Universität Rostock

Traditio et Innovatio

Activity, Context, and Plan Recognition with Computational Causal Behaviour Models

Dissertation
to obtain the academic degree of
Doktor-Ingenieur (Dr.-Ing.)
of the Faculty of Computer Science and Electrical Engineering
at the University of Rostock

submitted by
Dipl.-Inf Frank Krüger
born on 19.12.1980 in Rostock, Germany
from Rostock

Date of submission: 29.06.2016
Date of defence: 26.09.2016

1st reviewer: Prof. Dr.-Ing. Thomas Kirste, Universität Rostock
2nd reviewer: Prof. Dr. rer. nat. habil Adelinde Uhrmacher, Universität Rostock
3rd reviewer: Prof. Dr.-Ing, Frank Ortmeier, Otto-von-Guericke-Universität Magdeburg

urn:nbn:de:gbv:28-diss2018-0009-0

To my family.

Abstract

As computers are becoming more and more a part of our everyday life, the vision of Mark Weiser about ubiquitous computing becomes true. One of the core tasks of such devices is to assist the users in achieving their goals. To do this, the assistive system has to have knowledge about the current situation as well as the user's goal. Such knowledge allows the assistive system to provide strategies to support the users in achieving their goals beginning from the current situation. A GPS navigation device is a simple, yet well known instance of such an assistive system. It recommends a route based on the current location and the manually specified goal.

Obviously, effective assistance can only be provided if accurate knowledge about the user's situation and his goal is available. This requires to reason about the actions of the user and to cope with uncertainties that are inherent to human behaviour. The problem becomes even harder, as in real world settings, users cannot be observed directly but through sensors that introduce noise and ambiguity as additional sources of uncertainty. Several applications in the literature showed that probabilistic methods can be used to infer the required information from sensor data. However, massive amounts of training data are needed in order to train classifiers to achieve good recognition rates. This is expensive and prevents trained models from being reused.

Recently, researchers employed models of human behaviour in order to reduce the need for training data. These models are generalisable – they allow the specification of human behaviour without the need for training samples. To this end, these models can be reused in different settings. While these models allow the synthesis of probabilistic models, only few attempts have been made to assess their capabilities with respect to low level sensors such as accelerometers. In fact, different researchers stated that inferring high level knowledge about the user from low level sensor data is an open research topic.

To address the above problems, objective of this thesis is to answer the question *"how to achieve efficient sensor-based reconstruction of causal structures of human behaviour in order to provide assistance?"*. To achieve that, in the first step the meaning of this question is analysed and requirements for an inference system are derived. A review of the literature is then conducted and a meta analysis is performed to assess the capabilities of the different approaches and the complexity of their evaluation setting. The results of this analysis show that none of the approaches from the literature satisfies all requirements.

To answer the research question, the concept of Computational Causal Behaviour Models (CCBMs) is introduced. CCBM allows the specification of human behaviour by means of preconditions and effects and employs Bayesian filtering techniques to reconstruct action sequences from noisy and ambiguous sensor data. Furthermore, a novel approximative inference algorithm – the Marginal Filter – is introduced. The Marginal Filter is specifically tailored for categorical state spaces, which are generated by CCBM. To investigate the capabilities with respect to recognition performance and reusability, different experiments are then conducted. Each experiment addresses different aspects of the research question. A detailed analysis of the results of these experiments shows that CCBM is able to achieve good recognition rates. Moreover, the Marginal Filter is shown to outperform the standard method for approximative Bayesian inference – the Particle Filter. Furthermore, it is shown that CCBM satisfies the requirements.

Keywords: activity recognition, plan recognition, probabilistic inference

Zusammenfassung

Die Vision des „Ubiquitous Computing“ von Mark Weiser wird langsam wahr – immer mehr Computer umgeben uns in Alltagsgegenständen. Eine Hauptaufgabe dieser Geräte ist es, Nutzer dabei zu unterstützen ihre Ziele zu erreichen. Dabei ist es notwendig, dass Assistenzsysteme in der Lage sind, Informationen über die aktuelle Situation des Nutzers und seine Ziele zu erfassen. Dieses Wissen wird dann von den Assistenzsystemen verwendet, um Strategien zum Assistieren des Nutzers entwickeln. Ein GPS Navigationsgerät ist ein sehr einfaches Beispiel für solche Assistenzsysteme. Basierend auf der aktuellen Position und einem, wenn auch eingegebenen, Ziel, ist es in der Lage dem Nutzer eine Route vorzuschlagen.

Eine wichtige Voraussetzung für das Umsetzen der Assistenzfunktionalität ist sicheres Wissen um die aktuelle Situation und potentielle Ziele des Nutzers. Um solche Informationen bereitzustellen ist es erforderlich, dass das Assistenzsystem Schlussfolgerungen aus den Handlungen des Menschen zieht. In der Realität ist es nicht möglich Menschen, direkt zu beobachten sondern unter Verwendung von Sensoren, deren Daten verrauscht und mehrdeutig sind. Dadurch wird die Erfassung der notwenigen Informationen weiter erschwert. In der Literatur konnte gezeigt werden, dass probabilistische Methoden in der Lage sind, Informationen über die aktuelle Situation und das Ziel des Nutzers aus unscharfen Sensordaten zu schließen. Es sind große Mengen an Trainingsdaten notwendig, um Klassifikatoren so zu trainieren, dass gute Erkennungsraten erreicht werden. Das Erzeugen von Trainingsdaten ist jedoch teuer. Außerdem lassen sich durch Trainingsdaten keine allgemeingültigen Schlüsse ziehen, was die Wiederverwendbarkeit so trainierter Modell verhindert.

Modelle menschlichen Handeln wurden in letzter Zeit von verschiedenen Forschern verwendet, um die Notwendigkeit von Trainingsdaten zu reduzieren. Solche Modelle erlauben eine generalisierbare Beschreibung ohne auf Trainingsdaten angewiesen zu sein. Dadurch können die resultierenden Modelle in verschiedenen Szenarien wiederverwendet werden. Obwohl diese Modelle das Erzeugen probabilistischer Modelle erlauben, wurden nur wenige Versuche unternommen, ihre Eigenschaften in Hinblick auf die Verwendung von „low-level“ Sensoren wie Akzelerometer zu untersuchen. Verschiedene Forscher halten die Verbindung von „high-level“ Inferenz mit „low-level“ Sensoren für ein offenes Forschungsfeld.

Diese Dissertation zielt auf die Beantwortung der folgenden Frage ab:*„Wie kann man die effiziente sensorbasierte Rekonstruktion kausaler Strukturen menschlichen Handelns zum Zwecke der Assistenz erreichen?“*. Dazu werden im ersten Schritt Anforderungen an ein Inferenzsystem hergeleitet. Danach wird eine detaillierte Analyse verwandter Arbeiten durchgeführt und mithilfe einer Metaanalyse die Eigenschaften aktueller Ansätze und deren Evaluationsszenarien erfasst. Diese Analyse zeigt, dass keiner der aktuellen Ansätze alle Anforderungen erfüllt.

Um die Forschungsfrage zu beantworten, wird das System der „Computational Causal Behaviour Models“ vorgestellt. Es erlaubt die Spezifikation menschlichen Verhaltens durch die Verwendung von Vorbedingungen und Effekten. Bayes'sche Inferenz wird verwendet um Aktionssequenzen aus unscharfen Sensordaten zu rekonstruieren. Darüber hinaus wird ein neuartiger Algorithmus zur approximativen Inferenz vorgestellt – der Marginale Filter. Der Marginale Filter ist entworfen, um den speziellen Eigenschaften kategorialer Zustandsräume, wie sie von „CCBM“ erzeugt werden, gerecht zu werden. Mittels verschiedener Experimente wird die Erkennungsgenauigkeit und die Wiederverwendbarkeit solcher Modelle untersucht. Dabei behandelt jedes der Experimente andere Teilaspekte der Forschungsfrage. Die Analysen der Ergebnisse der Experimente zeigen, dass „CCBM“ in der Lage ist, gute Erkennungsgenauigkeiten zu erreichen. Außerdem wird gezeigt, dass der Marginale Filter die Standardmethode für approximative Bayes'sche Inferenz in Hinblick auf die Erkennungsgenauigkeit übertrifft. Zuletzt kann auch die Wiederverwendbarkeit in allen Experimenten nachgewiesen werden.

Keywords: Aktivitätserkennung, Planerkennung, probabilistische Inferenz

Acknowledgements

When I joined MMIS almost seven years ago, I had no idea about what it means to do research in artificial intelligence. Since then I learned a lot about this topic and was finally able to write this thesis. During the last years, there were a lot of people who supported me and enabled me to write this thesis. Now, I would like to thank all of them.

In particular, I would like to thank my supervisor Prof. Thomas Kirste who enabled me to write this thesis and provided such a good research atmosphere. I also would like to thank Prof. Lin Uhrmacher and Prof. Frank Ortmeier, who accepted to review this thesis. Furthermore, I would like to thank my colleagues and friends for their support during this time. Special thanks go to my family, my wife, and my daughters.

Contents

List of Abbreviations XV

1. Introduction and Motivation 1

1.1. Introduction 1
1.2. Motivation and Requirement Analysis 3
1.2.1. Motivational Examples 3
1.2.2. Requirements analysis 4
1.3. Problem Statement 6
1.4. Contribution and Results 7
1.5. Outline 8

2. Computational State Space Models 9

2.1. Activity and Plan Recognition 10
2.1.1. The Problem of Behaviour Recognition 10
2.1.2. Activity Recognition 12
2.1.3. Plan Recognition 14
2.1.4. Actions and States 17
2.1.5. Computational State Space Models 19
2.2. Related Work 19
2.2.1. Classification Scheme 20
2.2.2. Classification Results 24
2.2.3. Policy Recognition in the Abstract Hidden Markov Model 25
2.2.4. Learning and Inferring Transportation Routines 26
2.2.5. Synthesising Generative Probabilistic Models for High-Level Activity Recognition 28
2.2.6. Action Understanding as Inverse Planning 29
2.2.7. Goal Recognition over POMDP 30
2.2.8. Accommodating Human Variability in Human-Robot Teams 32
2.2.9. Location-Based Reasoning about Complex Multi-Agent Behavior 33
2.2.10. Summary 34

3. Computational Causal Behaviour Models 35

3.1. Statistical Model 36
3.1.1. Action Execution Model 38
3.1.2. Action Start Time and Action Duration Models 40
3.1.3. Goal Selection Model 41
3.1.4. Action Selection Model 42
3.1.5. Observation Model 44
3.2. Inference Algorithms 45
3.2.1. The Particle Filter 46

3.2.2. The Marginal Filter ... 47
3.3. The Computational Causal Behaviour Model Toolbox ... 50
3.3.1. Modelling Language ... 51
3.3.2. CCBM Inference Tools ... 54

4. Causally Correct Annotation 55
4.1. Annotation of Human Behaviour ... 55
4.2. Model-based Semantic Annotation for Human Behaviour ... 57

5. Methods 63
5.1. Research Questions ... 64
5.2. Empirical Data ... 66
5.3. Experimental Procedure ... 67
5.4. Evaluation Methods ... 70
5.4.1. Different Estimation Tasks ... 70
5.4.2. Evaluation of Action Recognition ... 71
5.4.3. Evaluation of Contextual Information ... 73
5.4.4. Evaluation of Goal Recognition ... 74
5.4.5. Assessing the Size of Effects ... 74

6. Experiments 75
6.1. Experiment X1: Three Person Meeting ... 76
6.1.1. Objective ... 77
6.1.2. Trial Setting ... 77
6.1.3. Experimental Setup ... 80
6.1.4. Results ... 84
6.1.5. Discussion ... 88
6.2. Experiment X2: Kitchen Task ... 90
6.2.1. Objective ... 90
6.2.2. Trial Setting ... 91
6.2.3. Experimental Setup ... 94
6.2.4. Results ... 97
6.2.5. Discussion ... 105
6.3. Experiment X3: Indoor Localisation ... 107
6.3.1. Objective ... 109
6.3.2. Trial Setting ... 109
6.3.3. Experimental Setup ... 112
6.3.4. Results ... 114
6.3.5. Discussion ... 117

7. Discussion and Conclusion 119
7.1. Summary ... 119
7.2. Discussion ... 123
7.3. Limitations ... 124
7.4. Outlook and Future Research ... 124

A. Details 151
A.1. Additional Information about Related Work ... 151
A.2. Example ... 153
A.3. Construction of Hidden Markov Models ... 154

A.4. Notes on Intention Recognition 154

B. Additional Information about Experiments **157**
B.1. Experiment X1 157
B.2. Experiment X2 159
B.3. Experiment X3 165

List of Figures

2.1. Graphical representation of three different types of classifier 11

3.1. Graphical model used in CCBM 37
3.2. Example of an action duration PDF and the resulting CDF. 41
3.3. Example of an action duration PDF and the resulting action duration function. . . 41
3.4. Schematic illustration of the S state, X state and representation unit assignment in Particle Filter and Marginal Filter. 49

4.1. The proposed workflow for semantic annotations of human behaviour. 58
4.2. Specification of the annotation LTS action take 60
4.3. Screenshot of the annotation procedure using ELAN. 60

5.1. Confusion matrix for multi-class classification 71

6.1. Overview of the Smart Appliance Lab. 78
6.2. Graphical representation of the transition matrix of the baseline HMM for the meeting experiment 81
6.3. The action start-presentation. 82
6.4. Overview of the accuracies of the baseline HMM for the meeting experiment. 84
6.5. Overview of the results of all CCBM configurations for the meeting experiment. . 85
6.6. Convergence points for the baseline HMM and the τ_c-based CCBM configurations. 86
6.7. The goal PDF for M17 of the HMM baseline classifier and τ_c-based CCBM classifiers 87
6.8. Relationship of activity recognition and goal recognition of CCBM and baseline classifiers 88
6.9. The physical trial setup of the kitchen experiment 92
6.10. Accuracies of the baseline classifier per subject. 98
6.11. Boxplots of the accuracies of all CCBM configuration for the kitchen experiment. 99
6.12. Accuracy comparison of selected CCBM configurations to HMM, by subject and filter method. 100
6.13. Confusion matrix based performance measures per class. 100
6.14. Interaction plots for the significant interactions. 102
6.15. Jensen Shannon distance and accuracies for different values for Target and Distance. 104
6.16. Estimating the probability of state properties. 105
6.17. Comparison of different performance measures sensitive for causality. 105
6.18. The problem of simultaneous identification of multiple persons in partially observed environments. 108
6.19. The spatial layout of the indoor localisation trial setting 109
6.20. Graphical illustration of the location sequence of the first repetition of person A1 in trial T1 111

6.21. Overview of the recognition accuracies for the location-based estimate of all classifiers. . . . 115
6.22. Overview of the activity recognition (AR) accuracies the CCBM-based classifier. 116
6.23. Linear models fitted to predict accuracies of the different classifier from the number of involved persons. . . . 117

B.1. The probability of each agent for being seated at each time-step. . . . 158
B.2. Frequencies of actions in the dataset. . . . 159
B.3. Linear Models fitted to predict the normalised remaining time to goal (RT) from the normalised goal distances for different goal distance approximations. . . . 159
B.4. Effect of scrambling on (expected) log probability of observations vs. normalised relative run position. . . . 160
B.5. Confusion matrices for QDA, HMM, and CCBM. . . . 161
B.6. Interactions between Mode, Distance, and Weight. . . . 161
B.7. Interactions between Observations, Distance, and Weight. . . . 162

List of Tables

2.1. Factors for analysing related work. 21
2.2. Overview of related work and evaluation studies. 23
2.3. Evaluation of the requirements of selected related work. 34

4.1. Examples of annotation schemes 56
4.2. Example of the results of the first annotation step. 58
4.3. Example of the results of the second annotation step. 59
4.4. Example of the results of the third step of the annotation process. 59

5.1. Overview of the experiments and the research questions. 65
5.2. Modelling factors and their meaning for experimental configurations. 70
5.3. Overview of different effect size measures and their interpretation. 74

6.1. Different types of agendas including their frequencies. 78
6.2. Overview of the compliant meeting. 79
6.3. Factors and levels for the Meeting experiment. 83
6.4. Performance comparison of different CCBM configurations with corresponding HMM configuration. 85
6.5. Mean precision per goal for each classifier. 86
6.6. Task script of the kitchen trial 93
6.7. Factors and levels for experimental configurations. 96
6.8. Cohen's κ and overall accuracies for selected configurations. 101
6.9. Median SpU and XpS values and ratios, across all runs of the kitchen experiment. 103
6.10. Median values and IQR (Q1,Q3) for Accuracy and JSD for predicate estimation. 104
6.11. Overview of the different agent configuration. 110
6.12. State space size for the different problems. 113
6.13. Factors and levels for the indoor localisation experiment. 114
6.14. Overview of the mean recognition accuracies of the baseline classifiers. 115
6.15. Linear models to predict the accuracy from the number of involved persons. 116

A.1. Overview of plan recognition aproaches. 151
A.2. Overview of AR approaches. 152

B.1. Extract of the preprocessed position data from the first meeting dataset $\mathcal{D}_1$. 157
B.2. Selected probability density functions for each action including their parameters. 158
B.3. Performance comparison of different CCBM configurations with appropriate HMM configuration. 159
B.4. Value domains of location slot by domain object for the kitchen experiment. 160
B.5. The domain objects and their slots for the kitchen experiment. 162
B.6. Detailed properties of linear models in Figure B.3 162

B.7. Action sequence of subject S1 (aLTS annotations). 163
B.8. Duration models selected for action classes. 164
B.9. Significance of effects of CCBM configuration factors on Accuracy, using 216 CMf / CPf configurations. (2 modes, 3 observations, 3 distances, 6 weights, 2 durations.) 164
B.10. Overview of the annotation sequence of the first iteration for one participating person. 165
B.11. Comparison of the CCBM to the HMM recognition accuracies for the different trial runs. 166

List of Abbreviations

ACT-R Adaptive Control of Thought – Rational 20
ADL activity of daily living 12
aLTS annotation LTS 57
AR activity recognition XII
CCBM Computational Causal Behaviour Model I
CDF cumulative distribution function 41
CMUMMAC Carnegie Mellon University Multimodel Activity 57
CSSM Computational State Space Model 7
DBN dynamic bayesian network 26
DSL domain specific language 18
DT decision tree 13
DTW dynamic time warping 73
GPS Global Positioning System 3
HMM Hidden Markov Model 11
IMU inertial measurement unit 13
iLTS inference LTS 61
JSD Jensen-Shannon distance 74
LTS labelled transition system 17
MAP maximum a-posteriori 33
MDP Markov Decision Process 29
MF Marginal Filter 8
NB Naïve Bayes 11
PDDL Planning Domain Definition Language 20
PDF probability density function 31
PF Particle Filter 8
PIR passive infrared 25
POMDP Partially Observable Markov Decision Process 31
QDA quadratic discriminate analysis 94
PR plan recognition 10
rANOVA repeated measures analysis of variance 74
RBPF Rao-Blackwellized Particle Filter 26
RFID radio-frequency identification 92
STRIPS Stanford Research Institute Problem Solver 19
SVM support vector machine 10

1 Introduction and Motivation

"It is a capital mistake to theorise before one has data" – Sherlock Holmes

SYNOPSIS: *This first chapter describes the aim of this thesis. For this reason, in the first step a list of requirements is derived by briefly analysing two instances of assistive systems. In the next step, a detailed problem statement is provided. Finally, it is described how this thesis contributes to addressing the states problems.*

CHAPTER SOURCES: *Parts of this Chapter have been previously published in the following publication(s):*

- *Towards Creating Assistive Software by Employing Human Behavior Models [129]*

1.1. Introduction

With the rise of ubiquitous computing [246] in our everyday life, the number of devices surrounding us is steadily increasing. These devices allow for supporting users in everyday activities without being perceived explicitly. Consequently, the number of context aware applications increases. Knowledge about the user's activity, goal, and additional contextual information is crucial for such applications [98]. Example applications can be found in the domains of smart environments [55, 260], security and surveillance [84], man-machine-collaboration [95], and personal assistive systems [99, 179].

