sessions and each activity is saved with their 3-axis accelerometer and 3-axis gyrometer data. Inbetween all activities a 'NULL'-activity is performed, which consists of preparing the next activity and closing the preceding activity. The data gathering extends over days, which is the reason why the sessions are put together to one dataset for further analysis. The structure of the dataset can be seen in figure 9, where the labeled activities are displayed.

labels	activities			
1	NULL			
2	comb hair			
3	wash face			
4	wash hands			
5	brush teeth	electric	non electric	
6	make bed			
7	change clothes			
8	put blinds up/down			
9	prepare food			
10	eat with	folk	spoon	chopsticks
11	open/close window			
12	read newspaper	read book		
13	putting shoes on			
14	drink from/with	straw	mug	cup

Table 1: List of daily activities, which are performed during recording

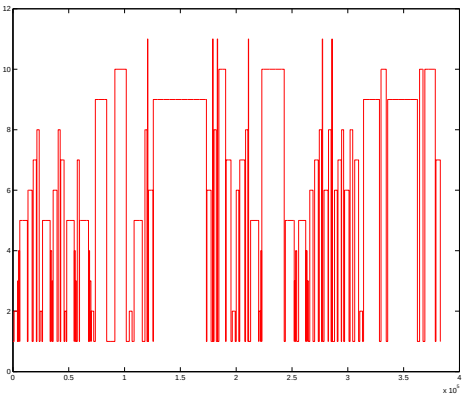

Figure 9: Labeled activities in the dataset

5.2. Data Segmentation

The data segmentation is needed to identify the segments of the data stream containing information about activities. Segmentation of a continuous sensor stream is difficult, as activities blur into each other more likely than separating it with pauses.
Another problem is the definition of the activities with its boundaries.[10] For example, 'Tooth brushing', might start with taking the tooth brush or putting toothpaste on it or raise the tooth brush to one's mouth or brushing.

In this thesis segmentation is performed via annotation during recording. This process saves

information over the start and the duration of the activity. The annotation of the data is automated via an app, which allows the recording within specific time units. This time units are determined by the user and the annotation of the data accords to the user's purpose.

5.3. Preprocessing

5.3.1. Fundamentals

In the first attempt the data are analyzed and only data which are recorded more than once are used for analysis. This is done with one of two methods.
The first method cuts out 'read newspaper/read book', 'putting shoes on' and 'drink from/with straw/mug/cup' from the whole dataset. Be aware that the 'NULL'-classes between the activities remain in the dataset. The second method redefines the labels of those classes to the label of the 'NULL'-class.
Some of the other activities are put together to one: 'Tooth brushing electric' and 'Tooth brushing non electric' get label 5 as well as 'Eat with folk', 'Eat with spoon' and 'Eat with chopsticks', which are assigned label 10.

Each sensor records data in three dimensions and a fourth dimension for each sensor is added. This fourth dimension describes the magnitude $\sqrt{x^2 + y^2 + z^2}$. For each dimension the features are calculated. This means the feature calculation gives four values per sensor, except from one feature, the correlation. The correlation gives 3 values per sensor, because the magnitude correlates of course with the three other dimensions.

5.3.2. Signal Processing

Raw sensor data can be interrupted by artifacts and noises. This is a flaw which has to be eradicated as much as possible.
In the preprocessing step the artifacts and noises are reduced and the signal is prepared for later feature extraction.[10] The noise and artifacts are disturbances which can corrupt the human activity recognition.

Filters play a major role to get rid of artifacts and noises during preprocessing. In this thesis experiments are done with a median filter and a 3rd order low-pass Butterworth filter. This filters are also used among others in [4], that is the reason why those are chosen. The 3rd order low-pass Butterworth filter has a cutoff frequency of 20Hz. This rate is sufficient, as the frequency of human body motions is 99% below 15Hz.[4] Details about the experiments with filters can be found in section 6.2.2.

Median filter

Median filters have a broad application area, because their behavior is well known. The non-linear median filter is able to filter out noise without blurring edges or other sharp details. Median filters are most likely to get applied, if the noise pattern consists of strong, spike-like components. The median filter considers each entry of the array and looks at nearby neighbors to decide the representative character of its surrounding. It replaces the values with the median of the values in the surrounding. The median is calculated by sorting the values into numerical order and then the median of these values is assigned to the representing value. The application of the filter causes a smoothing of the picture. The smoothing effect on the data of this thesis can be seen in figure 10 below, where a small part of the data is displayed. The original data is displayed in blue color and the median filtered data is displayed in red color.[2]

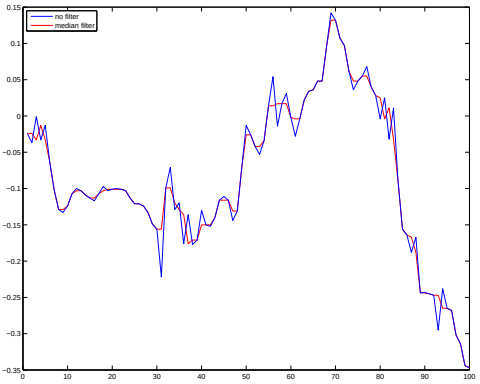

Figure 10: 3rd order one-dimensional median filter: segment of the dataset

Butterworth filter

A Butterworth filter has a maximally flat amplitude response in the pass-band. The attenuation level is -3db and -20db. The design of the Butterworth filter is easy, meaning if it is designed in analog domain, it can be easily converted into a digital filter. Also in higher order the Butterworth filter maintains the main shape. The filter is described with two parameters, which are the cutoff frequency and the number of poles. Butterworth filters are used in this thesis for cutting high frequencies and therefore behaving as Low Pass Filters. In small scale, the functionality of a 3rd order low pass Butterworth filter with 20Hz cutoff frequency is shown in figure 11.[24]

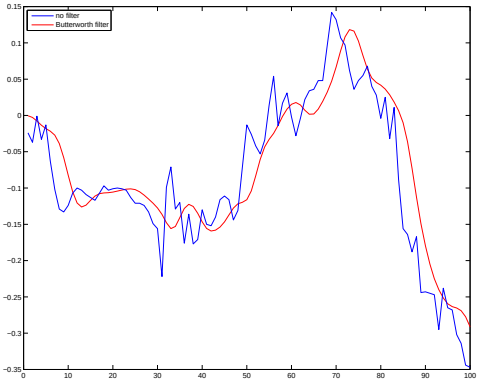

Figure 11: 3rd order low pass Butterworth filter, 20Hz cutoff frequency: segment of the dataset

5.4. Feature Mapping

In the feature extraction step the raw data are converted into features. This features are calculated over each annotated activity with a shifted window. This window of size 50 (containing 50 data vectors) is shifted with a 50% overlap, which is the most significant value for overlap in past works.[40][8] The mean, standard deviation, correlation[8][35], energy[8][35] and frequency domain entropy[8] are calculated for this data, as those are the most popular features for acceleration signal in activity recognition.[18]

The features can be divided into time domain features and frequency domain features. Time domain features are mean, standard deviation and correlation. Frequency domain features are energy and entropy. The energy and entropy calculation is much more expensive in comparison to the time domain features, because of the Fourier transformation (FFT).[37]
A periodic function in time is described with a direct current (DC) component. The DC component over the window is the mean value. Standard deviation is important for the reason of different range of values for different activities. Periodicity in the data is saved in the energy feature. Correlation between axes is useful to differentiate activities with translation in one dimension. As example, walking and stair climbing can be distinguished over correlation data.[8][35]

In table 2 the calculation methods for the different features are depicted. It is important to use a minimum number of features that allow good performance and at the same time minimize computational costs and memory.[10]
Experimenting with the features showed that entropy doesn't improve the results. That is the reason why it is not pointed out in the descriptions of section 6. The best combination of features are mean, standard deviation and correlation. For details refer to section 6.2.