According to [120], context aware systems consist of two components. The first component is called intention analysis (or "inference system") and tracks the user's actions. This means it estimates the user's current activity, the information about the environment (referred to as contextual information), possible future actions and the final goal the user is actively trying to achieve. The second component is the strategy synthesis component. This component usually adjusts the environment to the user's needs. Within this thesis, we focus on the first component, namely the inference system.

More precisely, this thesis addresses the question of:

IQ *How to achieve efficient sensor-based reconstruction of causal structures of human behaviour in order to provide assistance?*

To this end, we first analyse the meaning of this question. The word *efficient* means that the inference system under development has to be able to work online. Whenever a new observation about the dynamic system[1] arrives, a new estimate about the system's situation[2] has to be computed. The term "online" does not refer to constraints of actual execution time, as this would require knowledge about the hardware and the application of an realtime operating system, but rather in the sense of computational complexity. Estimating the dynamic system's state online means that computing the estimate for one time-step has to be independent of the length of the observation sequence so far and of the overall length of the observation sequence. In fact, the complexity of computing an estimate only depends on the belief[3] about the current situation and the number of possible situation changes. Inference complexity for the complete observation sequence has to be linear in the number of time-steps[4] T – $\mathcal{O}(T)$. In other words, inference complexity has to be constant for a single observation item. This allows, by adjusting the necessary hardware, to ensure the system's ability to react in an appropriate time interval.

The term *sensor-based* indicates that the inference system has to draw conclusions about the human protagonist by means of sensors. Knowledge about actions, executed by a dynamic system, can typically not be obtained directly[5], but has to be observed through sensors. Sensors measure physical properties and their changes, which are affected by the actions of the dynamic system. Sensor data typically does not allow to directly conclude the dynamic system's state as it is subject to noise. Additionally, sensor data is typically ambiguous. Answering the initial question **IQ** thus means handling the uncertainty inherent to the sensor observation [46, 237].

As sensors typically do not allow to draw conclusions about the actions of a dynamic system directly, *reconstruction* means estimating the sequence of states or actions of the dynamic system. Based on sensory observation, the inference system has to compute the most likely sequence of actions.

As the term *causal structures* indicates, throughout this thesis we focus on causal action sequences. This means that there are causal links between the actions of the dynamic system and the internal state. Actions, on the one hand, influence the state of the system but are, on the other hand, influenced by the dynamic system's state. Not all actions can be executed in each state (e.g. a room cannot be left, if the door is closed). Exploiting causal structures allows the definition of restrictions to the dynamic system, which in result can potentially reduce the inference complexity.

This thesis focusses on *human behaviour*. Therefore, the dynamic system under observation is a human protagonist. Consequently, the answer to the initial question **IQ** also includes handling

[1] Here, we refer to the system under observation as dynamic system, which at each time is in a state $s \in S$ represented by a point in the state space S. A function $f : S \to S$ allows the system to change that state. In the following this function is referred to as "action".

[2] Situation reflects the dynamic system's overall state.

[3] Belief, in this context, stands for the inference system's representation of the current state of the dynamic system. In the scope of this thesis the belief is represented by a probability distribution over states.

[4] The number of time-steps T refers to the number of observation items of the observation sequence rather than to an actual time.

[5] Within the scope of this thesis, we use the term "direct observation" to refer to obtaining information about a dynamic system without noise or ambiguity.

uncertainties inherent to human behaviour. Additionally, as the term behaviour suggests, we are especially interested in the reconstruction of sequences of actions, the human protagonist executes in order to achieve a goal. Within this thesis, we assume the human to be trying to reach a previously set goal state. However, this does not mean that the human is a completely rational agent [200, p.36], as human protagonists do not always choose the best action with respect to reaching this goal [208].

In the remainder of this chapter, the initial question **IQ** is further substantiated. Firstly, requirements for the inference system are derived. Secondly, a detailed description of the problems that are targeted in this thesis is presented. Finally, the contribution and the results of the thesis are highlighted.

1.2. Motivation and Requirement Analysis

This section first introduces two examples of assistive systems. Based on these examples, requirements for an inference system are then identified and later used to derive a classification scheme for related literature.

1.2.1. Motivational Examples

Consider the following example of a Global Positioning System (GPS) navigation device: a GPS navigation device provides a route from the current location to a target location. Before starting, the target location has to be entered in the device. Then, according to the needs of the user, the device computes the most appropriate route from the current location to the defined target location. During the trip, the GPS navigation device keeps track of the current location and checks whether the car is on the correct route or not. Before the car arrives at a street crossing, the navigation device provides assistance to the driver by suggesting further directions. If, either by accident or by the driver's choice, the car leaves the precomputed route, the GPS navigation device recognises the deviation and provides appropriate assistance. This typically means that the device first suggests the driver to go back to the originally precomputed route and later starts adjusting the route by re-computing it with the current location. Additionally, typical GPS navigation devices are able to inform users about potential issues (e.g. exceeding the speed limit or an upcoming traffic jam).

The GPS navigation device is a very simple example of an assistive system. During the entire trip the device has to keep track of the current situation of the car. This is done by analysing data from the internal GPS tracking module. Although GPS is not able to provide an exact location estimate, the navigation device is able to refine this estimate by assumptions about the current situation (e.g. cars usually move along streets instead of fields or houses). Single measurement errors determined by the GPS system are corrected by the navigation system.

Now, consider the more fictional example of a kitchen assistance device: The device observes the user within the kitchen environment by means of sensors. When the user decides to prepare a meal, he[6] has to select the recipe from the list of recipes provided by the device. A sequence of actions (plan) is precomputed by the device according to the preferences of the user. The device starts providing assistance in executing the selected task. During the entire procedure the kitchen assistant keeps track of the current situation. Whenever a sub-task is completed the device suggests the next task and monitors whether the user executes it correctly. If the user decides to change the order of some sub-tasks, the device notices the deviations from the original plan and adjusts the assistance to the change. If, on the other hand, the user forgets

[6] To simplify the discussion, the personal pronoun "he", as well as the possessive pronoun "his" are used as a substitution of "he / she", and not as an indication that the person in question is a male.

to add some necessary ingredients, the device refers the user to the selected recipe and reminds him to add the ingredients. The kitchen assistance device helps novice users as well as impaired users, such as people suffering from cognitive decline, during the execution of tasks in everyday life. A more sophisticated version of the kitchen device might even attempt to recognise the recipe the user is currently trying to follow from the actions the user is executing without requiring the user to select the recipe beforehand.

Similar to the navigation device, the kitchen assistance device requires to observe the user by means of sensors. However, using location estimating sensors only is not enough, as they do not provide enough information about the manipulated objects. Similar to the navigation device, the kitchen device has to cope with uncertainties inherent to sensors.

The provided examples are typical instances of assistive systems. Both can be divided into the inference system, which estimates the current situation, and the strategy component, which actually provides assistance. In the following, these two instances are used to derive general requirements for the inference system component of assistive systems.

1.2.2. Requirements analysis

According to Indulska and Henricksen [105], a context aware system, which assistive systems are a type of, has to fulfil the following five requirements:

IH1 **Support for imperfect context information**: A context aware system has to cope with incomplete, imprecise and ambiguous information. The examples provided above use sensors as source of information.

IH2 **Support for context histories**: A context aware system requires information about past and future time-steps. Both introductory examples provide a plan from the initial state to the selected goal state. This plan contains both, information about past actions and possible future actions.

IH3 **Support for software engineering**: A context model should be introduced in early stages of the software development lifecycle. Based on this context model, runtime context models can be produced by refinement. Thus, such context models have to be reusable. In the above mentioned examples, both, the navigation device and the kitchen device, are refined for the specific application scenario (i.e. the map of roads and the particular kitchen layout).

IH4 **Support for runtime querying and reasoning**: The runtime context model has to represent context information at runtime. This, on the one hand, means that the context aware system has to provide the information timely. On the other hand, this means the context aware system has to maintain all necessary information.

IH5 **Support for interoperability**: The context aware system has to be able to cooperate with components that were not known at design time. This also includes sensors that can differ in different application scenarios. The context model has to represent the rich state space of the dynamic system to ensure that context information provided by previously unknown components (e.g. sensors) can be incorporated in the runtime model.

Based on this list of requirements for context aware systems introduced by Indulska and Henricksen [105] and an analysis of the introductory examples, the following requirements for the inference system can be derived. Each of these requirements is provided in the following. Furthermore, the method and the criterium used to check whether a considered approach satisfies the respective requirement is provided.

R1 **Plan**: The inference system has to provide knowledge about the user's potential future actions and goals. Knowledge about the user's plan can enable to assistive system to

provide appropriate support [98] in reaching the goal. In the above introduced examples, the goal is provided by the user. The potential action sequence leading to this goal, in contrast, is computed by the inference system. A deviation from the plan is noticed by the system and used to recompute the plan. This requirement is directly related to IH2.
Method: Analysis of the inference approach and evaluation of the recognition performance
Criterium: The approach has to be able to estimate the user's action sequence and the final goal with similar recognition performance as baseline classifiers.

R2 **Online**: The system has to be able to provide information online. It is only possible to provide effective assistance, if the system is capable of providing information about the user online. With respect to the navigation device, if the system provides information about future directions too late, the assistance makes no sense at all. Furthermore, it can produce potentially dangerous situations, as some drivers can turn around even on highways resulting in an increased occurrence of wrong-way drivers. Concerning the kitchen assistance device, assistance provided too late can distract and annoy people. If the assistance has to be provided online, also the user's state has to be estimated online. This requirement represents one part of requirement IH4.
Method: Analysis of computational complexity
Criterium: The computational complexity must be independent from the length of the observation sequence to allow the estimation to be online.

R3 **Uncertainty**: The inference system has to cope with uncertainties inherent to sensory observations. Sensors are unable to provide exact information about the observed objects but rather provide noisy and ambiguous statements about the current situation. Processing sensor data to estimate information about the user means to cope with such uncertainties. Both, the navigation device and the kitchen assistance device rely on sensor observations that cannot be guaranteed to be correct. Furthermore, as in the case of the kitchen assistance device, it is often not possible to directly interpret sensor data. This requirement is directly derived from IH1.
Method: Analysis of the inference method
Criterium: The inference method has to be able to cope with ambiguous and noisy sensor data.

R4 **Latent infinity**: The inference system has to cope with a very high (potentially unlimited) number of possibilities. When considering one specific action sequence of the dynamic system under observation, only a finite number of possibilities actually occur. However, as there is no prior knowledge about the finite part, the dynamic system's state space can be considered as infinite. The inference system has to be able to handle this latent infinity during inference. With respect to the initial examples, this means that although the number of streets, crossings, and potential locations is very high (possible infinite), the navigation device has to cope with it. For the kitchen example, the drawers or cupboards, for instance, might contain a very high number of kitchen utensils. If for k utensils, it has to be tracked whether they were taken or not, the inference system has to cope with 2^k difference possibilities. This further increases with an increase of the number of locations. Also, the number of recipes is potentially unlimited. However, the kitchen assistance device has to cope with this (possibly infinite) state space. This requirement is a consequence of IH4 and IH5.
Method: Analysis of the modelling and the inference methods
Criterium: The modelling approach has to be able to construct infinite state spaces. Furthermore, the inference method must not rely on the state space to be explored before runtime.

R5 **Reusability**: The basic functionality of the inference system has to be reusable independently of the actual purpose. This means that only simple adjustments of the system

have to enable its usage in different settings. For instance, the navigation device has to be able to be reused, even if external conditions, like the map of roads, change. Only by loading an updated map, the navigation system is able to provide assistance in previously unknown regions or countries. Similarly, the kitchen assistance device has to be reusable independently of the actual kitchen layout or the available sensor modalities. A kitchen assistance device that has to be adjusted manually to every change is neither practical from economic nor from usability point of view. Several aspects of reusability are discussed below. This requirement reflects IH5.
Method: Proof by demonstration
Criterium: The approach has to provide a modelling formalism that allows to adjust the inference model to the specific needs of the experiment by parameters.

Requirement R5 can further be split up into different aspects. With respect to reusability on the level of the human behaviour model, the following three aspects were identified.

R5.1 **Application domain**: A behaviour model developed for a specific scenario of an application domain has to allow the reuse in a different setting of the same application domain. While for obvious reasons it is not possible to directly reuse the complete context model, *"simple"* changes to the configuration have to be sufficient to adjust the model to the new setting. According to Indulska and Henricksen [105], the runtime model is a refinement of the context model. Refining the context model for different scenarios of the same application domain should neither require the setup of a training phase nor changes of the context model. With respect to the above described examples, necessary refinements have to include a concretisation of the map of roads or the actual layout of the kitchen only.

R5.2 **Sensor data**: The inference system has to be able to reconstruct action sequences from different sources of observations. A change of the sensor modality must neither require the system to be re-developed nor affect the human behaviour model. Especially, the context model has to allow for reuse even if the sensor modality changes from environmental observation (dense sensing)[238, 248] to action observation (wearable sensing)[244, 139] or back. To this end, the human behaviour model should be independent from the actual sensor infrastructure. As described above, fulfilling this requirement demands the usage of rich state spaces.

R5.3 **Number of Persons**: Often, more than one person is involved in an activity. Thus, a context model also has to handle multiple persons. The human behaviour model has to be independent of the actual number of persons. If no changes on the causal level are necessary (a conversation, for example, requires at least two persons), the number of persons must not affect the model. Again, a refinement of the context model has to be sufficient in order to adjust the runtime model to the actual needs.

Based on the five requirements a classification scheme is derived in Section 2.2.1. This classification scheme is then used to categorise related work from the literature (see Table 2.2). In the following, a detailed description of the problem targeted in this thesis is provided.

1.3. Problem Statement

As discussed in Section 1.1, this thesis aims at efficiently reconstructing causal human behaviour from sensor observations to eventually provide assistance. To answer the question **IQ**, an inference system shall be designed that satisfies the five requirements. To this end, the following questions are addressed and answered in this thesis.

Q1 **How do current methods for efficient reconstruction of causal human behaviour from sensor data address the list of requirements?**
Based on the review of the literature, different approaches are identified that satisfy (a subset of) the requirements. To allow comparison of different approaches from the literature with respect to the problem domain targeted within this thesis, a classification scheme is derived. A meta analysis is conducted based on this classification scheme. Then, the most appropriate approaches, with respect to the classification scheme, are selected and discussed in detail. Finally, for each selected approach it is analysed how the satisfaction of the requirements is achieved.

Q2 **How to design an inference system to satisfy the requirements?**
Based on an analysis of the related work and their drawbacks with respect to the list of requirements, an inference system is developed.

Q3 **How can efficient inference be achieved?**
Due to the requirement for latent infinity, efficient approximate inference methods have to be applied. In general the framework of Bayesian filtering allows inference to be done efficiently. Particle filters (also known as sequential importance sampling) are typically applied. However, in order to achieve reasonable recognition rates, a novel filtering method has to be developed that focusses on the problem domain of inference of causal sequences of human behaviour.

Q4 **Is it possible to ensure the annotation to be causally correct?**
To allow an assessment of the quality of the reconstructed action sequences, a comparison with the ground truth has to be performed. When exploiting causal dependencies of actions of human behaviour, it has to be ensured that this ground truth is causally correct. Furthermore, this thesis also targets at the reconstruction of contextual information. For this reason, an annotation methodology is developed that allows to produce a semantic annotation by means of causally correct action sequences.

In the next section, the contributions of this thesis are listed.

1.4. Contribution and Results

The contribution of this thesis is fourfold.

C1 **The concept Computational State Space Models (CSSMs) is introduced.**
CSSMs are a paradigm that summarise the common statistical structure of several recent modelling approaches. It is discussed that CSSMs are in principle able to answer the question **IQ**. Additionally, it is shown that no approach to CSSM exists that satisfies the entire list of requirements. It is argued that, due to the lack of appropriate inference techniques up to now, it could not be shown that CSSMs can handle problems related to question **IQ**. Based on the concept of CSSM, a classification scheme is derived and related work is classified accordingly. This contribution addresses question Q1

C2 **The concept of CCBM is introduced.**
CCBM represents an instance of CSSMs. CCBM employs several aspects of existing related approaches. By using reusable models of human behaviour and the ability of coping with uncertain sensor data and efficient inference, CCBM combines the advantages of recent approaches. It is shown that CCBM satisfies the entire list of requirements. Three different experiments are used to show that the application of CCBM allows to answer the initial question. Here, question Q2 is targeted.

C3 **A novel inference algorithm – the Marginal Filter – is introduced.**
It is discussed that employing causal models by means of preconditions and effects results in discrete state spaces with sparse transition matrices. Furthermore, it is highlighted that

the Particle Filter (PF) – an approximate inference technique that is used predominantly in the literature – suffers from several disadvantages when employed in categorical state spaces. The Marginal Filter (MF) is then introduced to target these disadvantages. The MF specifically tailored for inference in such state spaces. Question Q3 is addressed by this contribution.

C4 **A novel annotation process to semantic annotation is introduced.**
It is discussed that exploiting the causal structure of human behaviour requires the annotation to be causally correct in the first place. For this reason an annotation process is introduced allowing the annotation of causally correct action sequences by semantic means. This contribution provides an answer to the question Q4.

The experiments in Chapter 6 show that CCBM is able to reconstruct the action sequence of the human protagonists at the same level as the baseline classifiers while satisfying the requirements at the same time. By providing empirical evidence, it is shown that CCBM allows to handle state spaces that are by orders of magnitude larger than those of related approaches and satisfies the requirements at the same time. In summary, the results of this thesis show that CCBM indeed provides an answer for the initial question **IQ**.

1.5. Outline

The remainder of this thesis is structured as follows: Chapter 2 provides an overview of the current state of the art. To this end, an overview of both, low-level reasoning as well as high-level reasoning techniques is given. It it argued that both are essential premises for answering the initial research question **IQ**. The paradigm of Computational State Space Model is introduced and used to classify the related work. Finally, selected related work representing preparatory work for this thesis is discussed.

Chapter 3 presents Computational Causal Behaviour Models as method used within this thesis. A detailed description of CCBM, including the statistical model and inference algorithms is provided. The extent to which the requirements are fulfilled is shown. Finally, a brief overview of the CCBM toolbox is provided.

Chapter 4 concentrates on the problem of providing annotation of reasonable quality. For this purpose, first, the need for annotation in general and causally correct annotation in particular are discussed. Then, a novel annotation process is introduced that allows to create causally correct annotation.

Chapter 5 gives an overview of the methods used to investigate the initial research question **IQ**. In the first step, the question **IQ** is subdivided into research questions that target single aspects of question **IQ**, which can be answered by experiments. Then, a brief discussion about the need of empirical data in favour of simulated data is presented. The general experimental procedure is described. Finally, a list of evaluation methods is presented that are used to answer the research questions.

Chapter 6 provides the experiments that were conducted in order to answer the research questions. For each experiment, methods, results, and discussion are provided.

Finally, the last chapter (Chapter 7) discusses in how far the initial research question **IQ** is answered. An outlook to possible future work is provided.

2
Computational State Space Models

"All models are wrong, but some are useful." – George E. P. Box

Synopsis: *This chapter provides an overview of related work in the domains of activity and plan recognition. The basic concepts of activity, goal and plan recognition are introduced. It is argued that neither activity recognition nor plan recognition are sufficient to answer the initial question. The concept of Computational State Space Models is introduced and described. Based on this concept, related work on activity, action, context, goal, and plan recognition is classified. Finally, a comprehensive review and classification of related work is presented.*

Chapter Sources: *Parts of this Chapter have been previously published in the following publication(s):*

- *Computational State Space Models for Activity and Intention Recognition. A Feasibility Study [133]*
- *Where are My Colleagues and Why? Tracking Multiple Persons in Indoor Environments [132]*
- *Towards Creating Assistive Software by Employing Human Behavior Models [129]*
- *Plan Synthesis for Probabilistic Activity Recognition [131]*

The following chapter provides an overview of related work. In the first step an overview of the problem of human behaviour recognition is given and the basic concepts are introduced. Then, the basic principle of AR is introduced and it is argued that for several reasons AR is not able to answer the initial question. Goal and plan recognition are then introduced and it is discussed that these concepts provide mechanisms for high-level reasoning. Again, it is argued that the sole application of such concepts is not sufficient to provide an answer to the initial question **IQ**. The concept of CSSM is introduced. Based on the requirements, collected in Section 1.2.2, a classification scheme is derived. It is argued that a combination of both domains, low-level sensors and high-level reasoning, could satisfy these requirements. Based on the introduced classification scheme, the related work is evaluated. Finally, some selected work is described in detail. It is shown that none of the approaches satisfies all requirements, but each provides relevant groundwork.

Current work distinguishes between AR and plan recognition (PR) [230]. Activity recognition is known as the task of inferring the user's current action from noisy and ambiguous sensor data. Plan recognition, in contrast, is referred to as inferring the action sequence leading to a goal under question by using (partial) action observations. An integrated approach, recognising the current activity from low-level sensors, the plan (including future actions) being executed and the goal the user tries to achieve, while satisfying the requirements at the same time, is missing. As Sukthankar et al. point out, *"bridging the gap between noisy, low-level data and high-level activity models is a core challenge of research in this area."*,([230]). A satisfaction of the initially stated requirements demands a combination of both domains of research. This chapter's aim is to, first, introduce the fundamental concepts of activity and plan recognition, then derive a classification scheme, and eventually, classify related work according to this scheme.

2.1. Activity and Plan Recognition

In the following, an overview of the research domains of AR and PR is provided. It is argued that, indeed, a gap between AR and plan/goal recognition exists that prevents the application of PR in the real world. This observation is also supported by Sukthankar et al. [230] and Kautz [114], who explicitly distinguish between low-level sensor data and high-level behaviour recognition. To satisfy the requirements, it is, however, necessary to reason about high-level behaviour on the basis of low-level sensors. Thus, both research domains, AR and PR, have to be combined. For this reason, in the following, an overview of both fields is given. Beside the description of the core problems, an overview of the most prominent work is given.

2.1.1. The Problem of Behaviour Recognition

This section introduces the basic concepts of human behaviour recognition from observation data. As defined by Baxter et al. [21], here, behaviour recognition refers to the overall process of activity, goal, and plan recognition. To this end, a general model is introduced which is then refined incrementally.

Objective of recognising the behaviour of a human protagonist is to label temporal segments by use of observation data. The labels thereby originate from a set of labels $\mathcal{A}$. $\mathcal{A}$ can be thought of as the alphabet of activities and a temporal sequence $p \in \mathcal{A}^*$ of labels $a \in \mathcal{A}$ as words over the alphabet $\mathcal{A}$. In general, no restrictions to the structure of p exist.

From the inference point of view, the aim of behaviour recognition is to conclude a hidden variable X from an observable variable Y. Figure 2.1 (a) provides a graphical illustration of this task. Here, the hidden variable X represents the activity a of the human protagonist. With respect to the probabilistic structure, two fundamentally different approaches to behaviour recognition exist: discriminative and generative classifiers [167]. In the following, both types are discussed.

Discriminative classifier A discriminative classifier determines the conditional probability $P(X \mid Y)$ – for an observed $y \sim Y$, it determines a probability distribution over X. To this end, discriminative classifiers model the dependence of the hidden variable on the observed variable. Discriminative classifiers are often preferred to generative classifiers, as they directly map the observation data to activity labels [167]. Typical instances of discriminative classifiers are linear regression models or support vector machines (SVMs).

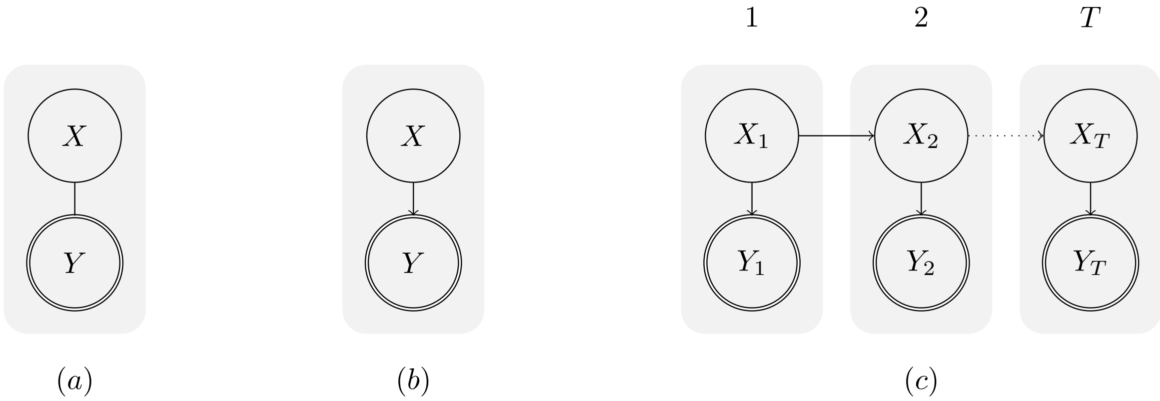

Figure 2.1.: Graphical representation of three different types of classifier. X represents a hidden state and Y an observation that is used to conclude information about X. (a) discriminative classifier, (b) generative classifier without temporal knowledge, (c) generative classifier with temporal knowledge.

Generative classifier In contrast to the discriminative classifier, generative classifiers provide the joint probability $P(X, Y)$. In other words, the generative classifier determines the probability of a joint occurrence of X and Y. The graphical model of the generative classifier is depicted in Figure 2.1 (b). By exploiting the causal link between X and Y, the joint probability can be factored into $P(X, Y) = P(Y \mid X)P(X)$. Here, $P(X)$ represents the prior probability of X and $P(Y \mid X)$ the conditional observation probability. This allows to include knowledge about the underlying process and to revert the dependency of the hidden variable on the observed variable. Providing the sensor model $P(Y \mid X)$ is typically easy, as it can be based on experiences [181, p.5]. The Naïve Bayes (NB) classifier is a typical instance of generative models.

Another advantage of generative models is that it is easy to incorporate knowledge about the temporal sequence of activities. This is done by including a transition model $P(X_t \mid X_{t-1})$ to describe temporal dependencies between different activity classes. Figure 2.1 (c) illustrates this graphically. Typical instances of temporal generative classifiers are Hidden Markov Models (HMMs) and Kalman Filters.

While temporal generative models allow to provide temporal smoothed estimates of the sequence of activities, they do not raise any restrictions to the possible sequences of activity labels p. In general, not all possible activity sequences actually happen. In fact, the sequence of activities that may actually happen has to be causally valid. In the following two different approaches to restrict the number of possible sequences of labels to the subset of causally valid activity sequences are discussed.

Explicit enumeration of activity sequences One approach to restrict the number of activity sequences is to explicitly enumerate all valid sequences. Such enumeration is known as plan library [115], where each plan represents a sequence of activities. From the viewpoint of generative models, a plan library can be represented as transition model, generated from supervised label training. An example of a plan library represented as transition model is depicted in Figure 6.2. The advantage of explicit enumeration is that due to the restriction to the set of label sequences plan library based approaches provide good recognition rates. However, the plan library has to be created manually, which is a tedious task due to the high number of action sequences [198]. Additionally, the number of valid activity sequences grows exponentially with the number of time-steps which additionally increases the effort to manually create a plan library.

Plan synthesis Another approach to restrict the number of possible action sequences is to employ a structured state representation for the hidden variable X and generate only causally valid action sequences. Here, the hidden variable represents not only the current activity, but also the current state of context, the activity is executed in. The restriction of the number of plans is implemented by restricting the activities to be executed in a particular state. The execution of activities depends on the current state. The valid action sequences are "generated" incrementally during the inference task. This allows to restrict the number of activity sequences without explicit enumeration to the subset of causally valid activity sequences. This technique is known as inverse planning [17], as it employs ideas from the domain of automated planning to infer the action sequence of a human protagonist. This technique is, for instance, used by Geib and Goldman [77] and Ramírez and Geffner [189].

After having introduced the general concepts of human behaviour recognition, the subsequent sections provide an overview of the specific realisations of these concepts. To this end, in the first step the concept of AR is introduced. AR utilises both discriminative and generative models, but usually do no raise any restrictions to the temporal sequence of labels. In the second step PR – a high level reasoning method that employs restrictions to the temporal structure of activity sequences – is discussed. Finally, the concepts of state and action are further discussed.

2.1.2. Activity Recognition

Activity The literature defines the term *"activity"* in a heterogeneous way. Rodríguez et al., for instance, summarise activity as that *"what users are doing"*,([194]). In contrast, van Kasteren [237] defines activity in terms of activities of daily living (ADLs) and Sukthankar [228] in terms of low-level motion data.

To abstract from certain settings and types of sensor data, for the scope of this thesis, the term activity is defined as in Definition 1.

Definition 1 (Activity) *An activity is the condition of performing an operation by a human protagonist. Performing an activity provides no further information about manipulated objects, intended purpose, or consciousness of the protagonist.*

This definition agrees with the implicit notion of activities in stateless AR by Bulling et al. [40]. Activities often comprise basic operations as sitting, standing, or walking [141, 18], but also the execution of gestures [7, 250] can be considered as activity.

Activity recognition Recognising and distinguishing different activities from sensor data is known as *"activity recognition"*. Analogous to the term activity, several different definitions exist. Here, we focus on the recognition of the activity executed by the human protagonist.

Definition 2 (Activity Recognition) *Activity Recognition is the task of inferring the user's current activity from noisy and/or contradictory sensor data.*

According to Bulling et al. [40], the objective of AR is *"to provide information on a user's behaviour that allows computing systems to proactively assist users with their tasks"*,([40]). Applications of AR arise in different domains, such as video surveillance [84], man-machine collaboration [250, 43] or training assistance in sports [72].

The activity recognition workflow Several approaches to recognise the activity of a (group of) person(s) exist. All of them share a common workflow, which Bulling et al. [40] called the AR chain. Objective of this workflow is to handle the inherent uncertainties of sensor data. This workflow has also been used for processing the sensor data within this thesis. Experiment X2 (see Section 6.2), for instance, applies this workflow on sensor data from wearable sensors.

According to Bulling et al. [40] the AR workflow consists of the following steps:

B1 Collection, preprocessing and annotation of raw sensor data,
B2 Sensor data segmentation and feature extraction,
B3 Classifier training based on training data subset, and
B4 Performance evaluation based on test data subset.

With respect to the first step (B1) – sensor data collection – it has to be noted that sensors differ in the amount of information they provide. Location-based sensors allow almost direct interpretation of the sensor data and thus allow location-based AR to achieve high recognition performances [97, 50]. Motion data, in contrast, for instance from accelerometers, does not allow such direct understanding, but requires sophisticated algorithms such as decision trees (DTs) [18], SVMs [191, 33], or random forests [45, 221]. In fact, Chen et al. state that *"wearable sensors are not suitable for monitoring activities that involve complex physical motions and/or multiple interactions with the environment"*,([46]).

AR utilises discriminative and generative models to estimate the activity of human protagonists from a wide variety of sensors. However, AR is not able to provide information beyond the estimated activity class. While knowledge about the user's current activity allows applications to be enriched, effective assistance can not be provided without knowledge about contextual information such as objects currently manipulated or the state of the environment [98].

Often, the process of AR is extended by including knowledge about the temporal structure. This is done by applying temporal generative models as for instance HMMs. The training of an HMM adds conditions to the sequences by introducing probabilities to the transition matrix. This, on the one hand, allows for temporal reasoning and typically increases the recognition performance, but, on the other hand, requires large amounts of training data to prevent overfitting. However, it is typically not possible to collect enough training data, so that each possible transition appears at least once. Furthermore, according to Chen et al. [46], who use the term data-driven approaches to represent AR approaches in the sense used here, typical AR approaches suffer from reusability and scalability problems.

Related work on activity recognition Activity recognition has successfully been integrated in various applications, ranging from the domain of sports [137, 136] to the domain of construction of automotives [222, 155].

An analysis of the related work on the topic of AR was done based on a review of the surveys on human AR by Bulling et al. [40], Lara and Labrador [139], Chen et al. [46], and Aggarwal and Ryoo [2]. For each work the applied sensors, the number of activity classes to be distinguished, and the utilised classifiers were analysed. Furthermore, the scenario that is used for experimental validation is collected for each work. Finally, it was reviewed whether the accuracy was used for evaluation. The accuracy was found to be the dominating performance metric used in the domain of AR and was therefore selected. Table A.2 gives an overview of the analysed relevant related work in the domain of AR.

The analysis of the relevant related work showed that 23 out of 37 works used inertial measurement units (IMUs). Overall, 13 different types of sensors where used for AR studies. The median number of classes to be distinguished is 8 with $IQR = 5 - 14$. Regarding the classifier, it was found that 17 different classifiers were used. The single most often used classifier is HMM, followed by DT, SVM, and NB. AR is employed in various application domains. The most frequently selected domains are ADL (19) and Location (8).