Features	calculation	variables
mean	$\mu = \frac{1}{w}\sum_{j=1}^{w} x_j$	$w \ldots$ window length $x_j \ldots$ values
standard dev.	$\sigma = \sqrt{(\frac{1}{w}\sum_{j=1}^{w}(x_j - \mu)^2)}$	$\mu \ldots$ mean
energy	energy$= \frac{1}{w}\sum_{j=1}^{w} \mid x_j \mid^2$	$x_j \ldots$ discrete FFT components
correlation	$cov(x,y) = \frac{1}{w}\sum_{j=1}^{w}(x_i - \mu_x)(y_i - \mu_y)$ $corr(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y}$	$cov \ldots$ Covariance
entropy	Frequency-domain entropy is calculated as the normalized information entropy of the discrete FFT component magnitudes of the signal.[8]	

Table 2: Features

5.5. Training

This thesis deals with cHMM and kNN models, for further details of the experiments compare section 6.2. The supervised models need to get trained before operating.[10] Therefore the data has to be split into training and test data.
As some activities are not so common it is not possible to divide the data (see figure 9) in usual way like 20% test data and 80% training data. In this study one activity is cut out from each activity class, with the 'NULL'-class behind. Those activities are put together as test data. The rest of the data is used as training data. The structure of one sort of training and test data can be seen in figure 12 and 13.

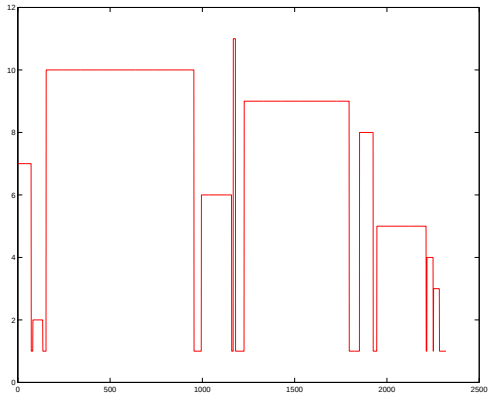

Figure 12: Training data

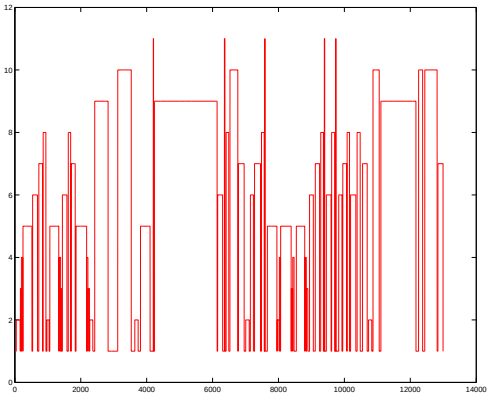

Figure 13: Test data

The training needs a training set $\{(\mathbf{X}_i, y_i)\}_{i=1}^{N}$ consisting of N pairs of feature vectors $\mathbf{X}_i$ with corresponding labels y_i. In cHMMs (compare sections 3.2) the model parameters $\theta = (\pi, A, B)$ are learned by minimizing the classification error.[10] In this thesis the parameter A, transition matrix, can be calculated with the labeled training data. The parameter B is a list of pairs (μ, Σ) that define the distributions, which can also be calculated. Only π has to be guessed and the model is defined by means of these parameters.

In contrast, nonparametric classifiers like kNN models employ as parameters the labeled training set and match the label of the k-nearest neighbors to the test sample, details can be found in section 3.3.[10]

5.6. Classification

The classification consists of two steps.
The first one maps a set of class labels to each feature vector of the test data with corresponding scores.[10]
In the second step the scores are used in this thesis to calculate the maximum score and take the corresponding class label y_i as the classification output.[10]

5.7. Performance Evaluation

The classification of the activities can be either correct 'True Positive'(TP) and 'True Negative'(TN) or wrong 'False Negative'(FN) and 'False Positive'(FP). The performance metric which is used for this model is a confusion matrix, with accuracy, sensitivity(=recall), specificity and precision. The calculation formulas of the characteristics are displayed in figure 3.[10]

	calculation
accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
sensitivity = recall	$\frac{TP}{TP+FN}$
specificity	$\frac{TN}{TN+FP}$
precision	$\frac{TP}{TP+FP}$

Table 3: Calculation of performance parameters

5.7.1. Confusion Matrix

The confusion matrix gives a breakdown of the misclassified activities by the model. The rows show the instances in each actual activity class (defined by the ground truth) and the columns show the instances for each predicted activity class (defined by the classification output).[10] The values in one row are the results from the comparison of all ground truth instances, from the actual class, to the predicted class labels.[10] The ground truth is the accurate labeling of all activities.[17] An example of an simple confusion matrix can be seen in table 4, where the last column describes the recall values, the last row the precision values and the last box describes the accuracy.
In the confusion matrix accuracy, precision, recall and specificity are calculated for each activity class. For comparison reasons the 'NULL'-class calculations of the characteristic (accuracy, precision, recall, specificity) are quoted in the discussions of the experiments in section 6.2.
If the dataset is unbalanced, for example when the number of ground truth instances vary significantly, the overall accuracy is not representative for the whole classifier. A normalized confusion matrix inhibits this problem by using percentage of the total number of ground truth activity instances.[10] This problem occurs also in small scale in this thesis, that is the reason why all parameters are included in performance evaluation and no normalization is done.

	activity 1	activity 2	activity 3	recall
activity 1	11	2	0	84.62
activity 2	0	4	0	100
activity 3	1	0	5	83.33
precision	91.67	66.67	100	86.96

Table 4: Simple confusion matrix

6. Discussion

In this section different experiments with the recorded data are discussed in detail. In the first attempt a validation with the provided data from Bulling *et al.* [10] and Anguita *et al.* [4] is accomplished to justify the use of the continuous Hidden Markov Model (cHMM). In further work various experiments are done to reflect the behavior of the recorded data.

Filters are normally used to improve the results. In this case Butterworth filter and median filter are applied, compare section 5.3, but no qualitative improvement can be recognized. Instead it increases the computation costs, what makes the application of filters unnecessary for this study. Another attempt to get an improvement is to introduce sub-activities. This is done by applying a cHMM with more than one state for certain activities, depending on the best accuracy. Of course an improvement is recognized, which implies an increase in runtime. As this step is parameter sensitive it is not worth the effort.
The results show that gyrometer data are superfluous for the sort of daily activities recognized in this study. In general, there is an important aspect of gyrometer data when considering walking upstairs and downstairs, for instance. Accelerometers are able to handle rotation changes and linear movement, where gyrometers are limited to rotation only.[10]
The most important outcome of this thesis is that considering continuous and discrete data is most effective and should be paid attention in further research.

6.1. Validation

A validation is inevitable to get an overview of the correctness of the described cHMM and the legitimacy to use it in further experiments.
In this section the validation and its results are discussed in detail. The cHMM is validated with the data of Bulling *et al.* [10] and the data of Anguita *et al.* [4]. The results are compared with those results from [10] and [4], respectively and entitles the use of the cHMM.

6.1.1. Data provided by Bulling *et al.* [10]

Bulling *et al.* provides data for person-dependent (pd) training and person-independent (pi) training. The pd training means training a model with data from one person and pi training means training one model with data from more than one person. The results in figure 14 [10] show that the pd training circumstances lead to more appropriate results than the pi training. This is of course comprehensible, as different activities can be accomplished in various ways and this affects the precision and recall. For example preparing dinner can have a lot of sequences, putting out the needed foods for cooking at once or taking the foods step by step. To evaluate the applied cHMM it is useful to compare the results to those from pd training.
In [10] data from 2 persons performing 12 activities are recorded: opening a window, closing

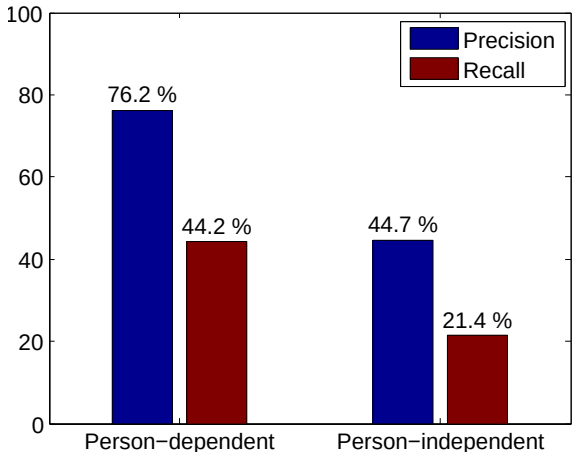

Figure 14: Precision and recall for person-dependent and person-independent evaluation using a single accelerometer attached to the right hand [10].

a window, watering a plant, turning book pages, drinking from a bottle, cutting with a knife, chopping with a knife, stirring in a bowl, forehand, backhand and smash. Additionally, a non-specific activity was performed called 'NULL'-class.[10]
The inertial measurement unit (IMU) is placed on 3 positions, the upper arm, the lower arm and the hand wrist on the right side. For evaluation, the hand position is used, as in this study only hand movement is recorded. With the IMU, data from a 3-axis accelerometer and a 2-axis gyrometer is collected with a 32Hz sampling rate.[10] The features for evaluation are mean, standard deviation, correlation and energy, compare table 5.

activities	features	sampling rate
opening a window	mean	32Hz
closing a windows	variance	
watering a plant	correlation	
turning book pages	energy	
drinking from a bottle		
cutting with a knife		
chopping with a knife		
stirring in a bowl		
forehand		
backhand		
smash		
NULL		

Table 5: Specification of the data of Bulling *et al.* [10]

6.1.2. Data provided by Anguita *et al.* [4]

Anguita *et al.* gathered data from 30 volunteers between 19 and 48 years old, which followed a defined protocol of activities. This protocol consists of six activities, which are standing, sitting, laying down, walking, walking downstairs and upstairs, which are collected via a Galaxy S II. smartphone on their waist. Accelerometer and gyrometer data are recorded with a sampling rate of 50Hz and 5 seconds break between two activities.[4] An overview of the different activities, features and the sampling rate can be seen in table 6.

In this thesis only a part of the 561 features vector of the provided data is picked for evaluating the constructed cHMM. The feature data is already noise reduced by a median filter and a 3rd order low-pass Butterworth filter with a 20Hz cutoff frequency and others.[4] In particular the data of mean, standard deviation and correlation is used from accelerometer and gyrometer for all three axes X, Y, Z. This features are the most similar features to the calculations performed in this thesis.

activities	features	sampling rate
standing	mean	50Hz
sitting	standard deviation	
laying down	correlation	
walking		
walking downstairs		
walking upstairs		
NULL		

Table 6: Specification of the data of Anguita *et al.* [4]

6.1.3. Results from cHMM compared with results from Bulling *et al.* [10]

In this section the results of Bulling *et al.* [10] are discussed. This means data provided by Bulling *et al.* [10] are used to calculate the features: mean, standard deviation, correlation and energy for all 7 axes. This 7 axes come from the 3-axes accelerometer and 2-axes gyrometer gathered data, including one axis for each sensor, representing the magnitude. This features are further used to calculate the cHMM model introduced in this thesis.

The calculated results from the cHMM are compared with the results in Bulling *et al.* [10]. This circumstances are shown in figure 15, where precision and recall of pd evaluation is compared to the applied cHMM in this thesis using the same dataset. The same characteristic, namely lower recall than precision, can be seen. The different values between results of Bulling *et al.* [10]

and this thesis are mostly caused by the number of features and the model used in [10]. They only calculated two features, mean and standard deviation and the classification algorithm uses a folding step.[10] In comparison this thesis takes into account the mean, standard deviation, energy and correlation features.

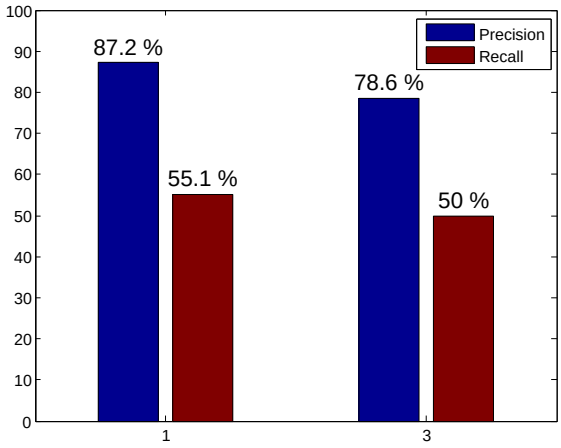

Figure 15: Precision and recall for pd evaluation with sensor placement on the right hand and cHMM used in this thesis.

In figure 16 the original labeled path is represented in red color and the result after applying the cHMM is represented in blue color, which is called Viterbi path [32]. It can be seen, that the Viterbi path fits the original path quite well.

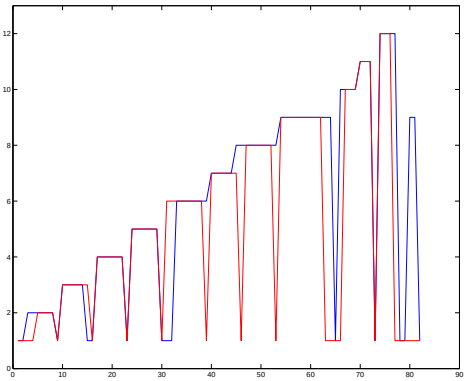

Figure 16: Viterbi path compared to original label path

In table 7 the confusion matrix of the cHMM is illustrated. In [10] the precision lies by 87.2% in the pd case using the sensor placed on the hand wrist. This relates to the results of an accuracy of 81.71% and good recall and precision values for each activity class. Therefore it follows, that the used cHMM is accurate.

	NULL	Open window	Drink	Water plant	Close window	Cut	Chop	Stir	Book	Forehand	Backhand	Smash	Recall (%)
NULL	11	2	0	0	0	1	0	2	4	1	0	1	50
Open window	0	4	0	0	0	0	0	0	0	0	0	0	100
Drink	1	0	5	0	0	0	0	0	0	0	0	0	83.33
Water plant	0	0	0	6	0	0	0	0	0	0	0	0	100
Close window	0	0	0	0	6	0	0	0	0	0	0	0	100
Cut	2	0	0	0	0	6	0	0	0	0	0	0	75
Chop	0	0	0	0	0	0	5	1	0	0	0	0	83.33
Stir	0	0	0	0	0	0	0	6	0	0	0	0	100
Book	0	0	0	0	0	0	0	0	9	0	0	0	100
Forehand	0	0	0	0	0	0	0	0	0	3	0	0	100
Backhand	0	0	0	0	0	0	0	0	0	0	3	0	100
Smash	0	0	0	0	0	0	0	0	0	0	0	3	100
Precision (%)	78.57	66.67	100	100	100	85.71	100	66.67	69.23	75	100	75	81.71

Table 7: Confusion matrix of cHMM

6.1.4. Results from cHMM compared with results from Anguita *et al.* [4]

In figure 17 the periodic character of the provided training data can be seen. The original labeled path is colored red and the resulted Viterbi path of the model is colored blue. The outcome shows a good fit from the calculated Viterbi path to the original labels and confirms the presumption that the model works correct.

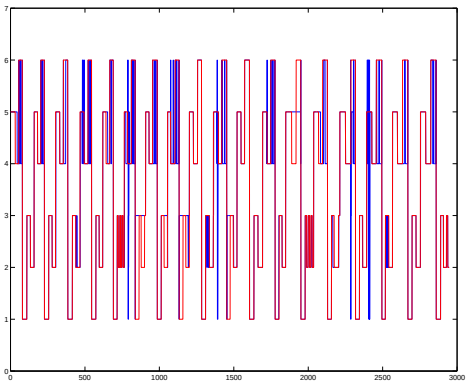

Figure 17: Viterbi path compared to original label path

In table 8 the confusion matrix of the cHMM used in this thesis is depicted. In table 9 the results of the experiment in the paper of Anguita *et al.* [4] is reproduced. The rows represent the actual class, the columns the predicted class and the last box represents the accuracy of the model.[4]

In contrast to the cHMM applied in this study, Anguita *et al.* used a multiclass Support Vector Machine (MC-SVM). The different accuracy can be attributed to the less used features, the usage of only one accelerometer and one gyrometer dataset, and the more complex MC-SVM model in [4].