Evaluation of activity recognition Regarding the evaluation of AR, it can be seen that typical AR approaches distinguish between eight activity classes in median in the domains of ADL and Location. Since objective of this thesis is to reconstruct causal human behaviour sequences from sensor data, which essentially has to combine AR with high-level reasoning, the experimental validation of the proposed approach should at least cover these values. Additionally, as the most common classifiers for AR are HMMs, the use of HMMs as baseline classifier seems reasonable. Also, the choice of the accuracy as evaluation criterium for recognition performance is considered as meaningful.

Activity Recognition for low-level reasoning To conclude, the domain of AR provides different techniques for low-level reasoning. Several issues, however, prevent AR from being applied for high-level reasoning. Firstly, as there is no restriction to the label sequence $\mathcal{A}^*$, with increasing the size of $\mathcal{A}$, also the intricacy of the recognition process increases. Introducing causal relations between the labels of $\mathcal{A}$, would result in a sparse transition matrix of labels. This, however, requires high-level reasoning.

The process of AR allows an effective temporal labelling by means of sensor data. However, high-level reasoning, as required for assistance, needs to estimate the current activity of the user and also needs to provide knowledge about contextual information and possible future actions. As discussed, plain AR is not able to provide such knowledge.

Another point, raised by Chen et al. [46], is that AR approaches, as discussed here, are not reusable. Thus, for each application, even from the same application domain, large amounts of training data are necessary, in order to successfully recognise activities. This is also true for the other aspects of requirement R5, discussed in Section 1.2.2. The concept of transfer learning [177] provides approaches to target that issue. However, they are according to Pan and Yang [177], currently limited to small scale with limited variety.

2.1.3. Plan Recognition

The previous section showed that AR provides methods to reason about unrestricted activity sequences from noisy and ambiguous sensor data. As described in Section 2.1.1, different options exist to restrict the possible sequences of actions. This section discusses PR – a mechanism to restrict the number of possible action sequences by enumerating all valid plans.

Plan Knowledge about the current activity and additional context information can put applications in the position to provide context sensitive services. However, proactive assistance requires knowledge about the user's potential future actions and the final goal. The literature typically defines a plan as in Definition 3 in terms of sequences of actions leading to a goal [106, 75].

Definition 3 (Plan) *A plan is a sequence of actions starting from an initial state to a goal state. The user actively tries to achieve the goal.*

Consequently, the specification of a set of plans can be interpreted as restriction to the possible sequence of actions as described in Section 2.1.1. Furthermore, different plans can be considered as equivalent if they lead to the same goal. In other words, a goal can be regarded as a set of equivalent plans. Goal recognition can thus be considered as sub-task of PR.

Plan recognition PR targets at recognising the action sequence the user is going to execute in order to achieve a goal. PR is typically defined as in Definition 4. Here, the literature distinguishes between the agent that executes the plan and the observer that tries to estimate the agent's plan.

Definition 4 (Plan Recognition) *Plan recognition is the observer's task to estimate the plan executed by an agent under observation.*

Depending on the agent's knowledge about being observed or not, the literature distinguishes two different types of PR [53]:

Keyhole plan recognition denotes PR, where the observer has no knowledge about being observed or does not care about it [15]. *"The observer observes the agent by looking through a keyhole."* Different approaches to keyhole PR are proposed by Kautz and Allen [115], Lesh and Etzioni [143], Bouchard et al. [31] and Avrahami-Zilberbrand and Kaminka [15].

Intended plan recognition means that the observer tries to infer the *"intended"* plan of the agent. The agent is aware of being observed and might adapt his behaviour by this knowledge. Thus, in difference to keyhole PR, the intended plan is not necessarily the same as the actually executed plan. According to Carberry [44], this adjustment aids or hinders the recognition. Contributions to the field of intended PR are the works of Sidner [213] and Lisý et al. [151].

Adversarial PR, where the agent actively tries to prevent recognition by deception [116], often seen as third case [76], can be seen as special case of intended PR. However, here we focus on keyhole PR. It can be seen as the more general case as it makes no assumptions about possible cooperation of the agent being observed [31].

According to Armentano and Amandi [11], PR approaches can be distinguished not only by considering the agent's knowledge about being observed, but also by considering the output of the PR process. They divide them into:

Consistency based approaches check whether observations made about the behaviour of an agent match known plans. Plans not matching the observations are eliminated whereas matching plans are provided as inferred plan. Examples for such approaches are Kautz and Allen [115], Ramírez and Geffner [187], Levine and Williams [146] and the symbolic PR approach of Avrahami-Zilberbrand and Kaminka [15].

Probabilistic approaches, in contrast, provide a probability distribution of possible plans. Probabilistic approaches do not only respect prior probabilities about possible plans [11], but also enable further usage of decision theoretic approaches to select appropriate assistance. Examples are provided by Ramírez and Geffner [189], Geib and Goldman [78], Schwering et al. [206] and Raghavan et al. [186].

Here we focus on probabilistic PR as it allows to provide a basis for assistance.

Related work on plan recognition Apart from the particular PR setting, a large variety of different approaches to PR exists. Blaylock and Allen [25] for example use n-grams to recognise the goal schema and then apply Dempster-Shafer evidence theory to estimate the corresponding goal parameters. The use of uncertain observations require probabilistic methods to be applied. Kiefer and Stein [117] and Geib and Goldman [78], for instance, employ probabilistic grammars to handle such ambiguous information. Raghavan et al. [186] use Bayesian Logic Programming and Markov Logic Networks to account for such uncertainties.

In contrast to AR, most PR approaches assume the agent to be directly observed[1]. While this is typically not viable, current PR approaches restrict themselves to the use of datasets that are either artificially constructed or generated from man machine interaction. In fact, Blaylock

[1]The agent's action sequence is used directly during PR instead of being observed through sensors.

and Allen [26] argue that datasets originated from either simulation or computer interaction are most feasible for PR.

A review of relevant related work on PR was conducted by analysing the following surveys on plan and intention recognition: Carberry [44], Armentano and Amandi [11], Sadri [203], Han and Pereira [87], Sukthankar et al. [230]. Table A.1 gives an overview of PR approaches. The approaches are classified according to whether they are keyhole PR (*F.keyhole*) approaches or not, whether they provide the posterior plan/goal distribution (*F.probability*) or not, and if they use plan synthesis to generate possible plans instead of enumerating them (*F.plan.synth*). Additionally, the factor *F.direct* indicates in how far the observation can be considered as direct. *M.accuracy* and *M.convergence*[2] signal usage of the respective performance measure.

An analysis of the related work on PR showed that all approaches use direct observation with different characteristics. They are either manually specified as in [115], generated by simulation [189, 188], created by man machine interaction [143, 101], or by use of accurate sensors that are assumed to be perfect [146]. Few approaches exist that aim at recognising the agent's plan from noisy sensor observations [37, 149]. They strive at combining low-level sensor readings with high-level behaviour reasoning, as it is the scope of this work, and are therefore discussed in more detail in Section 2.2. Additionally, it can be observed from Table A.1 that only two approaches use the accuracy to evaluate the PR approach. Convergence, in contrast, is used by four approaches. In general, it has to be noted that many works on PR approaches omit the evaluation but rather focus on discussing technical aspects.

Plan recognition for high-level reasoning In summary, the domain of PR provides different approaches for high-level reasoning about the user's plan and the final goal from observation sequences. Typical approaches thereby focus on direct observation. In contrast to AR, which uses noisy and ambiguous observations that impede a direct conclusion of the executed activity, most PR approaches imitate uncertainties by use of incomplete but direct observation sequences. Consequently, PR approaches do not cope with contradictory observation, as sensors (e.g. IMUs) would provide.

Regarding reusability, PR approaches employ high-level representations of plans. This allows models to be reused within the same application domain, as typical for knowledge-driven approaches [46]. Thus, the application of PR based approaches would satisfy the requirements R1 and R5.

A simple combination of AR and PR, where the first generates an estimate about the current activity and the latter employs this estimate to conclude the plan and goal would result in a combination of the disadvantages of both approaches. Firstly, the restrictions raised by PR will have no influence on the estimated sequence of actions of the AR. Thus, AR will not benefit from the specification of possible action sequences. Secondly, as AR cannot produce perfect estimates, the PR has to cope with contradictory observation sequences. A combination requires both techniques to be interleaved in order to combine the advantages to provide high-level reasoning from low-level sensor data.

With respect to high-level reasoning, PR focusses on the recognition of the action sequence rather than on single actions. For this purpose, typical approaches to PR restrict the possible action sequences by creating a plan library by enumerating all valid action sequences. Creating such plan libraries is known to be a tedious task [199]. To overcome this issue, the next section describes the concept of plan synthesis. Plan synthesis employs states and actions to allow possible action sequences to be generated instead of manually specification by enumeration.

[2]The PR measure "convergence" indicates the performance the approach's capabilities to recognise the goal by providing the number of observations that were processed until the correct goal was recognised (see Section 5.4 for a more detailed information).

2.1.4. Actions and States

The previous sections introduced AR as mechanism for low-level reasoning and PR for high-level reasoning. It has been shown that AR allows to cope with noisy and ambiguous sensor data but does not raise any restrictions to the sequence of activities. PR, in contrast, employs sets of plans to restrict potential action sequences. The manual specification of such plan libraries requires an enumeration of all plans.

This section introduces the concept of plan synthesis by means of states and actions. For this purpose, first the concept of an labelled transition system (LTS) is introduced. Then, different modelling mechanisms to synthesise such LTSs, algebraic and model-based languages, are discussed. Finally, it is argued that the model-based description is better suited for the purpose of this thesis.

LTS for plan synthesis As discussed in Section 2.1.1, plan synthesis employs a structured state representation and actions to represent the set of causally valid action sequences. Human behaviour can be considered as a dynamic system, where the state of the environment changes over time through the execution of actions. A dynamic system can be formally captured by an LTS [96]. An LTS is a triple $\langle S, \mathcal{A}, \rightarrow\rangle$, where S is a not necessarily finite set of states and $\mathcal{A}$ is a set of actions. $\rightarrow \subseteq S \times \mathcal{A} \times S$ is a ternary relation that reflects the labelled transitions between states. If for two states $s, s' \in S$ an action $a \in \mathcal{A}$ exists, where $\langle s, a, s'\rangle \in \rightarrow$, the action a is said to be applicable in state s. The state s' is reachable from s by applying a. This is written as $s \stackrel{a}{\rightarrow} s'$. If for an action $a \in \mathcal{A}$ and all states $s \in S$ there exists at most one state $s' \in S$ such that $s \stackrel{a}{\rightarrow} s'$, the action a is said to be deterministic or has deterministic effects. $s \stackrel{a}{\rightarrow} s'$ can then be considered as function defined by $s' = a(s)$. If all actions are deterministic, the LTS itself is deterministic. For the scope of this thesis, we consider deterministic LTSs.

Bulling et al. [40] divides AR methods to stateless and stateful. The first type omits the state of the world but rather recognises the activity from the sensor signal, whereas the latter uses a model of the world to represent context information (see Section 2.1.2). The latter approach not only achieves higher recognition performance but also increases the amount of inferred information. Here, the term context refers to the state $s \in S$ of an LTS graph $\langle S, \mathcal{A}, \rightarrow\rangle$. Thus, properties of objects (e.g. the current location of a pot or the filling level of bottle) or the location of a persons can be interpreted as context. Estimating the location of a person [97, 248], thus, means estimating context information about the person.

Latent infinite LTS The set of states S and the relation $\rightarrow$ can be infinite, even if the set of actions $\mathcal{A}$ is restricted to be finite. For instance, a state feature that represents a counter variable and actions that increase this counter[3] will create a state space that represents the set of natural numbers. Also, the use of continuous state spaces (e.g. for location estimation) would result in an infinite number of states. However, often such methods use parametric methods to represent such continuous state spaces. A two dimensional gaussian distribution (e.g. used by Kalman filters) is typically represented as tuple $(\mu, \sigma) \in \mathcal{R}^2$. A discrete state space with n state features, in contrast, is represented as $(n-1)$-d-simplex. Albeit the discrete state space is finite, its computational complexity is higher than for the exemplified continuous state space.

From the inference perspective, handling such infinite LTSs is feasible as long as only a finite set of states is considered. This is possible, as an actual sequence of actions can be represented by a finite part of the LTS. The function, represented by the action a, *computes* the successor state from the original state, resulting in an incrementally expanded LTS graph. The resulting LTS is latently infinite, although generated by use of a finite (or compact) description.

[3] The set of natural numbers and the successor function, for example, represent such LTS, where even if the number of actions is restricted to one, the number of states is countable infinite.

Representing such functions requires actions to be represented as algorithms. For the scope of thesis we call such algorithmic language a computational action language, as it allows to represent LTSs in an algorithmic way. An algorithmic description is used to describe the relation $\rightarrow$.

Modelling mechanisms for LTS One important aspect of algorithmic descriptions is that they enable the reuse of models. According to Mernik et al. [157], domain specific languages (DSLs) are, among others, one key to enable reusability of software systems. This is particularly true for application generators, which use high-level abstractions to generate special purpose applications [126].

In the domain of software engineering and testing, formal specification languages are used to describe the behaviour of a system under development. Formal specification languages, such as Z [217], the Vienna Development Method [109], or the Common Algebraic Specification Language [161] are used. Such formal specifications are used to assess the correctness of implementation by tests based on the description of the system's behaviour. The literature distinguishes between two types of specification languages, algebraic and model-based languages [61, 96, 215]. Each of them uses a different approach to model the system's state and its operations.

Algebraic languages Algebraic languages describe the state of a system under test *"in terms of combinations of operations required to achieve that state"*,([61]). The system is defined by equivalence of (combinations of) operations to other operations. The state thereby does not reflect the internal aspects of the system itself. Popular examples are the specifications of abstract data types, where the state is defined by interactions of operations [85]. OBJ [83] and the Common Algebraic Specification Language [161] are examples of such languages.

The algebraic method provides a list of operations including interaction patterns such as sequences or orderings. The state space of the generated LTS usually consists of the sequences of operations. Partial order plans [125, 212] are typical examples of a dynamic system described in the algebraic way. The system's state is defined by the list of actions already completed. Due to the abstract definition of the system's state, reasoning about algebraically defined systems allows to infer the sequence of operations executed so far, but fine-grained information about the internal state of the dynamic system is not inferable. The usually small state space, created from a small set of operations, in contrast, limits inference intricacy. It can easily be seen that the system's state is defined as a single variable which abstracts all details away [61]. In fact, Sommerville [215, p.229f.] argues that algebraic methods are typically used to specify the interface of a system. Algebraic methods are used when the system's action is specified independently from the system's internal state.

Model-based languages In contrast to algebraic languages, model-based languages represent the system's state by the internal aspects of the system. Operations are specified by their interactions with the state. Pre- and postconditions are used to reflect the state-operation relation $\rightarrow$. The most prominent instances are Z [217] and the Vienna Development Method [109].

Model-based languages define the system in terms of state variables (each modelling one aspect of the system) and operations that interact with those. The system's state advances by applying operations on that state. Operations are defined in how far they modify the system's state; preconditions are used to restrict applicability to subsets of states, postconditions (also known as effects) model state evolvement after operation execution. The model-based specification describes the state of the dynamic system by providing a list of state variables including possible values, which usually leads to more complex state spaces than the algebraic method.

Operations are defined by their relation to that state. Additionally, in contrast to algebraic languages, model-based languages allow actions to be defined depending only on (parts of) the systems state. This enables the reuse of models on the level of actions. Furthermore, modelling the system's internal state, rather than sequences of actions, allows reasoning about the state of the world.

The language of the Stanford Research Institute Problem Solver (STRIPS) [73] can be considered as model-based. STRIPS is often used in the domain of automated planning and scheduling. The internal state of the system is represented by multiple state features. The detailed description of the internal state, on the one hand, allows to reason about single state features, but, on the other hand, easily leads large state spaces.

Model-based description for inference Denney [61] derived input value domains for system tests from both types of specification and highlights that the algebraic method is favoured when considering the interaction of operations while the model-based method is preferred when it comes to testing specific aspects of the system's state. Both descriptions allow to restrict the number of action sequences by modelling actions depending on the system's state. The representation of the system's state, however, differs.

Here, the model-based description is preferred to the algebraic description as it allows to model actions independently from each other and thus allows to reuse individual action specifications. Furthermore, it is easier to describe the system's state in terms of properties of the environment than in terms of actions performed so far.

2.1.5. Computational State Space Models

CSSMs describe a state space by computational means. For this purpose, a model-based description is employed to specify an LTS. The model-based description thereby uses state features to describe a state in the LTS. One particular occupancy of the state features describes one particular state of the LTS. Actions are defined, by preconditions and effects on these state features. Preconditions restrict the number of states an action can be applied to and thereby restrict the so-called branching factor. Effects describe how the state changes if a particular action is applied to it. The state space of the LTS is then created by incrementally applying all possible action to the set of initial states. In general, no restriction to the expressiveness of the language exists. It has to provide a computational description that describes the resulting state s' of applying an action a to a state s. To this end the language is called computational action language. Consequently, the resulting state space is not necessarily finite, albeit the number of actions and state features is finite. A definition of the concept CSSM is provided in Definition 5.

Definition 5 (Computational State Space Model) *A Computational State Space Model is a state space model, where the transition model of the underlying system is described by a computable function using an algorithmic representation. The resulting probabilistic model supports latent infinite state spaces.*

2.2. Related Work

In the previous section, a clear distinction of activity, context, and plan recognition was introduced and the current state of research in the particular domains was illustrated. Finally, the concept of Computational State Space Models was introduced. CSSMs use computational action languages to describe the dynamics of human behaviour by means of an LTS.