	Walking	W.Upstairs	W.Downstairs	Sitting	Standing	Laying Down	Recall (%)
Walking	422	0	74	0	0	0	85.07
W.Upstairs	0	400	71	0	0	0	84.93
W.Downstairs	0	39	381	0	0	0	90.71
Sitting	2	1	3	406	68	11	82.69
Standing	6	9	5	26	468	18	87.97
Laying Down	0	0	2	298	44	193	35.94
Precision (%)	98.14	89.09	71.08	55.62	80.69	86.94	77.03

Table 8: Confusion matrix of cHMM

	Walking	W.Upstairs	W.Downstairs	Sitting	Standing	Laying Down	Recall (%)
Walking	492	1	3	0	0	0	99.12
W.Upstairs	18	451	2	0	0	0	95.75
W.Downstairs	4	6	410	0	0	0	97.62
Sitting	0	2	0	432	57	0	87.98
Standing	0	0	0	14	518	0	97.37
Laying Down	0	0	0	0	0	537	100
Precision (%)	95.72	98.04	98.80	96.86	90.09	100	96

Table 9: Confusion matrix of MC-SVM [4]

6.2. Experiments

First, activities which appear only once during recording are either put into the 'NULL'-class or cut out from the overall activities. Some activities like 'Tooth brushing electric' and 'Tooth brushing non-electric' are put together to become more appropriate results compare 5.3.

Therefore, eleven activities are tested against each other with different sorts of test and training data and different features. The form of the test data and training data differs. It is not so accurate to leave 20% for testing, because of the small training sample and the recording type of the data. That is the reason why for each activity one of the overall events is cut out with 'NULL'-class behind each activity or with 'NULL'-class in front of each activity. The structure of the whole dataset can be seen in figure 9 and an example of a training and test set can be seen in figure 12 and 13.

The different training sets lead to different results, whereby the best result is supplied by the test set, consisting of the 3-rd of each activity class and the following 'NULL'-classes.
In the next step the different features are tested, answering the question which combination of features fits best. The conclusion is, that the combination of mean, variance and correlation is the best combination. Also, the accelerometer and the gyrometer data are tested against each other. In this case accelerometer data on their own produce more accurate results, whereas gyrometer data corrupt the good results to some degree. During the experiments a cHMM is fitted, to get sub-activities from the 'NULL'-class, searching for the best number of states over cross-correlation. Overall nothing improves, but if cHMMs for sub-activities from different classes are combined an improvement of nearly one percent can be expected. As this is not a huge improvement and the aim is a short run time combined with a huge accuracy, the cHMM fit is not used in further attempts.
The next step is the inclusion of a feature vector, defining the room in which each activity is performed. This leads to a huge improvement and lets presume that the combination of smart home sensor data and data from wearable sensors should be combined to get the best results. In this case the 'NULL-'classes have the half room number of the previous activity and half the room number of the following activity. Of course this are not the exact room numbers, as the 'NULL'-activities can be performed in more than one room during executing. But this fictive example allows the statement of improvement, when room sensors are included.

6.2.1. Experiment with different training and test sets

In this section, the experiment of using different sort of training and test data partitions is analyzed. For example, the test data includes the third repetition of each single activity class and takes either the 'NULL'-class behind or in front of each cut activity. Another approach uses the second repetitions, again with either the 'NULL'-class in front of the activities or behind. The results of these experiments are compared to each other and the conclusion is, that the third

repetition, with the 'NULL'-class behind each activity, leads to the best results.

Figure 18 and 19 show the results of the Viterbi path (blue) compared to the original labeled path (red), for the 3rd repetitions, with the 'NULL'-class either behind (figure 18) or in front (figure 19). The greatest difference can be seen in the area of activity 7, namely 'Putting blinds up/down'. This is the case here, because a different 'NULL'-class is in front of activity 7.

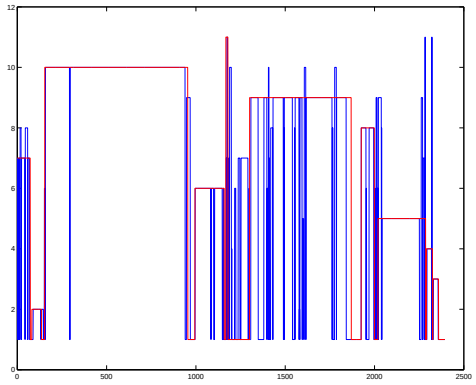

Figure 18: 3rd activities with 'NULL'-class behind

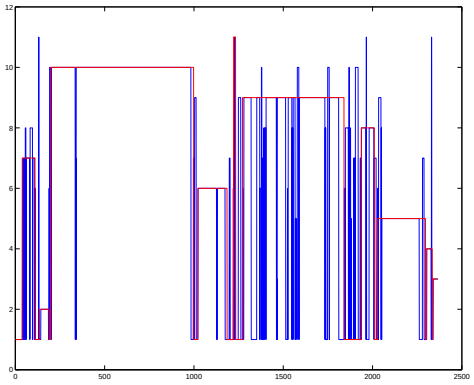

Figure 19: 3rd activities with 'NULL'-class in front

Figure 20 and figure 21 show the 2nd repetitions with 'NULL'-class behind and in front, respectively. The visible difference can be seen, for instance, between activity 6, 'Make bed' and activity 9, 'Prepare food'.

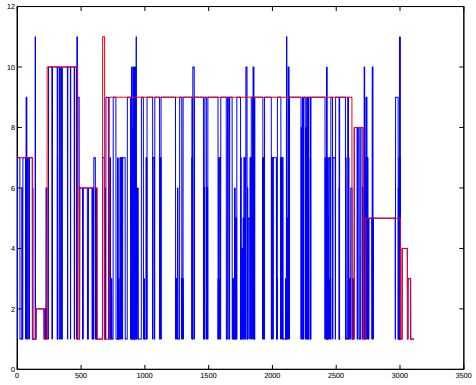

Figure 20: 2nd activities with 'NULL'-class behind

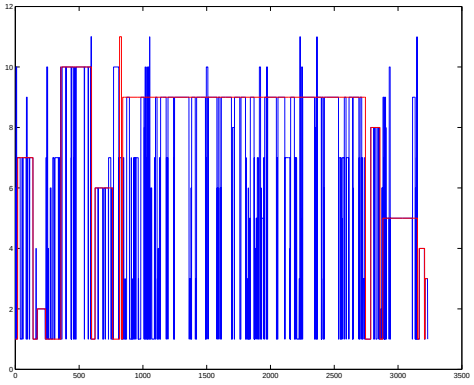

Figure 21: 2nd activities with 'NULL'-class in front

In table 10 the changing accuracy, specificity and sensitivity is illustrated. Therefore, the conclusion must be drawn, that the 3rd activities with 'NULL'-class behind, implies the best results.

In table 11 different feature combinations are constructed and the accuracies are compared. 'Whole' means the whole dataset is used in this analysis step, namely mean, variance, correlation as well as energy for all axes X, Y, Z and the magnitude. For more details compare section 5.3. 'Cut magnitude and energy' means that this features are left out and an improvement can be recognized. 'Cut energy' only operates with the features mean, variance and correlation with the magnitude. 'Cut magnitude' deals with the features mean, variance, correlation and energy, but without the magnitude. It can be noticed that the dataset 'Cut magnitude and energy' leads to the best accuracy, except for the used 2nd repetition with 'NULL'-class in front, where the dataset 'Cut magnitude' is the most accurate.

Experiment	accuracy	specificity	sensitivity
3rd back	80.24	89.08	63.81
3rd front	79.84	88.35	61.25
2nd back	63.99	82.66	67.91
2nd front	64.67	85.48	56.18

Table 10: Accuracy, specificity and sensitivity for different train and test sets.

Experiment	whole	cut mag. & energy	cut energy	cut magnitude
3rd back	73.52	80.24	77.07	74.77
3rd front	72.15	79.84	76.50	74.94
2nd back	64.15	63.99	60.48	66.27
2nd front	64.48	64.67	60.74	68.41

Table 11: Comparison of different features for different train and test sets.

6.2.2. Experiment with filter and no filter

In this section a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20Hz is used to remove noise, based on the paper of Anguita *et al.* [4]. The impact of the median filter and Butterworth filter can be seen in detail in figure 10 and 11. For more details compare section 5.3.

In table 12 accuracy, specificity and sensitivity of filtered and non-filtered data are represented. If filters are applied, the sensitivity gets better, but the accuracy and specificity gets worse. This reaction makes filters in this study unnecessary. In table 13 an improvement is realized, when different combinations of features are used. The results of 'Whole' data as well as 'Cut magnitude' are improved while applying filters.

Filters are not necessary for the dataset in this study, as the smallest amount of features with the best result is search for, which leads to a reduction in runtime.

Experiment	accuracy	specificity	sensitivity
no filter	80.24	89.08	63.81
filter	79.53	88.26	67.62

Table 12: Accuracy, specificity and sensitivity for filtered and non filtered data

Experiment	whole	cut mag. & energy	cut energy	cut magnitude
no filter	73.52	80.24	77.07	74.77
filter	74.19	79.53	76.07	75.23

Table 13: Comparison of different features for filtered and non filtered data

6.2.3. Experiment with Accelerometer/gyrometer

In this experiment accelerometer and gyrometer data are analyzed on their own and as one combined dataset. The outcomes of the accelerometer and gyrometer data on their own are further compared with the results of Bulling *et al.* [10].

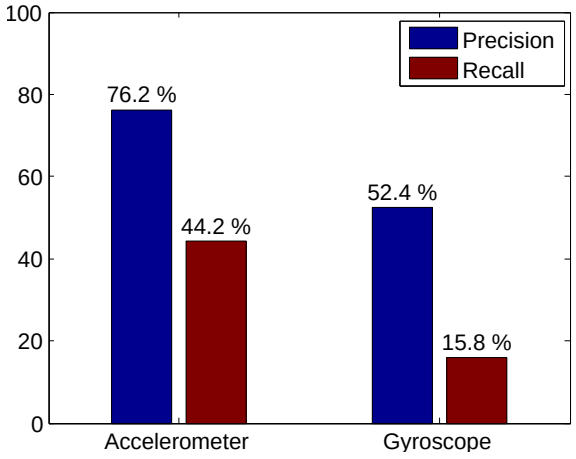

Figure 22: Precision and Recall of gyrometer and accelerometer data provided by Bulling *et al.* [10]

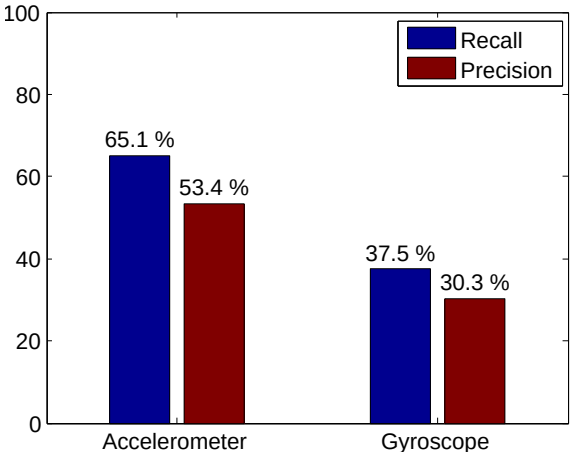

Figure 23: Precision and Recall of the gyrometer and accelerometer data in this study

The accelerometer data on it is own is more accurate than only using the gyrometer data. This results coincide with those of Bulling *et al.* [10]. In [10] the gyrometer data on their own is also worse than the accelerometer data on their own. These circumstances are illustrated in figures

22 and 23, where precision and recall are symbolized as blue and red bars. The left figure 22 shows the result of one accelerometer sensor located on the right hand wrist, provided by Bulling *et al.* [10]. The right figure 23 shows the results of the data in this study. It can be noticed, that the results in the study of Bulling *et al.* [10] reach higher level of precision and lower level of recall than the results in this study. This is mostly caused by the greater dataset in [10]. The results in this study are not worse, but differ in activities as well as range and sort of recording. The combination of accelerometer and gyrometer data is less accurate, therefore the conclusion can be drawn that gyrometer data is unnecessary in this case, compare table 14.

Experiment	accuracy	specificity	sensitivity = recall	precision
accelerometer	83.08	91.39	65.08	53.39
gyrometer	63.87	86.96	37.46	30.33
acc. & gyr.	80.24	89.08	63.81	46.96

Table 14: Accuracy, specificity and sensitivity for gyrometer, accelerometer and whole data

In figure 24 a good phenomena can be seen, namely the misclassification of one activity class with another. In this case activity 9 ('Prepare food') is often misclassified as activity 10 ('Eat'). Also the much worse fit in figure 24 is visible, contrary to figure 25. Figure 25 shows, by far, the better fit of the Viterbi path (blue) to the original labeled path (red). In figure 26 the results of the combined accelerometer and gyrometer data are depicted.

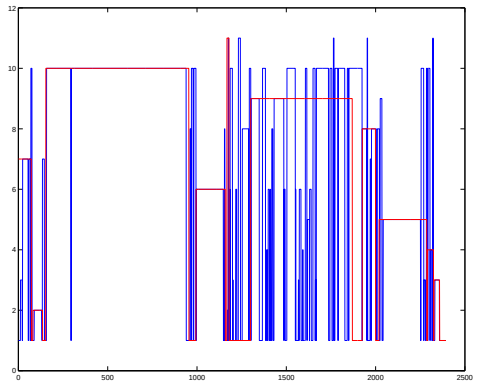

Figure 24: Gyrometer

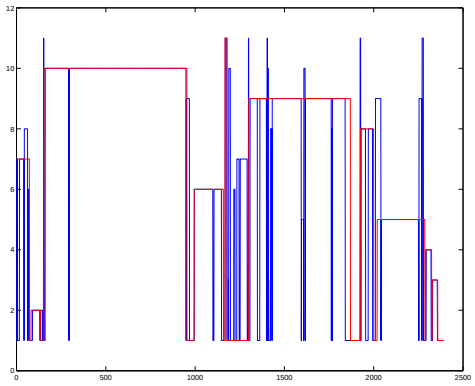

Figure 25: Accelerometer

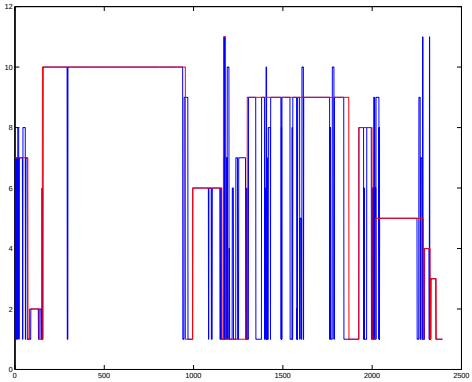

Figure 26: Whole dataset

6.2.4. Experiment with extra cHMM

In this experiment activities are divided in sub-activities with an extra cHMM. This is accomplished in the following way:
For each often misclassified activity-class, a cHMM is applied to divide the activities in sub-activities. The number of sub-activities depends on the number of states in the cHMM. Hence an iteration is done, constructing a 2-state to 5-state cHMM and choosing the cHMM with best accuracy. If the accuracy is higher in the present step than the previous step, the present model is taken as new model. This iteration is proceeded till the previous model is better than the present model. Than the iteration terminates and the previous model is finally chosen.
For example, assume 'Prepare food' is divided in 2 sub-activities, then the activity becomes two classes $[9, 10]$ instead of one $[9]$. All following classes are accommodated to fit the number-chain 1-12, in comparison to the original number-chain 1-11.