Based on the analysis, provided in the above mentioned surveys on activity and goal recognition [40, 139, 46, 2, 229], 25 studies were selected to assess the current state of research. For this purpose, a classification scheme is derived in this section. Additional studies were added to this list if they were found to represent essential information with respect to at least one factor of the classification scheme. A comparative overview of these studies is given in Table 2.2. From them, seven studies are described in more detail. These studies were considered as most relevant, as each of them represents an important groundwork with respect to question **IQ**. Bui et al. [37] (Study 21) introduced a manually specified state space model to jointly recognise the user's activity, plan, and additional context information from noisy location data. Liao et al. [149] (Study 22) extended this approach by replacing the need for manual specification with training. Both approaches were designed for the specific problem of inferring the user's state from location data and are neither reusable nor generalisable to different observation data. Training data is used to learn the user dynamics (including probabilities for state changes) and the observation model. Burghardt et al. [41] (Study 25) investigated different modelling formalisms to describe human behaviour to eventually generate HMMs. State transition probabilities are generated from these models, rather than by use of training data. Likewise, Baker et al. [17] (Study 1) provided a computational description of the user's dynamics. They replaced the need for training data with prior knowledge and goal distance based action selection. Ramírez and Geffner [189] (Study 4) extended this approach by considering observation noise as additional degree of uncertainty. They used Planning Domain Definition Language (PDDL) to describe a planning problem and illustrated how to represent PR based on noisy observations as inverse to the planning problem. Hiatt et al. [95] (Study 2) used a model-based description in terms of Adaptive Control of Thought – Rational (ACT-R) to describe the human decision making. In addition to goal-based utility functions, they introduced the usage of situation-based heuristics to determine a probability distribution of plans. These approaches used a computational description of the system dynamics and explicit mechanisms to reflect the process of human action selection to recognise the user's plan. However, they only considered simulated data and direct action observation. Sadilek and Kautz [202] (Study 13) were the first to combine a model-based description of the environment with low-level sensors. They showed how to simultaneously infer the activity, the plan, and additional context of multiple users from location data. However, the proposed approach had several drawbacks, such as the need to fully expand the state space. In the following, first the classification scheme is introduced and then the seven selected studies are described in more detail.

2.2.1. Classification Scheme

In order to classify the related work according to the requirements that were collected in Section 1.2.2, a number of factors, each representing individual parts of the requirements, were identified. The reason for the use of these factors is that most publications omit direct statements about the requirements. Additionally, other factors were added to conclude about the complexity of the targeted problem. Table 2.1 gives an overview of the factors used to classify the related work. In the following, first the surrogate factors for each requirement are introduced. Then, in the second step, the additional factors are introduced and described.

Requirements Requirement R1, states that an inference system has to provide information about the user's potential future actions and the final goal. The factor *F.target* reflects this requirement by providing information about the estimation target of the respective approach. Possible values are (A)ctivity, (C)ontext, (G)oal, and (P)lan. The requirement for online inference (R2) is represented by *F.complexity*. *F.complexity* provides the computational complexity of the approach for one time-step. A value of 1 signals that the complexity is independent

Factor	Description
F.latent.infty	Method allows inference in latently infinite state spaces (typically employing a computational action language).
F.plan.synth	Plan synthesis is supported. Otherwise, the approach requires to create plan libraries by explicitly enumeration.
F.duration	Durative actions are supported. (This will significantly increase inference complexity, as the starting time for an action becomes another state variable, which has a large value space.)
F.action.sel	Explicit mechanisms for modelling human action selection based on opportunistic and/or goal driven features are supported.
F.probability	Method provides (an approximation of) the posterior probability distribution over states (or actions, depending on the mechanism). This is a prerequisite for selecting assistive interventions using decision-theoretic methods (i.e. that aim at maximising the expected utility).
F.struct.state	The state maintained by inference provides a structured representation of the environment state. This allows the formulation of state predicates and the dynamic synthesis of contingency plans. (Otherwise the state typically represents the action currently executed.)
F.non.monoton	Non-monotonous action sequences are considered, that – temporarily – may increase goal distance. (This affects the number of plans that need to be considered. Methods using explicit plan enumeration usually avoid non-monotonicity.)
F.complexity	Filter step complexity (computational complexity for the filtering step from t to t+1). If greater than $\mathcal{O}(1)$, for instance $\mathcal{O}(t)$, then online filtering is essentially intractable.
F.model.based	Method is based on a model-based description.
F.CSSM	The approach is considered as Computational State Space Model.
F.source	The source of state transition in the model. L – machine learning is used, or P – prior knowledge is used to estimate the state transitions.
F.target	The method targets at estimating the (A)ctivity, the (C)ontext, the (G)oal and/or the (P)lan.
Method	Type of inference method used.
Scenario	Scenarios considered in experimental tasks.
N.states	Number of S states considered. (See text for further explanation.)
N.plan.length	Lengths of plans considered in study.
N.classes	Number of classes in classification target used for performance evaluation.
N.subjects	Number of subjects participating in trials (or sim in case evaluation is based on simulated observations).
M.accuracy	Accuracy is provided as performance measure.
M.conf.based	Other quantities based on confusion matrices (true–positive rate, precision, etc.) are provided as performance measures.

Table 2.1.: Factors for analysing related work.

from the number of observation items processed so far. Higher values indicate online inference to be impossible. The factor *F.Method* serves as surrogate measure for the requirement R3, as it provides information about whether the approach allows to process uncertain sensor data. Additionally, the factor *F.probability* indicates an estimation of the posterior probability, which signals the approach's capabilities of handling probabilities. Finally, the factor *N.subject*, which provides the number of subjects used for evaluation, allows to conclude the usage of real instead of simulated observation data. A value of sim, signals that the evaluation was based on simulation. Requirement R4 is directly reflected by the factor *F.latent.infty*. The factors *F.plan.synthesis*, *F.model.based*, and *F.prior* reflect different aspects of the requirement R5, as they signal the usage of a knowledge driven approach. As discussed, the usage of knowledge driven approaches allow to create reusable models. The following three classes of criteria can be distinguished: CSSM properties, criteria regarding the complexity of the evaluation setting and evaluation factors. Each of them is described in the following.

CSSM properties The *F factors* represent properties provided by CSSMs. The use of these factors allow a more detailed assessment of the capabilities of the approaches in favour to a binary classification for each requirement. They illustrate how many features of CSSMs are provided by the approach. Thus, they show in how far an approach can be considered as CSSM. The factor *F.CSSM* summarises this assessment and indicates whether the proposed method is considered as CSSM, namely if Definition 5 holds. The *F factors* are directly related to the requirements derived in Section 1.2.2.

Complexity of the evaluation setting The *N factors* provide statements to quantify the complexity used to evaluate the approach. Knowledge about the complexity of the experimental setting gives an indication about the general capabilities of the proposed approach. The following dimensions of complexity were identified:

CD.1 The factor *N.states* provides the number of states used to represent the experimental setting. Typically, with increasing the number of states, the inference complexity increases.

CD.2 *N.plan.length* lists the average length of the action sequence. Again, this has influence on the inference complexity, as the number of possible plans increases exponentially with increasing of the length of the action sequence. Imagine, the repeated decision between two actions. The number of possible plans would be given by 2^n, where n is the plan length.

CD.3 *N.classes* assesses the number of target classes to discriminate between. As can be seen from Table A.2, the median number of target classes in the field of AR is 8 ($IQR = 5-14$). Inference complexity typically increases with increasing the number of target classes.

Finally, the factor *N.subjects* gives information about the generalisability of the approach, as it describes the number of human subjects the sensor data was obtained from.

It has to be noted that the factor *N.states* has several drawbacks. Firstly, the number is often not explicitly stated in the literature but rather has to be inferred from the textual description. Secondly, depending on the discriminative capabilities of the observation sequence, only small parts of the state space are considered during inference. This is often the case in studies based on simulated data, where the observation sequence is constructed from the sequence of ground actions. Ramírez and Geffner [188] and Baker et al. [17], for example, use this kind of observation in their studies. Finally, the expressiveness of the factor *N.states* depends on the representation of the state space. With respect to the above mentioned issues, the factor *N.states* is considered as surrogate for state space complexity.

Evaluation factors Several methods exist for estimating the performance of discrimination tasks (see Section 5.4). In the field of AR, the preferred performance criteria are the accuracy and other confusion matrix based methods. This can also be observed from Table A.2. The *M factors* illustrate in how far confusion matrix based evaluation methods are used. As the accuracy was found to be the dominant metric, the factor *M.Accuracy* is introduced to indicate the usage of the accuracy for evaluation. As remaining criteria, the *Scenario* and the inference *Method* are used. They allow to provide an overview about the intended purposes of the approaches.

The factors introduced above allow to classify the related work. The *F factors* give an overview of the approach's general capabilities, whereas the *N factors* provide information about the complexity of the evaluation scenario. A classification based on these factors allows to evaluate the related work with respect to the requirements (see Section 1.2.2) and the experimental evaluation. In the following section, this classification is presented.

	Reference	F.latent.infty	F.plan.synth	F.duration	F.action.sel	F.probability	F.struct.state	F.non.monoton	F.complexity	F.model.based	F.CSSM	F.prior	F.target	Method	Scenario	N.states	N.plan.length	N.classes	N.subjects	M.accuracy	M.conf.based
1	[17][188]	■	■	□	■	■	■	■	1	■	■	■	GP	B^{D}	L	70000†	20	3	sim	□	□
2	[95][243]	■	■	□	■	■	■	■†	1	■	■	■	GP	B^{D}	OM	-	-	◇	sim	□	□
3	[188][87]	■	■	□	■	■	■	■	t	■	■	■	GP	B^{P}	K	10000†	-	3	sim	□	■
4	[189][87]	■	■	□	■	■	■	□	1	■	■	■	GP	B^{PL}	K	70000	8	5	sim	■	■
5	[98][46]	■	□	□	□	■	■	□	1	◇	□	■	AC	B^{D}	A	200000	5†	6	6	□	■
6	[99][46]	■	□	□	■	■	■	□	1	■	□	■	ACP	B^{D}	K	70000	40	◇	3	□	□
7	[56][2]	□	■	□	□	■	■	■	1	◇	□	■	AC	B^{D}	O	250000†	-	5	5	■	□
8	[57][197]	□	■	□	□	■	■	■	t	■	□	■	A	N^{BN}	M	1000†	-	15†	sim	□	■
9	[69][229]	□	■	■	□	■	□	■	1	◇	□	□	A	B^{H}	K	28	6	6	-	■	■
10	[111]-	□	■	□	□	■	□	■	1	◇	□	□	A	B^{H}	A	300†	12†	15	3	■	□
11	[169][46]	□	■	■	□	■	□	■	1	◇	□	■	A	B^{RP}	K	96	-	13	2	□	■
12	[168][46]	□	■	■	□	■	□	■	1	■	□	■	ACGP	B^{RP}	O	3500†	3	3	2†	□	□
13	[202][229]	□	■	■	□	■	■	■	t	■	□	■	ACGP	O^{ML}	M	-	20†	4	14	■	■
14	[251][46]	□	■	■	□	■	■	■	1	◇	□	□	AC	B^{D}	AK	528†	-	33	3†	■	■
15	[38][46]	□	□	■	□	■	□	□	1	□	□	■	A	N^{MH}	O	720†	-	2	1	□	■
16	[31][46]	□	□	□	□	□	□	□	1	□	□	■	GP	L^{DL}	K	-	15	6	sim	□	■
17	[47][46]	□	□	□	□	□	□	□	1	■	□	■	A	L^{DL}	AK	-	24†	8	3	■	□
18	[117]-	□	□	□	□	□	□	□	t^2	□	□	■	AGP	O^{G}	M	-	50†	◇	2†	□	□
19	[199][46]	□	□	□	□	■	□	□	1	□	□	■	ACGP	L^{P}	A	100†	40†	7†	6	■	□
20	[212][2]	□	□	■	□	■	□	□	1	□	□	■	A	B^{MF}	A	20000	14†	14	3	■	■
21	[37][46]	□	■	■	□	■	■	■	1	■	□	■	ACGP	B^{RP}	O	74†	-	2	2	□	□
22	[149][46]	□	□	■	□	■	■	■	1	□	□	□	ACGP	B^{RP}	M	-	-	6†	1	■	□
23	[174][229]	■	■	□	■	■	□	■	1	■	■	■	GP	B^{D}	M	49†	-	◇	sim	□	□
24	[28][98]	□	□	□	□	■	■	■	1	□	□	■	AC	B^{H}	A	50,181,120	11	11	1	□	□
25	[41]-	□	■	■	■	■	□	□	1	■	□	■	A	B^{H}	M	9	10	-	-	□	□

Table 2.2.: Overview of related work and evaluation studies. The superscript reference in the first column gives the original source. "■"=Feature included, "□"=Feature not included, "$x^{\dagger}$"=property x not explicitly stated, "-"=value unknown, "◇"=property not meaningful considering target, F.target: "A"=Activity, "C"=Context, "G"=Goal, "P"=Plan, Method: B:variant of sequential Bayesian fitering (exact: "H"=HMM or extension, "D"=other DBN, "Pl"=transformation into a planning problem, "P"=POMDP; approximate: "Pf"=Particle Filter, "RP"=Rao-Blackwellized Particle Filter, "Mf"=Marginal Filter), N:Non-sequential Bayesian inference ("MH"=Metropolis Hastings, "BN"=unrolled Bayes Net), O:other exact method("G"=grammar-based, "ML"=Markov Logic Network) Scenario: "K"=kitchen task, "A"=other ADL, "O"=office, "M"=miscellaneous other scenario

2.2.2. Classification Results

Table 2.2 gives an overview of related work, classified according to the classification schema introduced above.

Recognition target Regarding the recognition, only five studies (Studies 21, 12, 22, 19, and 13) were concerned with the integrated estimation of all estimation targets. The majority of the studies (19 studies) considered the recognition of the user's current activity, at least. From these, 10 studies also considered the recognition of context information in addition to the activity. The recognition of the user's plan, including his goal, was considered by 12 studies. Here, the dominant approach is the application of plan synthesis instead of explicit enumeration of plans. However, only three (Studies 21, 12, and 13) of the approaches that targeted activity, context and plan recognition, synthesised the possible plans. Neither Sadilek and Kautz [202] (Study 13), Bui et al. [37] (Study 21), nor Nguyen et al. [168] (Study 12) supported latent infinite state spaces or appropriate action selection mechanisms but rather relied on manual specification or training of state transition probabilities.

Reusability Regarding the reusability, from the 25 studies, only 19 studies used a DSL that allowed further reuse of the description. From these, the majority of 12 studies used a model-based description that would potentially allow a reusable description at the level of actions by simultaneously representing the environment as state of the system. Only five of them can be considered as CSSM (Studies 1, 3, 5, 2, and 4). While these studies illustrated that it is indeed possible to create reusable models for plan recognition by using computational action languages, they neither considered the recognition of the action sequence of real persons, nor did they use noisy sensor data for inference. In addition, albeit modelling information about the current state of the world, none of them considered the recognition of context information. There was no study that supported all features.

Complexity of the evaluation setting Concerning the considered complexity, an analysis of the state space sizes (CD.1) shows that only problems with small size were used. CD.1 gives quantitative information about the level of detail considered in the study. The median number of states considered in the studies is 1,000 (with interquartile range $IQR = 98 - 70{,}000$). From the 25 studies that were considered to assess the current state of research, only three used more than 100,000 states (Studies 24, 7, and 5), whereas 10 studies considered not more than 1,000 states.

The length of the action sequence (CD.2) used within the experimental evaluation of the studies also gives information about the complexity of the task under observation. Shorter sequences are usually easier to be handled, whereas longer sequences introduce additional complexity due to the growth of the number of action sequences that is exponential in the length. The median plan length considered in the studies is 14 ($IQR = 9 - 22$). Beside the state space size and the plan length, the use of action durations is another factor that increases inference complexity. Studies that considered action durations use a maximum number of 20,000 states.

It can be observed that CSSM-like approaches were evaluated through simpler scenarios with maximum state space size of 70,000 and maximum plan length of 20. Approaches that were not considered as CSSM, in contrast, had a maximum number of 50,181,120 states and maximum plan length of 50. With respect to the number of target classes to be distinguished (CD.3), CSSM-like approaches used a medium number of 3 ($IQR = 3 - 4$) classes, whereas non-CSSM approaches used 6 ($IQR = 5 - 13$) classes. Typically, inference becomes more challenging with increasing number of target classes as the probability of randomly selecting the correct class decreases with increasing number of classes.

Evaluation An analysis of the performance measures revealed that 10 of the studies used the accuracy to assess the recognition quality. Additionally, 11 studies used other performance measures based on the analysis of confusion matrix. No study used measures sensitive for causal structure of the estimated sequence. Appropriate performance measures that reflect agreement in the causal structure are discussed in Section 5.4.

Inference methods When considering the inference method, approximate methods are dominant. The use of approximate methods is necessary for large state spaces. Furthermore, the use of action durations or a high number of plan steps increases complexity, which renders exact methods to be infeasible and requires approximate methods to be applied. All studies, except for one that focussed on approximate online inference (complexity = $\mathcal{O}(1)$) used variants of the PF. Only the study of Shi et al. [212] (Study 20) used the D-Condensation filter, an approximate method tailored for categorical state spaces (see Section 3.2 for a detailed discussion).