The experiments show that the output becomes better, if activities, which are connected before, are automatically divided into sub-activities, like tooth-brushing.
Tooth-brushing is the obvious candidate where it makes sense, to use sub-activities. This is the case, as the classes 'Tooth brushing' and 'Tooth brushing electric' are put together during pre-processing. This activity afterwards consists of two classes $[5, 6]$ instead of one $[5]$. The original classes from 6 to 11 are accommodated.
The improvement when using the divided 'Tooth brushing' class is obvious, as tooth-brushing with an electric tooth-brush has a higher frequency, contrary to non-electric tooth-brushing and therefor is easy to divide.

In table 15 the difference between accuracy, specificity and sensitivity for normal cHMM and cHMM with sub-activities is shown. The sub-activities are constructed for 'Tooth brushing' and 'Putting blinds up/down'. It can be recognized that accuracy and specificity gets better, but not

as much to justify much higher runtimes. For that reason, it is decided not to use sub-activities.

Experiment	accuracy	specificity	sensitivity
cHMM	80.24	89.08	63.81
cHMM: Tooth & Blinds	82.62	90.67	61.90

Table 15: Accuracy, specificity and sensitivity for cHMM and extra cHMMs

In table 16 the parameter sensitivity can be seen in small scale. Parameter sensitivity means dependence of accuracy on certain parameters. During the experiments only the parameter, for 'k'-fold cross-validation and the parameter, number of activity classes divided, are considered. But also a change in tolerance and of maximal iterations within the cHMM affects the outcome. The tolerance is always set to $1e^{-5}$ and the maximal iteration is set to 10. Three cases are pointed out in table 16, first the 'Tooth brushing' and 'Putting blinds up/down' are divided with 5-fold cross validation, later on with 10-fold cross-validation. The last case is the combination of three divided classes 'Tooth brushing', 'Putting blinds up/down' and 'Prepare food' with 5-fold cross-validation. It can be seen that the best accuracy of those three cases resulted during the experiment with three divided activities and 5-fold cross-validation.

activities	classes	cross-val	accuracy
Tooth & Blinds	2	5	82.62
Tooth & Blinds & Food	3	5	83.08
Tooth & Blinds	2	10	82.62

Table 16: Different combinations of parameters for the extra cHMMs

In figures 28 to 30, the results of table 15 are displayed as Viterbi path (blue) correlating with the original labeled path (red). Those paths fit quite well to each other. Figure 27 points out the difference to the original model with no sub-activities. From the original activities 'Tooth brushing'(5),'Putting blinds up/down'(8) and 'Prepare food'(9) in figure 27 to figures 28 - 30, a fluctuation can be recognized. This fluctuations are caused by the new sub-activities. The results with sub-activities show a better fit to the original path in that region.

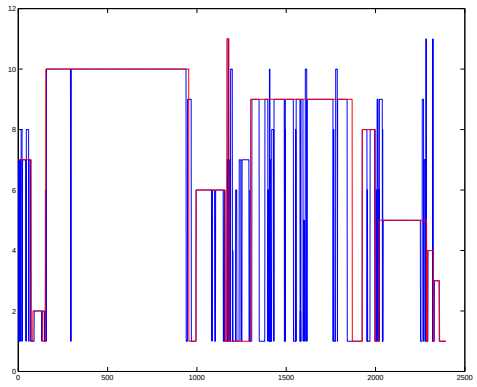

Figure 27: Whole dataset without magnitude and energy and without subactivities

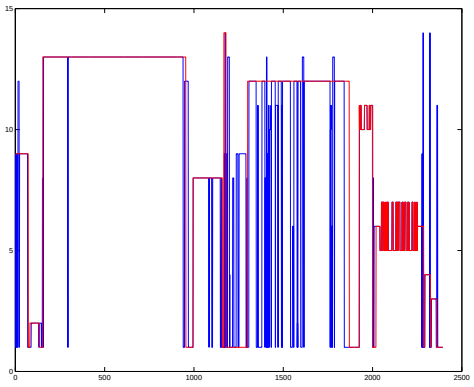

Figure 28: Extra cHMM for tooth brushing and putting blinds up/down, cross-val:5

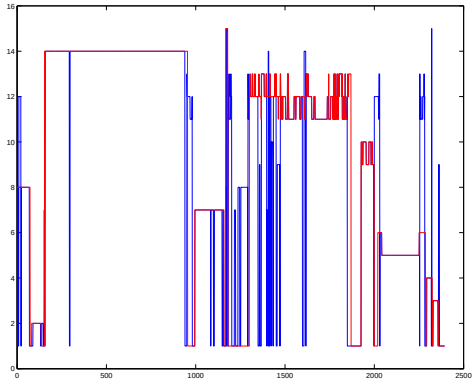

Figure 29: Extra cHMM for tooth brushing, putting blinds up/down and prepare food, cross-validation:5

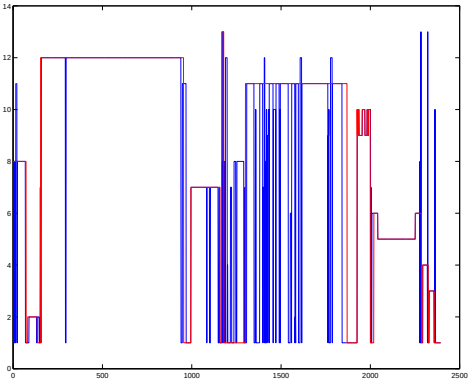

Figure 30: Extra cHMM for tooth brushing and putting blinds up/down, cross-validation:10

6.2.5. Experiment with continuous and discrete data

In the experiments above, the dataset is cut with method two, which is described in the preprocessing, compare section 5.3. That means the activities which occur only once in the recording period are contained in the 'NULL'-class activities. In this section the first cutting method is used, which is described in section 5.3. There the once recorded classes are cut out of the whole dataset, leading to an one percent improvement. In real environmental applications this makes just a small difference and therefore is unnecessary to be performed.

The combination of continuous data and discrete data is analyzed in this section. The test and

training data get an additional column. This column consists of the room number, where the activities are performed during recording. The 'NULL'-class activities become half the room label from the previous activity and half the room label from the following activity.
This process represents the usage of smart home sensors in combination with wearable sensors. An improvement of the results can be recognized and justify the detailed occupation with this topic in further research work.

In table 17 accuracy, specificity and sensitivity of continuous data and the combination of continuous and discrete data is displayed. The outcomes show that accuracy improves about 9% and sensitivity about 10%, while specificity stays nearly the same. This justifies the effort of collecting both data, continuous and discrete.

Experiment	accuracy	specificity	sensitivity
continuous	81.21	99.66	79.03
continuous & discrete	88.75	100	91.39

Table 17: Accuracy, specificity and sensitivity for continuous and continuous & discrete data

A great improvement can also be seen in the comparison of the Viterbi path colored blue and the original labeled path colored red in figure 31 and 32. The left figure 31 shows the continuous data results and the right figure shows the combined continuous and discrete data results. The differences can be seen, for instance, in the region of activity 'Prepare food'(9) and 'Change clothes'(7). This great differences are confirmed by considering the confusion matrices in table 18 and 19.