Location-based evaluation One interesting result of the analysis of the selected studies is that all studies concerned with the integrated recognition of activity, context and plans used sensor data that is easy to interpret. The studies of Bui et al. [37], Nguyen et al. [168], Sadilek and Kautz [202] (Studies 21, 12, and 13) used location data. The study of Roy et al. [199] assumed the sensors to be accurate. Easy to interpret sensors, such as pressure, passive infrared (PIR), and reed switch sensors, are used. Accelerometers were used only to detect whether the hand moves or not. However, the use of location data seems to be a good choice to illustrate the baseline capabilities of an integrated approach to activity, context, and plan recognition based on noisy sensor data.

Summary To conclude, several approaches exist that strive to reconstruct human behaviour from low-level sensors. Only few approaches exist that can be classified as CSSM and thus allow for handling infinite state spaces. However, there is no approach that fulfils all of the introduced requirements (see Section 1.2.2). In the following a detailed description of the seven studies is presented that contribute most to the research topic of this thesis.

2.2.3. Policy Recognition in the Abstract Hidden Markov Model by Bui et al. [37]

Bui et al. [37] propose a hierarchical model for recognising the user's currently executed plan from sensor data. Their approach was the first to integrate low-level sensor data and high-level behaviour recognition in terms of the user's plan.

Model The hierarchical model consists of policies, where each level in the hierarchy refines the above level by introducing a sequence of policies. A policy is similar to a contingency plan in that it does not describe one possible action sequence, but determines how a lower level policy is selected in a given state. Formally, a policy is defined as quadruple $\pi = \langle S, D, \beta, \sigma \rangle$, where S represents the set of applicable states, D the set of destination states, β the set of stopping probabilities, and σ a selection function $\sigma(s, \pi)$, specifying the probability that π is selected in state s. At the bottom level, policies can be understood as primitive actions that stop immediately after one time-step. Below this bottom policy level, a state level is attached, where each state emits observations. Each policy can be considered as sub-goal, the top level policy as top level goal. Thus, recognising the top level policy is similar to recognising the user's goal. If a lower level policy terminates, control is given back to the above level policy, which determines the stopping condition itself.

Inference method To estimate the policy sequence from sensor observations, a dynamic bayesian network (DBN) is constructed from the model. In the first step Bui et al. [37] illustrated how inference is done by use of direct state observation. By exploiting context specific independencies [32], they show that inference complexity is independent from the number of levels in the hierarchy of policies. Bui et al. [37] apply a Rao-Blackwellized Particle Filter (RBPF) to make inference more efficient compared to a PF.

Evaluation To show their approach to be working, Bui et al. [37] conducted an experiment with the objective to track a human protagonist while walking through an indoor environment. At the same time, the proposed approach allows to predict the building's exit that the user is most likely heading to. The hierarchical model is created by manually applying region-based decomposition to the state space, which basically creates a hierarchy of locations based on the hierarchy of regions of the environment. Altogether, three levels are created, where the highest level consists of two policies, one for each exit of the building. The bottom level contains nineteen policies, resulting in an overall state space of 74 states. The location of two persons is tracked based on cameras in two experiments for about 400 seconds.

Summary Bui et al. [37] were the first to combine noisy low-level sensor data with high-level PR. They illustrate their approach working by tracking and predicting a user's path through an office environment based on camera-based location data. By applying the RBPF, the authors showed their approach to be more effective than standard PFs. The policies and the policy selection function, which describes the transition probability of actions and upper level policies, were specified manually. This requires the state space to be finite, as it would otherwise impossible to manually define policy selection probabilities for each state. The authors showed that the hierarchical model leads to a natural decomposition of the environment. Several issues, however, prevent their approach from being applied to the integrated recognition of activity, context, and plans of the user. The proposed model allows only states to emit observations. As can be seen from Table A.2, the dominant sensors for AR are IMUs. This type of sensor is inherently unsuited to provide state observation, but in contrast allows to observe actions.

Another issue is that actions have to stop after one time-step. A re-selection of the same action with probability p_{self} represents a geometric action duration with success probability of p_{self} similar to standard HMMs. This is a limitation, as not all action durations can be modelled by the geometric distribution. Additionally, the manual specification of policies requires high effort to specify the model and prevents models from being reused. Finally, the sampled variables s (state) and t (termination level) have a categorical domain. As discussed in Section 3.2.2, PF-based approximation is inefficient for categorical state spaces due to the representation of densities based on particle counts, especially if the transition matrix is sparse.

In summary, the method presented by Bui et al. [37] provides an interesting basis, as it is the first to combine high-level reasoning with uncertainties inherent to sensors. The need for manual specification, however, contradicts to the requirements of reusability and latent infinity.

2.2.4. Learning and Inferring Transportation Routines by Liao et al. [149]

Liao et al. [149] present an integrated approach to PR from noisy sensor data to *"bridge the gap between the raw GPS sensor measurements and high-level information such as a user's destination and mode of transportation"*,([149]). In their article they extended previous work.

In [178], the authors apply a PF to infer the transportation mode and the most likely route in an urban environment from GPS data. The model was later extended [147] by a goal and a goal switching node, which accounts for goal selection and goal-directed movement. In [179], the authors prove their work to be feasible and present the system opportunity knocks. Opportunity knocks provides assistance in that it guides users through the urban environment by use of different transportation services. In Liao et al. [149], the previous model is extended by a novelty node, to model and detect novel, yet unknown, behaviour.

Model The model introduced by Liao et al. [149] is based on a directed graph representing the street map. Edges represent streets and vertices correspond to intersections of edges. Locations of the user are restricted to be on edges and are represented as distance to the starting vertex. The model consists of three layers, each of them modelling one aspect. The bottom level is composed of the current state x, namely the user's location and the velocity, the current transportation mode m, and the edge transition variable τ, which provides the user's decision about the future direction. Additionally, the nodes z, which represent GPS readings, and θ, which provides a snapping to an edge, are used as observable variables. The middle layer represents the user's current goal g and the current trip segment t and boolean variables f^g and f^t that signal changes about goals or trip segments. A trip segment represents a transfer from a start location to an end location by use of a transportation mode. For each transportation mode a counter variable is used to determine the waiting time to change. Aim of the top level of the model is to decide whether the user's current behaviour is novel. This is signalled by the boolean variable n. Novel behaviour causes the model to switch to an untrained mode, where goal g and trip segment t are set to be unknown.

Inference method Similar to Bui et al. [37], the authors applied the RBPF algorithm. Here, all variables, except for the user's location, were used as Rao-Blackwellising variables and were sampled accordingly. A Kalman filter was used to exactly track the location-based on the sampled velocity, the direction, and the edge association. The authors first learned the value domains of the variables in an unsupervised manner. To do this, the expectation maximisation algorithm was used.

Evaluation To evaluate the proposed approach, the GPS data of one person was recorded for 60 days. The data of 30 days was used to train the model. The goal threshold was set to one hour, which means that location is considered as goal when the location does not change for more than one hour. Locations, where the person might change the transportation mode (e.g. bus stops, parking lots), were extracted by analysing the transportation mode transition probabilities. Six of the goal locations and all transportation mode changing locations were learned. Finally, transition matrices for all other variables were learned, and the directed graph is build upon the most likely trajectories. The evaluation of the goal recognition showed that their model was able to recognise the correct goal, out of six possible goals, with an accuracy of 82% after 50% of the time. After 75% of the time the accuracy increased to 98%. Finally, the authors illustrated, how the proposed system can be embedded into an application named opportunity knocks, which assists users in finding the correct way to a predefined goal location in urban environments.

Summary In their work, the authors successfully demonstrate, how to simultaneously recognise the goal, the route, and the transportation mode from raw GPS data. Similar to Bui et al. [37], they illustrate how to combine low-level sensor readings with high-level goal recognition. Furthermore, they replace the need for manual specification of states, transitions, and probabilities, proposed by Bui et al. [37], by an automatic learning process. By applying the expectation maximisation algorithm, the authors show that the system is able to adapt itself to the user's needs. Otherwise, the system needs a huge amount of training data (30 days in the experiment), for reasonable adaption. This amount could be further reduced by use of prior knowledge about bus stops and other common (sub-)goals. From the inference perspective, like in Bui et al. [37], the model use PF based inference for categorical domain variables.

The proposed model is tailored for location-based reasoning and is not easily adaptable to other scenarios e.g. using wearable sensors. Moreover, it only allows states to be observed, which is a limitation when it comes to the application of IMUs. Finally, the proposed model does not allow to explicitly specify durations. The application of the Kalman filter as sub-model for moving on "edges" replaces the need for durations, but the waiting time for transportation changes is modelled by a counter. While this seems sufficient for the transportation mode tracking, an application in tracking ADLs requires more sophisticated duration modelling as the analysis in Section 6.2 suggests.

2.2.5. Synthesising Generative Probabilistic Models for High-Level Activity Recognition by Burghardt et al. [42]

Burghardt et al. [42] propose the synthesis of HMMs from different symbolic descriptions of the user's activity. They investigate different kinds of description formalisms such as task models and STRIPS. They demonstrate that it is possible to create the state space for probabilistic models from such descriptions.

Model In their work, the authors show how to use the different description formalisms to generate the transition matrix of HMMs. From the task model, the state space is generated by incrementally executing all potential actions, starting with the initial action. Thus, the state space is formed by user actions, where an individual state represents the actions that have to be executed in order to reach this state. The transition probabilities are calculated by considering the priorities of actions that are allowed to be executed after the currently executed action. The custom Collaborative Task Modelling Language allows the specification of observation models. Thus, the observation model can directly be carried from this model.

With respect to STRIPS, the synthesis of the HMM is done by incrementally applying the planning operators to the initial state until the state space is fully expanded. Preconditions are used to limit the application of actions to a subset of states. Effects allow actions to change the state. The state space is represented as finite LTS. A state is represented as a pair of an environment state and an action. Transition probabilities are uniformly distributed over all reachable states. To cope with sensor data, the authors propose the use of action observation models.

Inference method Burghardt et al. [42] generate the transition matrix of HMMs from the different modelling formalisms. Exact Bayesian inference is applied to estimate the most likely state sequence from noisy and ambiguous sensor data. For this reason the state space has to be fully expanded in order to represent the transition matrix of the HMM. As a consequence, their approach is not able to handle latent infinite state spaces.

Evaluation While Burghardt et al. [42] demonstrate their approach with an example in the domain of team meetings they do not provide any evaluation with respect to recognition accuracy. They rather focus on showing their approach to be valid by generating HMMs from the different descriptions. For this reason, they focus on team meeting where the team's action sequence has to be estimated from location data. In another work, Burghardt et al. [41] show that their approach allows to use a partial order planner to generate possible team meeting action sequence from a STRIPS model. However, they do not apply their approach to plan recognition from sensor data.

Summary Burghardt et al. [42] propose the generation of probabilistic models from different kinds of formal descriptions of human behaviour. Furthermore, the authors state that the use of such formal descriptions allows for reusability within the same application domain (R5.1). They exemplify the generation of HMMs from top-down task models and bottom-up STRIPS models.

The work of Burghardt et al. [42] is an important basis for this thesis, as it allows the generation from probabilistic models of reusable descriptions of human behaviour. However, the work has several limitations which hinder the direct application. Firstly, the usage of HMMs for exact inference requires the state space to be expanded prior to the inference. This limits the state space size and prevents the state space from being infinite. Moreover, state spaces have to be of very limited size in order to apply exact inference. To overcome the issue of large state spaces they sketched the idea to use approximate inference but never evaluated it.

The second issue is that the transition probabilities are uniformly distributed which means that all possible actions have equal probability. Burghardt et al. [42] describe the idea to generate the transition probabilities from heuristic functions like that from the planning domain or from knowledge-driven systems. In particular they discuss the application of production selection rules from ACT-R [8] but provided no evaluation.

The third drawback of their approach is the missing support for durative actions. While the approach would in general allow the use of the geometrical distribution to model action durations, the authors omit the discussion of action durations.

2.2.6. Action Understanding as Inverse Planning by Baker et al. [17]

Baker et al. [17] treat the problem of estimating the user's goal from observations of his behaviour through plan synthesis. They successfully demonstrate that, given a model-based description of the environment and possible actions to execute in that environment, it is possible to reason about the plan currently executed. Objective of their work is to compare different models of rational agents, comprising static goals, changing goals, and sub-goals, with human reasoning and goal prediction. To do this, they build a model of the environment, possible actions, and a (set of) goal states(s).

Model Baker et al. [17] use a model-based specification of the model to create a Markov Decision Process (MDP). The model consists of the state s, which essentially represents the state of the world, the currently executed action a, and a goal g, which the agent is trying to achieve. Contrarily to Bui et al. [37], who manually specify the transition probabilities and Liao et al. [149] who use machine learning methods to estimate the transition matrices, the authors make use of their model-based description and use an action selection function to determine action selection probabilities. Given the goal g, in state s_t and world w, the action a_t is selected with probability $P_\pi(a_t \mid s_t, g, w) \propto \exp(\beta Q^\pi_{g,w}(s_t, a_t))$, where Q represents the expected cost of taking action a_t in state s_t. The costs are thereby given as the minimal sum of costs of the action sequence leading to the goal state. Costs are assumed to be proportional to the negative length of the resulting action sequence, which basically leads to goal-directed action selection. The β parameter was used to control the level of goal-directedness of the action selection. High values of β lead to more goal-directed action sequences, whereas low values enforce the other direction, a value of $\beta = 0$ results in uniform action selection. Two additional parameters γ and κ control goal switching and the selection of sub-goals.

Inference method Baker et al. [17] apply the framework of MDPs. MDPs allow the full observation of the state sequence execution by the user. Thus, the proposed model only has to deal with one source of uncertainty, namely the selected action of the agent.

Evaluation Since the aim of their work is to compare human goal prediction with the computational prediction of a rational agent, three experiments with varying goal conditions were conducted. A maze world, where at each location, except for the border, nine actions are applicable, one for staying, and one for each adjacent cell, is used. Three different goals are distributed to the corners of the maze. Additionally, obstacles that represent walls to hinder direct goal approaching are added. Some obstacles contain a gap to allow a direct passage. In their experiments, the authors propose three models, each of them extending the action selection function of the former. In the first model, the goal is assumed to be static ($\gamma = 0, \kappa = 0$). By setting $\gamma \neq 0$, the second model allows goals to be changed. The third model allows for selecting sub-goals ($\kappa \neq 0$). Their analysis gives evidence that it is possible to map human action understanding to computational models of rational agents.

Summary Baker et al. [17] were the first to present a plan synthesis-based approach to recognise the goal of an observed actor. The model-based description allows them to completely omit the necessity of training data. Instead they propose the use of probabilistic action selection based on the goal distance. It can therefore be clearly considered as a CSSM. The use of a model-based description allows Baker et al. [17] to specify potentially unlimited state spaces. However, the authors note that other inference algorithms have to be applied in order to handle such state spaces. In their experiments the authors show that their models are able to explain human reasoning about goal-directed behaviour. While the work of Baker et al. [17] is seminal in the model-based specification of human behaviour models for automatic action understanding, it still suffers from inability to use partial action observations. Additionally, the lack of probabilistic action durations prevents further usage for recognition of activities, contexts and plans.

2.2.7. Goal Recognition over Partially Observable Markov Decision Process: Inferring the Intention of a POMDP Agent by Ramírez and Geffner [189]

In their work, Ramírez and Geffner [189] propose a model-based approach to PR from partial observations. They extend their previous work, where they introduced a planning-based approach to PR by identifying the goal an observed optimal plan would lead to [187]. There, similar to Baker et al. [17], they replace the typically used plan library by a model-based description of the planning problem. In a later work, Ramírez and Geffner [188] extend their approach to probabilistic PR, which provides a probability distribution over goals in favour to binary decision about goals.

Model Ramírez and Geffner [189] defined a planning problem by means of PDDL. A set of state features is used to represent properties of the environment. A set of actions, specified in terms of preconditions and effects related to these state features, is used to describe the system dynamics. The state space is created by the transitive closure of the set of actions on an initial state. A set of states is selected as goal states. In addition, a cost function is used to assign a non negative value to each pair $\langle a, s \rangle$ of action a and state s. For each pair $\langle a, g \rangle$, where g is a goal state, the costs are defined to be zero. The problem of planning is now defined as finding the optimal path (in terms of minimal costs) from the initial state to a goal state.

Similar to Baker et al. [17], they define PR as inversion of the planning problem. Given (partial) observations about an optimal plan, the PR problem is to infer the goal of the observed plan. Their key assumption here is that a rational agent will choose a cost optimal plan in order to reach the goal. Like Baker et al. [17], the authors use the Boltzmann policy as action selection function. They considere three different sources of uncertainty: (1.) the agent's plan is only partially observable, which means that some observations are dropped from the sequence of action observations, (2.) the agent's selection of action is not deterministic, but determined by an action selection probability density function, and (3.) actions are assumed to be non-deterministic, meaning that the outcome of an action is not fixed but rather probabilistic. Note that albeit Ramírez and Geffner [189] allow observation sequences to be incomplete, they do not consider noisy or ambiguous observations. Since this is the core meaning of the requirement R3, their work is not considered satisfying this requirement.

Inference method To solve the PR problem, Ramírez and Geffner [189] constructed a Partially Observable Markov Decision Process (POMDP) and use a POMDP planner that allows for belief tracking. Thereby, a plan is considered when the observed action sequence is embedded in the plan. This means that the plan contains the observed actions in the given order but not necessarily without gaps. The probability of each plan is then computed by considering the goal-directed action selection function and the observation likelihood. A probability density function (PDF) over possible goals is then created from the plan probabilities.

Evaluation By applying their approach to three planning domains, the authors show the approach to be feasible. An *office* domain is created based on the experiment of Bui et al. [37] to model the interactions of an agent within the office environment. The domain consists of 2.300 states, twenty-three actions, three goals and four possible initial states. With 30% (resulting in an avg. length of 4.9 items) of the observation sequence, from a set of fifteen observation items the approach is able to recognise the correct goal with an accuracy of .99. Extending the sequence to 70% (avg. length of 10.8) allows the correct goal to be inferred with absolute certainty.

The *kitchen* domain consists of about 70,000 states and twenty-nine possible actions. From the state space, sixteen initial states and five different goal states are selected. Here, the goals are to cook different dishes. Each dish requires up to three ingredients and several kitchen tools, which are placed randomly in the environment. The proposed approach is able to infer the correct goal with an accuracy of .96 (.84) with 70% (30%) of the original observation sequence.

Ramírez and Geffner [189] sketch several possible extensions, all of them increasing the amount of uncertainty. In particular, they suggest to use noisy observations where the set of observation items is unchanged, but an action may produce an observation belonging to another action. An HMM is proposed to handle this additional degree of uncertainty.

Summary Ramírez and Geffner [189] extend the goal recognition approach of Baker et al. [17] by considering incomplete observations and non-deterministic actions. A model-based description (in terms of PDDL) is used. The transition probabilities are not manually specified or learned from a large amount of training data but generated by exploiting the properties of rational agents. However, the proposed approach can be interpreted as a CSSM. The approach allows to account for uncertainty in observations and actions. However, here uncertainty in observation means that some observations are simply dropped from the sequence of observations. This means that although it is not possible to assess the number of missing observations,

observations provided to the observer are still always correct. This is clearly in contrast to the observation quality provided by sensors used for AR that, as in the case of IMUs, even does not allow to directly infer the activity from the observation sequence.

Another issue that prevents the proposed approach from being directly applied to the recognition of human behaviour is the inability of probabilistic modelling of action durations.

Finally, while the approach is, in general, able to handle latent infinite state spaces, the authors state that using the POMDP planner does not scale very well. This means that more sophisticated mechanisms are required in case the state space increases and observation noise, as produced by real sensors, is added.

2.2.8. Accommodating Human Variability in Human-Robot Teams through Theory of Mind by Hiatt et al. [95]

Hiatt et al. [95] are concerned with the problem of variability in human plan execution. They consider the collaboration of humans and robots. The robot is equipped with a model that simulates the human's theory of mind, and whenever there is reason to believe that the human acts incorrectly, the robot interferes. In their work, Hiatt et al. [95] apply a variant of the ACT-R [9] system, specialised for human-robot interaction (ACT-R/E [235]).

Model ACT-R uses declarative and procedural modules to represent declarative knowledge and memories, and production rules. Declarative knowledge is managed in chunks with an activation value. Different situational heuristics about how frequently or recently chunks were accessed control this activation value. A chunk is accessed if it matches the current context information. The activation value is used to decide upon multiple matching chunks. Declarative knowledge is used to represent the current cognitive state. Production rules, including a value of expected utility, represent actions that allow to change the state. The utility function is, among others, based on goal-directed behaviour. Preconditions allow to restrict the set of applicable rules per state (configuration of chunks). A more detailed description of ACT-R can be found in [9]. Models can differ by their initial chunk occupancy, different parameterisations for the activation value and utility functions, and the initial belief about the world.

Inference method By using the ACT-R model, the robot tracks the human teammate. Whenever the human executed an action, unexpected to the robot, several different hypotheses are considered that may cause an unexpected behaviour. For each branch, the probabilities of the different selections are calculated by either relating the current chunk's activation to the overall activation or by relating the utility value of a production rule to the overall sum of all utility values. Here, random noise is added, to account for uncertainty in the human's action selection. In order to explain the variability of the human's behaviour, different hypothetical models that differ in belief and parameterisation are considered.

Evaluation Hiatt et al. [95] performed an evaluation based on a simulation of two different scenarios. In the first scenario, the human-robot team is supposed to patrol in an environment. At some point, the human unexpectedly went to the wrong direction. After considering several hypotheses, such as the learned exception of forgetting the radio, the robot was able to interfere and informed the human about incorrect behaviour. In another experiment, study participants were asked if the robot's behaviour seems to be more intelligent and natural compared to two other behaviours not considering the theory of mind. The participants rated the proposed approach to be more intelligent and more natural, which provides evidence that the approach allows to reproduce human reasoning about observed behaviour.

Summary Similar to the approaches of Baker et al. [17] and Ramírez and Geffner [189], the authors use a model-based description to PR. As Hiatt et al. [95] state, the approach is able to reason about time intervals of seconds and minutes. Computational issues that should be handled by approximate inference, will arise by considering long term simulation. The proposed approach uses a combination of situational heuristics (context matching, recency, and frequency) and utility-based production selection. A model-based specification is used to describe the scenario, which allows for potentially infinite state spaces. Transition probabilities are governed by state properties and action related utility values. Thus, the approach can be considered as a CSSM. ACT-R allows to track human decisions and mental properties such as the content of memory chunks. The authors show that it can also be used for recognising goals and thereby incorrect behaviour. However, in their work, the authors assume the robot to be able to fully observe the person[4]. This is in direct contradiction to the use of sensors, which inherently provide noisy and contradictory observation of the environment.

2.2.9. Location-Based Reasoning about Complex Multi-Agent Behavior by Sadilek and Kautz [202]

Sadilek and Kautz [202] address the problem of recognising the behaviour of multiple interacting persons from GPS data. They use a model-based description of human behaviour in terms of Markov Logic as basis for probabilistic inference. Moreover, they consider the possibility that actions may not be successful, but rather be interrupted or fail.

Model In their approach the state of the world is described by parameterised predicates, each representing one fact about the world. Actions are described in terms of rules. Here, hard and soft rules are distinguished, where both describe facts about how predicates change over time in dependence to other predicates. In contrast to hard rules, which are applied whenever the precondition holds, the weight of soft rules describes how likely such rules are applied. The weights of soft rules are then learned from training data. The set of predicates is divided into observable and hidden predicates. Observable predicates reflect sensor data to state mappings, whereas hidden predicates reflect parts of the state space under question.

Inference method At inference time, based on the observed predicates, the state with the highest probability is estimated. The probability distribution is created by relating the sum of all weights of rules that hold to the overall sum of all world weights. The maximum a-posteriori (MAP) state sequence is inferred by applying a state of the art algorithm for solving Markov Logic Networks, which is constructed by grounding predicates and rules for each time-step. The complexity for inference from one time-step t to the next $t + 1$ is linear in t ($\mathcal{O}(t)$).

Evaluation In an experiment Sadilek and Kautz [202] showed that the proposed approach is able to handle 14 persons. The participants were instructed to play the game "capture the flag", where two teams, each consisting of seven persons, try to conquer the opponent team's territory. The location of each player was tracked by GPS loggers. Four repetitions were conducted, each lasting between four and fifteen minutes. Similar to Bui et al. [37], the area was divided into cells (each with an area of $9m^2$) that were used to "snap" the raw GPS sensor signal. The state of the world was described by 14 predicates, where some were marked as observable. The set of eight hard rules, describing physical constraints or rules of the game, was created. Seven soft rules were used to describe the snapping of GPS data to cells and

[4]One exception for the full observation are obstacles in the direct sight of view. While this introduces additional uncertainty, it still does not introduce noise to the observation.

Approach	Uncertainty	Online	Reusability	Latent Infinity	Plan
Bui et al. [37]	■	■	□	□	■
Liao et al. [149]	■	■	□	□	■
Burghardt et al. [41]	■	■	■	□	■
Baker et al. [17]	□	■	■	■	■
Ramírez and Geffner [189]	□	■	■	■	■
Hiatt et al. [95]	□	■	■	■	■
Sadilek and Kautz [202]	■	□	■	□	■

Table 2.3.: Evaluation of the requirements of selected related work. "■"=Requirement satisfied, "□"=Requirement not satisfied.

how likely events happen. After training the weight of soft rules from three games, the weights were used to infer the sequence of actions in the fourth game. A comparison to several baseline models showed significant improvement of the recognition performance.

Summary In their work, Sadilek and Kautz [202] illustrate how to combine a symbolic description of human behaviour with noisy and ambiguous sensor data. They use a model-based description of the behaviour of multiple interacting agents. They target the recognition of activities and context, such as agents being captured. Training data is used to learn the probabilities for soft rules, which is equivalent to transition probabilities between states. However, several issues prevent the proposed system from being used. Firstly, the approach scales poorly, which make the technique infeasible for online recognition. Secondly, due to the model only supporting rule probabilities, which is equivalent to transition probabilities, the model has no support for probabilistic durations except for the geometric distribution given by the state's self transition. Furthermore, similar to other approaches [37, 149], the proposed approach exploits the fact that location information is relatively easy to interpret. This allows, for example, the GPS location data to be snapped to a grid of cells of interest. The proposed approach is no CSSM as it does not allow for latent infinite state spaces. The state space has to be fully explored at inference time. In addition, transition probabilities have to be learned from training data instead of using heuristic alternatives based on the state space.

2.2.10. Summary

Table 2.2 contains a detailed classification of all considered related work. Additionally, Table 2.3 provides information regarding the satisfaction of the requirements. It can be seen that no approach exists that satisfies all requirements. In general, it can be concluded that the related works either supports sensor uncertainty, or reusability and latent infinity. The idea sketched in the work of Burghardt et al. [42] is the only one that satisfies uncertainty and reusability. Therefore, it seems reasonable to extend their ideas and combine the probabilistic approaches [37, 149] with the model-based specification [17, 189] in order to satisfy all requirements. Sadilek and Kautz [202] show how to support multiple persons. Finally, combining the approaches of Baker et al. [17] and Ramírez and Geffner [189] for goal-based action selection (without the need for training data) and the situational heuristics of Hiatt et al. [95] into a more general framework of action selection is reasonable. As described, Bui et al. [37] and Liao et al. [149] apply the RBPF to reduce inference complexity. Both approaches use a PF for approximate inference in categorical state spaces. The PF has several drawbacks (see Section 3.2). Thus, a novel inference method for categorical state spaces has to be developed.

Based on the work, presented in this chapter, in the next chapter, CCBM is introduced. It is shown that CCBM combines the described capabilities and thus satisfies all requirements.

3
Computational Causal Behaviour Models

"Statisticians, like artists, have the bad habit of falling in love with their models." –
George E.P. Box

SYNOPSIS: *This chapter introduces the Computational Causal Behaviour Model as an instance of CSSM. Based on the concepts introduced in Chapter 2, a statistical model is introduced and described. Finally, the CCBM toolbox, which implements the described features and provides support for experimental validation, is introduced.*

CHAPTER SOURCES: *Parts of this Chapter have been previously published in the following publication(s):*
- *Towards Creating Assistive Software by Employing Human Behavior Models [129]*
- *CCBM – A Tool for Activity Recognition using Computational Causal Behaviour Models [122]*
- *Computational State Space Models for Activity and Intention Recognition. A Feasibility Study [133]*
- *Tool Support for Probabilistic Intention Recognition using Plan Synthesis [130]*
- *Synthesising Sequential Bayesian Filters for Plan and Activity Recognition from Extended Precondition-Effect Rules [127].*
- *Marginal Filtering in Large State Spaces [172]*

The previous chapter provided an overview of the general concept of CSSMs and highlighted their usage in the literature. Additionally, it was argued that the combination of a reusable model-based description for generating the LTS graph of actions and states and a statistical model allowing to account for uncertainties enables high-level reasoning from low-level sensors. This chapter introduces Computational Causal Behaviour Models as one instance of Computational State Space Models designated for recognising human behaviour from noisy and ambiguous sensor data. CCBM employs a model-based specification language based on preconditions and effects to generate an LTS. A statistical graphical model is then used to provide the probabilistic semantics for this LTS.

In the following, first the statistical model that is used for inference based on the LTS graph is described. Based on this model, five sub-models are introduced, each focussing on different aspects. For each sub-model the probabilistic meaning is introduced. With respect to inference algorithms, an approximate inference method based on the PF is described and several issues are discussed. To overcome these drawbacks, a novel inference algorithm – the Marginal Filter – is then introduced. Finally, the CCBM toolbox, including the modelling language and the provided inference tools, is presented.

3.1. Statistical Model

The general framework of CSSMs allows the construction of statistical inference models based on an LTS. Inference within CCBM employs a statistical model to handle noisy and ambiguous sensor data This section describes the DBN, which is constructed by considering the LTS' underlying transition system.

Design rationale CCBM aims at reconstructing the user's current action, the plan and additional contextual information from noisy and ambiguous sensor data. For this purpose, the statistical model has to reflect the user's current action, the goal and contextual information. Additionally, the model has to map the dependencies between actions, states and goals. The execution of an action, for instance, depends on the current state, as not all actions can be executed in each state. Additionally, as users are assumed to be goal-directed, the user is supposed to execute actions depending on the current goal. As a result of an executed action, the environment state changes. Another aspect is that the execution of actions consumes time. Again, this has to be reflected by the statistical model.

The framework of Bayesian filtering has been shown to effectively estimate a dynamic system's state sequence from noisy sensor data [165, p.631ff]. A mechanism to describe the dependencies between multiple random variables in order to allow methods of temporal Bayesian filtering to be applied is provided by DBNs. This section introduces the DBN that is used to reflect the illustrated dependencies.

In general, DBNs distinguish between hidden and observable nodes. The former describe the state of a dynamic system that cannot be concluded directly while the latter describes variables that can be observed. Typically, observable nodes are used to represent sensor observations about hidden states. The framework of Bayesian filtering allows to draw conclusions about the hidden variables from observations about the observable variables.

In addition to the properties that describe the user's behaviour, which are typically modelled as hidden nodes, the model comprises observable nodes that represent sensor data used for inference. As discussed earlier, two different kinds of sensor data can be distinguished: state and action observation. As CCBM is supposed to used both types, two observable nodes have to be added to statistical model, each of them influenced by the respective node of the user's behaviour. In the following the resulting DBN is described in detail.

Probabilistic structure The DBN used as probabilistic model in the CCBM framework is illustrated in Figure 3.1. The hidden state X_t is a five-tuple $(A_t, D_t, G_t, S_t, U_t)$. Here, we call X_t the X state and S_t the S state. The action executed at time t is denoted by A_t, D_t is a boolean flag signalling whether A_{t-1} is terminated in the left-open and right-closed interval $(t-1, t]$ and a new action A_t has to be selected. U_t is the starting time of the action A_t. The variable S_t is the current environment state and G_t denotes the goal (or intention) of the user

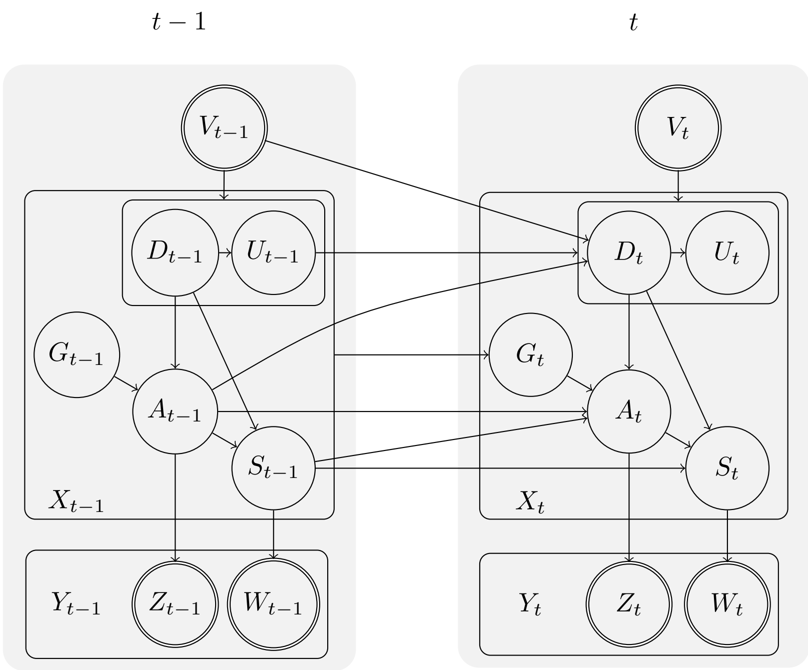

Figure 3.1.: Graphical model used in CCBM. Nodes with single outline denote hidden random variables, whereas nodes with double outline denote observable random variables. Boxes represent tuples of variables. Edges represent dependencies of a (tuple of) random variable(s) (edge ending) from another (tuple of) random variable(s) (edge starting).

at time t. The observation $Y_t{=}(W_t, Z_t)$ for time-step t consists of two conditional independent parts, the state observation W_t and the action observation Z_t. V_t is the timestamp of the observation data sequence.

The boolean variable D_t introduces a context specific independence [32] into the DBN. If $d_t{=}false$ the variables U_t, S_t, and A_t carry over their values from the last time slice and are independent from their other parents. The value of $d_t{=}true$ means that a new action has to be selected. As a result, the starting time U_t and the system state S_t have to be updated. The dependency of A_t on S_{t-1}, A_{t-1}, and G_t allows to represent situation-driven as well goal-driven behaviour. If a new action is selected, the value of S_t is the result of applying A_t to the previous state S_{t-1}. This directly denotes the effect of actions on the environment.

Extensions for multiple agents The model described above only incorporates a single execution instance (namely, one agent). The support for multiple execution instances is introduced by different refinements to the action related nodes. To provide several execution instances with different action sets, the model refines the nodes A_t, D_t, and U_t by sets of nodes $\mathcal{A}_t$, $\mathcal{D}_t$, and $\mathcal{U}_t$. Each set then consists of one node per execution instance. The node S_t, which represents the environment state is not further refined as all agents are assumed to act within the same environment. Consequently, this means that each execution instance (an agent) has knowledge about the entire environment. This refinement to the action related nodes is for instance done in Experiment X1 and Experiment X3 (see Section 6.1 and Section 6.3).