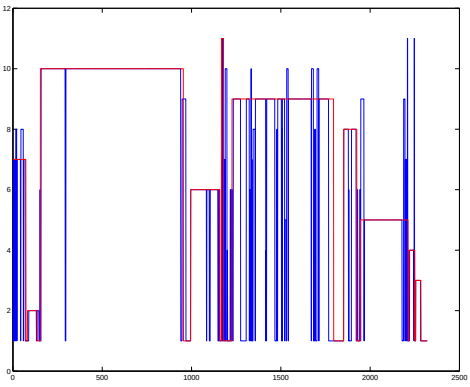

Figure 31: Continuous data

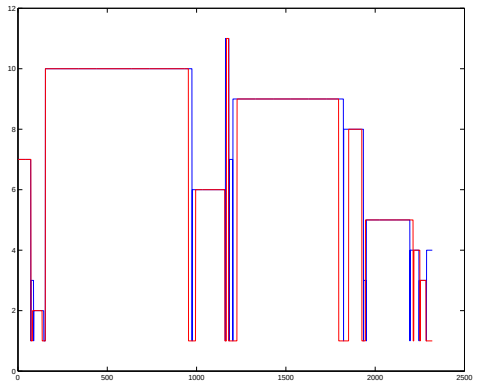

Figure 32: Continuous and discrete data

The confusion matrices show the detailed improvement from using continuous data only to

operating with the combined dataset, consisting of continuous and discrete data, compare table 18 and 19.
For example, 'Open/close window' is often misclassified as 'NULL'-class and as 'Changes clothes' in the continuous data case. After taking discrete sensor labels into account, it improves to 100% right classification. Another good example is the activity class 'Prepare food', which is misclassified in the continuous dataset with 'NULL'-class, 'Wash face', 'Tooth brushing', 'Make bed', 'Change clothes', 'Put blinds up/down' and 'Eat'. After the combination with discrete data, the classification improves to 100%.
Overall the classes are not so likely misclassified as 'NULL'-class anymore.
One drawback is that the 'NULL'-class is much more likely misclassified as other activities, which can be seen in the first row in figure 19.

6.3. Comparison with kNN-model

In this section a small insight, into other models, is given. For this a cHMM is compared to a k-nearest neighbors (kNN) classifier, with the same dataset. The kNN classifier is used, because of its easy fine-tuning. This guarantees an accurate representation of comparing different models. The kNN classifier is iterated with k between 1 and 20. The model with the lowest error rate, in this case with $k = 19$, is compared to the cHMM with the best accuracy.

In figure 33 the left bar displays the cHMM model with an accuracy of 80.24% and the right bar displays the kNN classifier with an accuracy of 32.16%. As expected, it is obvious that the cHMM model works much better than the kNN classifier. The conclusion is that cHMMs are more accurate than kNN classifiers. Therefore they have a broader application area in the field of activity recognition.

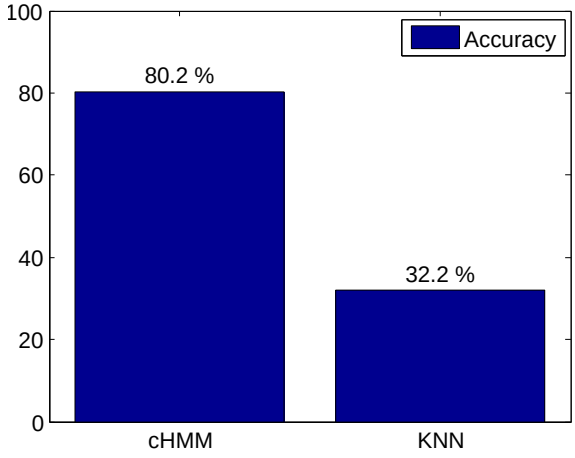

Figure 33: Comparison of cHMM and kNN classifier

	NULL	Comb hair	Wash hands	Wash face	Tooth brushing	Make bed	Change clothes	Put blinds up/down	Prepare food	Eat	Open/close window	Recall (%)
NULL	183	10	0	2	0	17	5	0	14	10	0	75.93
Hair	10	44	0	0	0	0	0	0	0	0	0	81.48
Hands	5	0	27	0	0	0	0	0	0	0	0	84.38
Face	7	0	0	25	0	0	0	0	0	0	2	73.53
Tooth	19	0	0	0	211	1	8	0	26	0	2	79.03
Bed	16	0	0	0	0	148	0	0	0	0	0	90.24
Clothes	18	0	0	0	0	0	34	20	0	0	0	47.22
Blinds	20	0	0	0	0	2	0	52	0	0	0	70.27
Food	116	0	0	2	7	10	2	21	378	33	0	66.43
Eat	16	0	0	0	0	0	0	0	7	778	0	97.13
Window	5	0	0	0	0	0	3	0	0	0	4	33.33
Prec. (%)	44.10	81.48	100	86.21	96.79	83.15	65.38	55.91	88.94	94.76	50	81.21

Table 18: Confusion matrix for continuous data

	NULL	Comb hair	Wash hands	Wash face	Tooth brushing	Make bed	Change clothes	Put blinds up/down	Prepare food	Eat	Open/close window	Recall (%)
NULL	16	9	23	37	0	21	20	37	50	20	8	6.64
Hair	1	47	6	0	0	0	0	0	0	0	0	87.04
Hands	0	0	32	0	0	0	0	0	0	0	0	100
Face	1	0	4	29	0	0	0	0	0	0	0	85.29
Tooth	2	0	4	17	244	0	0	0	0	0	0	91.39
Bed	0	0	0	0	0	164	0	0	0	0	0	100
Clothes	1	0	0	0	0	0	71	0	0	0	0	98.61
Blinds	0	0	0	0	0	0	0	74	0	0	0	100
Food	0	0	0	0	0	0	0	0	569	0	0	100
Eat	0	0	0	0	0	0	0	0	0	801	0	100
Window	0	0	0	0	0	0	0	0	0	0	12	100
Prec. (%)	76.19	83.93	46.38	34.94	100	88.65	78.02	66.67	91.92	97.56	60	88.75

Table 19: Confusion matrix for continuous and discrete data

7. Conclusion and Outlook

This thesis deals with human activity recognition in Ambient Assisted Living (AAL). A MPU-9150 sensor is used to record data, which includes a 3-axis gyrometer and a 3-axis accelerometer. This raw data are preprocessed and split into test and training data. Later on features are extracted. Based on these features a continuous Hidden Markov model (cHMM) is constructed.

The cHMM model is validated with provided data and results from Bulling *et al.* [10] and Anguita *et al.* [4]. Afterwards different experiments were accomplished to get the best results.
These experiments contain feature selection, training and test data splitting, impact of filters, sensor selection, impact of splitting single activities and insertion of additional discrete data. The outcome shows, that using only accelerometer data with mean, variance and correlation leads to the best results. The conclusion is that gyrometer data are unneeded for good results and the usage of filters does not really contribute to any significant improvement in this thesis. The activity splitting with extra HMM improves the results, but is not worth the drawback of slower runtime. The most important outcome of the experiments is, that the combination of discrete and continuous data considerably improves the results.
The experiments conclude with a comparison between a cHMM and a k-nearest neighbors classifier (kNN). The outcome of this experiment show the greater importance of HMMs in human activity recognition, in comparison to the not so accurate kNN classifier .

The results of the experiments have to be treated with caution, as the dataset is not big enough to get general statements and is only recorded from one person. Another drawback of the data is, the recording session. It is not recorded in one session, but instead in different sessions with breaks inbetween. Still real environment application is possible.

Further research should focus on the combination of discrete data from binary sensors and continuous data from wearable sensors. The research in this direction will lead to a more robust and trustable model.
Another approach to get more accurate outcomes is, to record a bigger dataset with more different activities and to include more people in the study. This broader consideration will maybe change the results, but definitely lead to results which allow to imply more general statements.

References

[1] G. Abowd and E. Mynatt. Designing for the human experience in smart environments. *Smart Environments: Technology, Protocols, and Applications*, pages 153 – 174, 2004. New York: Wiley.

[2] Md. Atiqur Rahman Ahad. *Low-level Image Processing for Action Representations*, volume 5 of *Atlantis Ambient and Pervasive Intelligence*, pages 9–37. Atlantis Press, 2011.

[3] M. Alwan, P.J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder. A Smart and Passive Floor-Vibration Based Fall Detector for Elderly. In *Information and Communication Technologies, 2006. ICTTA '06. 2nd*, volume 1, pages 1003–1007, 2006.

[4] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition using Smartphones. In *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, 2013. Bruges, Belgium.

[5] U. Anliker, J.A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E.B. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic. AMON: a wearable multiparameter medical monitoring and alert system. *Information Technology in Biomedicine, IEEE Transactions on*, 8(4):415–427, Dec 2004.

[6] Statistik Austria. Bevölkerung. http://www.statistik.at/web_de/statistiken/bevoelkerung/. Accessed: 2015-02-22.

[7] Statistik Austria. Bevölkerungspyramide am 1.1.2014 nach Staatsangehörigkeit. http://www.statistik.at/web_de/statistiken/bevoelkerung/bevoelkerungsstruktur/bevoelkerung_nach_alter_geschlecht/index.html. Accessed: 2015-02-21.

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

[9] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[10] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Computing Surveys*, 46(3):33:1–33:33, 2014.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[12] H. Foroughi, A. Naseri, A. Saberi, and H.S. Yazdi. An eigenspace-based approach for human fall detection using Integrated Time Motion Image and Neural Network. In *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, pages 1499–1503, Oct 2008.

[13] Zoubin Ghahramani. Unsupervised Learning. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 72–112. Springer Berlin Heidelberg, 2004.

[14] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[15] Jia-Wei Han, Jian Pei, and Xi-Feng Yan. From sequential pattern mining to structured pattern mining: a pattern-growth approach. *Journal of Computer Science and Technology*, 19(3):257–279, 2004.

[16] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. ITPro collection.; Morgan Kaufmann series in data management systems. Morgan Kaufmann Publishers, 3rd edition, 2012.

[17] Enamul Hoque and John A. Stankovic. AALO: Activity recognition in smart homes using Active Learning in the presence of Overlapped activities. In *PervasiveHealth*, pages 139–146. IEEE, 2012.

[18] Duy Tam Gilles Huynh. Human Activity Recognition with Wearable Sensors. *Darmstadt*, 2008.

[19] InvenSense Inc. InvenSense MotionFit^{TM} 5.1 SDK User Guide Release 1.0. http://invensense.com/mems/gyro/documents/AN-MPU-9150IMF%20InvenSense %20MotionFit%20SDK%20Quick%20Start%20Guide.pdf, September 2012. Accessed: 2015-02-06.

[20] InvenSense Inc. MotionFit^{TM} Wireless Developer Kit. http://invensense.com/mems/gyro/ documents/PB-MPU-9150IMF%20MotionFit%20Wireless %20Developer%20Kit%20Product %20Brief.pdf, 2012. Accessed: 2015-02-06.

[21] InvenSense Inc. MPU-9150 Product Specification Revision 4.3. http://www.inertialelements.com/docs/PS-MPU-9150A-00v4_3.pdf, September 2013. Accessed: 2015-09-07.

[22] Murphy Kevin and Matt Dunham. Probabilistic modeling toolkit. https://github.com/probml/pmtk3, December 2011. Accessed: 2015-08-13.

[23] Rinat Khusainov, Djamel Azzi, Ifeyinwa E. Achumba, and Sebastian D. Bersch. Real-Time Human Ambulation, Activity, and Physiological Monitoring: Taxonomy of Issues, Techniques, Applications, Challenges and Limitations. *Sensors*, 13(10):12852–12902, 2013.

[24] Wazir Muhammad Laghari, Mohammad Usman Baloch, Muhammad Abid Mengal, and Syed Jalal Shah. Performance Analysis of Analog Butterworth Low Pass Filter as Compared to Chebyshev Type-I Filter, Chebyshev Type-II Filter and Elliptical Filter. *Circuits and Systems*, 5(9):209–216, 2014.

[25] Chin-Feng Lai, Sung-Yen Chang, Han-Chieh Chao, and Yueh-Min Huang. Detection of Cognitive Injured Body Region Using Multiple Triaxial Accelerometers for Elderly Falling. *Sensors Journal, IEEE*, 11(3):763–770, March 2011.

[26] Muhammad Mubashir, Ling Shao, and Luke Seed. A survey on fall detection: Principles and approaches . *Neurocomputing*, 100(0):144 – 152, 2013. Special issue: Behaviours in video.

[27] Bogdan Muset and Simina Emerich. Distance Measuring using Accelerometer and Gyroscope Sensors. *Carpathian Journal of Electronic and Computer Engineering*, 5:83–86, 2012.

[28] Andrew Y. Ng and Michael I. Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, 2002.

[29] Chris D. Nugent, Jit Biswas, and Jesse Hoey. *Activity Recognition in Pervasive Intelligent Environments (Google eBook)*, pages 344 –. Springer, 2011.

[30] Heido McHugh Pendleton and Winfred Schultz-Krohn. *Pedretti's Occupational Therapy - Practice Skills for Physical Dysfunction*, pages 157–173. Elsevier Inc. Mosby, 2013.

[31] S.J. Preece, J.Y. Goulermas, L.P.J. Kenney, and D. Howard. A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data. *Biomedical Engineering, IEEE Transactions on*, 56(3):871–879, March 2009.

[32] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[33] P. Rashidi and D.J. Cook. Keeping the Resident in the Loop: Adapting the Smart Home to the User. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(5):949–959, Sept 2009.

[34] Parisa Rashidi and Alex Mihailidis. A Survey on Ambient-Assisted Living Tools for Older Adults. *Biomedical and Health Informatics, IEEE Journal of*, 17(3):579–590, May 2013.

[35] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity Recognition from Accelerometer Data. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, IAAI'05, pages 1541–1546. AAAI Press, 2005.

[36] Hendrik Schulze. MEMOS: An Interactive Assistive System for Prospective Memory Deficit Compensation-architecture and Functionality. In *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '04, pages 79–85, New York, NY, USA, 2004. ACM.

[37] Muhammad Shoaib, Hans Scholten, and P.J.M. Havinga. Towards physical activity recognition using smartphone sensors. In *10th IEEE International Conference on Ubiquitous Intelligence and Computing, UIC 2013*, pages 80–87, Los Alamitos, CA, USA, December 2013. IEEE Computer Society.

[38] Bundesanstalt Statistik Österreich. Frauen und Männer in Österreich- Statistische Analysen zu geschlechtsspezifischen Unterschieden. *im Auftrag des Bundeskanzleramts - Bundesministerin für Frauen, Medien und Öffentlichen Dienst erstellt*, pages 4–6, 2007.
https://www.bka.gv.at/DocView.axd?CobId=26402, Accessed: 2015-08-13.

[39] G. Van Den Broek, F. Cavallo, and C. Wehrmann. *AALIANCE Ambient Assisted Living Roadmap*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2010.

[40] Kristof Van Laerhoven and Ozan Cakmakci. What Shall We Teach Our Pants? In *ISWC '00: Proceedings of the 4th IEEE International Symposium on Wearable Computers*, page 77, Washington, DC, USA, 2000. IEEE Computer Society, IEEE Computer Society.

[41] Iain L. MacDonald Walter Zucchini. *Hidden Markov Models for Time Series: An Introduction Using R*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Chapman and Hall/CRC, 1st edition, 2009.

[42] Jie Wan, C. Byrne, G.M.P. O'Hare, and M.J. O'Grady. Orange alerts: Lessons from an outdoor case study. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pages 446–451, May 2011.

[43] Ge Wu and Shuwan Xue. Portable Preimpact Fall Detector With Inertial Sensors. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 16(2):178–183, April 2008.

[44] Zhe Xu, Travis Deyle, and Charles C. Kemp. 1000 Trials: An empirically validated end effector that robustly grasps objects from the floor. In *in Proc. IEEE Int. Conf. Robot. Autom*, pages 2160–2167, 2009.

List of Figures

List of Tables