The underlying assumption of a common goal node G_t is that a common goal is sufficient as long as we concentrate on collaborating agents. Modelling non-collaborating agents or even adversarial agents would require a refinement of G_t. To support different goals for each agent, the node G_t is further refined by the set of goals $\mathcal{G}_t$ – one goal for each agent. While the common goal node is used in Experiment X1, Experiment X3 uses separate goals for each agent.

Depending on the kind of observation data, the action observation node Z_t can be refined to a set of nodes $\mathcal{Z}_t$, where each node corresponds to one agent. Shared observation nodes, in contrast, are necessary when the assignment of observations to execution instances is not possible. This is, for instance, the case in Experiment X3, where state observations are used. Separate observation nodes for each execution instance would imply that each agent has a distinguishable and independent set of sensors. While this is typically the case for wearable sensors such as accelerometers, environmental sensors such as motion detectors do not allow this separation. The model depicted in Figure 3.1 can be seen as a generalisation of the different multi-agent refinements. Additionally, the support for multiple agents represents a prerequisite for the requirement R5.3.

Factorisation into five sub-models From the viewpoint of Bayesian filtering, the DBN can be used to derive a probabilistic interpretation of the transition model used for example for the *Forward Filtering Recursion* [165, p.609]. The state X_t is now represented by the 5-tuple $(A_t, D_t, G_t, S_t, U_t)$. For each node in the DBN, one sub-model can be factorised.

$$\begin{aligned} p(X_t \mid X_{t-1}) = & \quad p(A_t, D_t, G_t, S_t, U_t \mid A_{t-1}, D_{t-1}, G_{t-1}, S_{t-1}, U_{t-1}, V_t, V_{t-1}) \\ = & \quad p(S_t \mid A_t, D_t, G_t, U_t, A_{t-1}, D_{t-1}, G_{t-1}, S_{t-1}, U_{t-1}, V_t, V_{t-1}) \\ & \quad p(A_t \mid D_t, G_t, U_t, A_{t-1}, D_{t-1}, G_{t-1}, S_{t-1}, U_{t-1}, V_t, V_{t-1}) \\ & \quad p(U_t \mid D_t, G_t, A_{t-1}, D_{t-1}, G_{t-1}, S_{t-1}, U_{t-1}, V_t, V_{t-1}) \\ & \quad p(D_t \mid G_t, A_{t-1}, D_{t-1}, G_{t-1}, S_{t-1}, U_{t-1}, V_t, V_{t-1}) \\ & \quad p(G_t \mid A_{t-1}, D_{t-1}, G_{t-1}, S_{t-1}, U_{t-1}, V_t, V_{t-1}) \end{aligned}$$

By analysing the dependencies of the graphical model, each of the five sub-models can be simplified. Each sub-model is thereby used to determine the value of the corresponding random variables.

$$\begin{aligned} & p(S_t \mid A_t, D_t, S_{t-1}) && \blacktriangleright \textit{action execution model} \\ & p(A_t \mid D_t, G_t, A_{t-1}, S_{t-1}) && \blacktriangleright \textit{action selection model} \\ & p(U_t \mid D_t, U_{t-1}, V_t) && \blacktriangleright \textit{action start time model} \\ & p(D_t \mid A_{t-1}, U_{t-1}, V_t, V_{t-1}) && \blacktriangleright \textit{action duration model} \\ & p(G_t \mid X_{t-1}) && \blacktriangleright \textit{goal selection model} \end{aligned}$$

In the following, the different sub-models are described and their probabilistic meaning is introduced.

3.1.1. Action Execution Model

Objective Objective of the action execution model is to represent in how far the actions of the human protagonist influence the state environment. The action execution model "executes" a selected action and thereby changes the state according to the action's effect. To this end, it provides a probabilistic semantics for the action effects.

General framework The action execution model describes the PDF of the resulting state S_t after applying action A_t in state S_{t-1}. Depending on the value of D_t, the value is either copied or, if $d_t{=}true$, it is the result of applying the selected action to the previous state.

$$p(s_t \mid a_t, d_t, s_{t-1}) = \begin{cases} 1, & \text{if } d_t{=}false \wedge s_t{=}s_{t-1}, \\ 0, & \text{if } d_t{=}false \wedge s_t{\neq}s_{t-1}, \\ \psi(s_t \mid a_t, s_{t-1}), & \text{otherwise} \end{cases} \tag{3.1}$$

Design considerations With respect to the effects, two aspects have to be considered:

1 whether action effects are deterministic or not and
2 the point in time, when the action effect is applied to the state.

The representation of the action execution model in in (3.1) captures both, deterministic as well as non-deterministic effects. Non-deterministic action effects are typically considered within the framework of POMDPs [189]. They allow to model the agent's knowledge about the environment to be incomplete (e.g. the protagonist is not sure about the location of a tool). Furthermore, non-deterministic effects allow to model actions that can fail (e.g. the protagonist needs several tries to complete an action). Actions with deterministic effects, in contrast, ensure that a modelled effect becomes true after the action is executed. Deterministic action effects are used within MDPs as for instance in Baker et al. [17]. Both types of effects provide the same power of expressiveness, as non-deterministic effects can be coped by non-deterministic choice of deterministic actions. Consider an action a with preconditions π_a, prior probability p_a, and n probabilistic effects $\epsilon_a^{(i)}$. Each action effect has probability p_i. The non-deterministic choice of effects can be governed by n actions a_i with precondition $\pi_{a_i} = \pi_a$ and deterministic effects $\epsilon_{a_i}{=}\epsilon_a^{(i)}$. The probability of selecting action a_i is then given by $p_a \cdot p_i$. This could be implemented by introducing an additional action selection feature that represents the probability of each effect.

Regarding the time where the effect is applied, different alternatives are possible. A distinction is required, when action durations are introduced, only. Typically, three different types of effects can be distinguished: (1.) effects that becomes true at the start of the action (e.g. an agent that moves from a to b, leaves the location a at the start of the action), (2.) effects that becomes true at the end of the action (e.g. the agent arrives at location b at the end of the action), and (3.) incremental effects, where different effects become true during action execution (e.g. while the agent moves from a to b, it passes a third location c). The three alternatives can be covered by tuples of actions with start time effects. Consider an action a with preconditions π_a and effects $\epsilon_{a_1}, \ldots, \epsilon_{a_n}$, different effects for different points in time. Action a can be split into an inseparable action tuple $(a_1, \ldots, a_n)$. The preconditions of action a_1 are set to π_{a_1}, the effects to ϵ_{a_1}. The effects of each action $a_i \in a_2, \ldots, a_n$ are set to the respective effects ϵ_{a_i}. From the modelling perspective, lock predicates [256] can be applied to create inseparable action tuples[1].

CCBM realisation With respect to the time of the effect of actions, in CCBM, actions use start time effects. Action effects are instantaneous, meaning that if an action is applied at time t, the effects are true at time t. This allows to cover all alternatives by action tuples and an explicit duration model of each part of the entire action. As a result, durations are provided for actions rather than for effects.

Regarding the effects, within the framework of CCBM, we assume the human protagonist to have complete knowledge about the environment. As a result, deterministic effects are

[1]Introducing such dependencies seem to break up the independencies of actions at first glance. However, as the introduction of lock predicates is straight forward it can easily be automated by a preprocessing step, which in return allows the other action specifications to remain unchanged.

considered sufficient for the scope of this thesis. Furthermore, the use of deterministic action effects allows the usage of standard planning heuristics for goal-directed action selection (e.g. landmarks). As discussed, non-deterministic effects can easily coped by non-deterministic choice of actions.

The probability of state s_t, given a_t and s_{t-1} is 1, if s_t is the result of executing action a_t in state s_{t-1}.

$$\psi(s_t \mid a_t, s_{t-1}) = \begin{cases} 1, & \text{if } s_t = a_t(s_{t-1}) \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

3.1.2. Action Start Time and Action Duration Models

Objective Actions in the real world consume time. Additionally, the specific duration is often not known beforehand. Modelling actions of human protagonists, thus, requires to model action durations in a probabilistic way. For this purpose, both the action start time model and the action duration model introduce a probabilistic duration model for actions.

General framework The action duration model provides a PDF that determines the value of the variable D_t. Depending on the specific action and the current point in time, the action duration model has to determine the probability that the action has been finished during the last time interval. The action start time model maintains the start time of the action which is used together with the current time to determine the duration of the action so far.

Design considerations The literature uses different mechanisms to provide duration models. On the one hand, approaches like HMMs model durations by geometric distribution functions [108]. For instance, Bui et al. [37] use this type of duration modelling. Other approaches, like Ziparo et al. [265] restrict action durations to the χ^2 distribution. Different analyses showed, for example, the log-normal distribution to effectively describe action durations [140, 226].

A restriction of the duration model to the usage of one PDF allows to specialise the inference mechanism to that function. A restriction to the geometric distribution, for instance, allows to cover the action duration by the specification of the probability of the self transition. However, the choice of the correct action duration function depends of the application domain and the specific action to be executed.

A common method to represent the duration of actions independent from the specific function is the hazard function, which provides the probability that an action stops, given that it was still active in the previous time-step. To implement the action durations based on the hazard function, the start time of the current action as well as the current time have to be provided.

CCBM realisation In order to allow a large variety of action duration functions, CCBM employs an action duration model based on the hazard function. Consequently, the statistical model has to provide the action start time and the action duration model.

The action start time model $p(U_t \mid D_t, U_{t-1}, V_t)$, describes the PDF of the starting time U_t of an action A_t. As long a the action does not change ($d_t{=}false$), the value of U_t is carried over from the last state. In case a new action is selected at time-step t the starting time U_t is set to the current time V_t.

$$p(u_t \mid d_t, u_{t-1}, v_t) = \begin{cases} 1, & \text{if } (d_t{=}false \wedge u_t{=}u_{t-1}) \vee (d_t{=}true \wedge u_t{=}v_t) \\ 0, & \text{if } (d_t{=}false \wedge u_t{\neq}u_{t-1}) \vee (d_t{=}true \wedge u_t{\neq}v_t) \end{cases} \tag{3.3}$$

Based on the action start time model, the action duration model describes the probability that an action will stop based on the hazard function. The action duration model determines

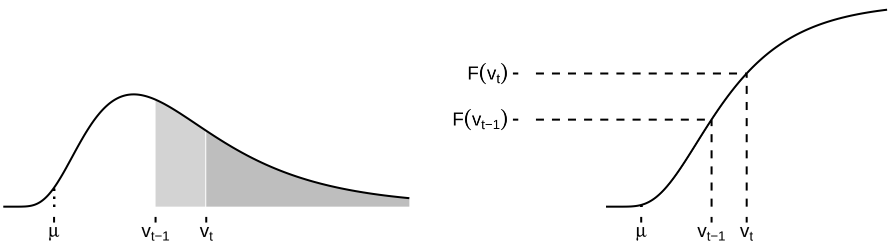

Figure 3.2.: Example of an action duration PDF and the resulting CDF. Example of a log-normal distributed ln $\mathcal{N}(\mu = 1, \sigma^2 = .5)$ action duration. Left: The probability density function, grey area $1 - F(v_{t-1})$, dark grey area $1 - F(v_t)$, light grey area $F(v_t) - F(v_{t-1})$. Right: The cumulative density function.

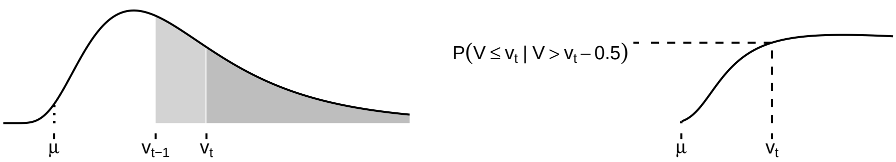

Figure 3.3.: Example of an action duration PDF and the resulting action duration function. Example of a log-normal distributed $ln\,\mathcal{N}(\mu{=}1, \sigma^2{=}.5)$ action duration and resulting action duration function. A time-step interval of .5 is chosen to illustrate the resulting function.

the duration of an action A_t and thus the value of D_t. The Bernoulli random variable D_t is defined by the action duration distribution $p(D_t \mid A_{t-1}, U_{t-1}, V_t, V_{t-1})$. The probability of action a to terminate in the interval $(v_{t-1}, v_t]$ is given by the differences of the cumulative distribution function (CDF) $F(v \mid a, u)$ at v_{t-1} and v_t. F here describes the CDF of the particular action duration function.

$$p(d_t \mid a_{t-1}, u_{t-1}, v_t, v_{t-1}) = \frac{F(v_t \mid a_{t-1}, u_{t-1}) - F(v_{t-1} \mid a_{t-1}, u_{t-1})}{1 - F(v_t \mid a_{t-1}, u_{t-1})} \tag{3.4}$$

Figure 3.2 illustrates an example action duration distribution function (left) and the according CDF (right) based on a log-normal distribution $ln\ \mathcal{N}(\mu{=}1, \sigma^2{=}.5)$. The grey (light and dark) areas illustrate the probability that the action is not terminated until time-step v_{t-1}. The light-grey area highlights the probability of the action to stop in the interval $(v_{t-1}, v_t]$, whereas the dark-grey area gives the probability that the action continues after time-step v_{t-1}. Figure 3.3 exemplifies the contrast between the example action duration CDF and the resulting density function of the actual termination probabilities.

3.1.3. Goal Selection Model

Objective As discussed in Chapter 1, we focus on rational agents that try to achieve a goal. Objective of the goal selection model is to provide a mechanism that allows to select and change a goal to follow. In result, this model allows to reason about the goal of the human protagonist. To this end, the goal selection model provides a probabilistic meaning for the freedom of the protagonist to choose a goal.

General framework The goal variable G_t plays a central role for goal-directed behaviour, as it represents the set of states that are tried to achieve. Consequently, the goal variable has to be adjusted to the goal under question. In general, the selection of the goal depends on the current state. Thus, the goal selection model represents the conditional PDF for $p(G_t \mid X_{t-1})$ accordingly.

Design considerations Different researchers use goal selection models in their approaches. Baker et al. [17], for instance, implement knowledge about changing goals such as sequences or hierarchies of goals. Both variations are implemented for example in the models M_2 and M_3 in [17]. Patterson et al. [179] make use of changing goals by introducing an explicit goal switching node to signal whether a new goal has to be selected or not. Ramírez and Geffner [189], in contrast, use goals that are fixed.

The use of hierarchical goals, as suggested by Blaylock and Allen [26] or Bui [36], implies a temporal ordering on the sets of goals. This hierarchy can be further extended until the bottom of the hierarchy only consists of states that are directly connected by actions. This hierarchy eases inference by reducing the number of possible plans but also decreases the level of freedom of the model. However, the use of sequences or hierarchies of goals requires knowledge about the target application, as the different sub-goals have to specified by the model developer.

CCBM realisation In the scope of this thesis, we focus on fixed and non-hierarchical goals, which are sufficient to address the question **IQ**. This type of goal representation was also used by Ramírez and Geffner [189]. Once a goal is selected, it will not be changed later. The goal G_t is therefore directly carried over from the previous state.

$$p(g_t \mid g_{t-1}, s_{t-1}, a_{t-1}, d_{t-1}, u_{t-1}) = \begin{cases} 1, & \text{if } g_t = g_{t-1} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

At time $t_0 = 0$ the prior goal probability $p(G_0^{(i)})$ for each goal i is set. This is similar to the concept of goal-based agents [200, p. 52].

The proposed representation of the goal selection model allows two different ways of goal recognition. Intra-model goal recognition infers the most likely goal by use of a model that embodies different goals. Baker et al. [17] and Liao et al. [149] for example successfully illustrated the usage of this technique. Inter-model goal recognition, in contrast, uses one goal per model and applies model selection techniques to select the most likely model and thus the most likely goal. Inter-model goal recognition was, for instance, utilised by Ramírez and Geffner [189] and Armentano and Amandi [12]. A brief discussion about advantages and disadvantages of intra- and inter-model goal recognition is given in Section A.4. The experiments, presented in this thesis apply intra-model goal recognition.

3.1.4. Action Selection Model

Objective Objective of the action selection model is to reflect the human protagonist's choice to select an action. To this end, the action selection model accomplishes two main tasks: (1.) it represents the goal-directed behaviour of the human protagonist and (2.) it describes deviations from the "best possible" action sequence – it embodies the "free will". The aim of this model is to provide a PDF that allows to reproduce what actions the human protagonist selects in a given situation. This differs from the "behavioural" sciences that target questions about how humans select actions (e.g. Tenenbaum et al. [233] and Gigerenzer [81]).

General framework The action selection model represents the non-deterministic choice if multiple actions are applicable in a given situation. A new action is selected if the boolean flag D_t signals the termination of action A_{t-1}.

$$p(a_t \mid d_t, g_t, a_{t-1}, s_{t-1}) = \begin{cases} \gamma(a_t \mid g_t, a_{t-1}, s_{t-1}), & \text{if } d_t = true \\ 0, & \text{if } d_t{=}false \wedge a_t{\neq}a_{t-1} \\ 1, & \text{if } d_t{=}false \wedge a_t{=}a_{t-1} \end{cases} \tag{3.6}$$

The new action is determined by the action selection function γ.

Design considerations According to Prescott et al. [183] the human choice of an action is influenced by a large set of factors. This includes situation-based conflict resolution strategies, as the specificity or the non-refractoriness known from the cognitive architecture ACT-R [8, pp.132–137]. Other action selection mechanisms are based on the action's utility in reaching the goal from the given state as known from the domain of automated planning. Ramírez and Geffner [189] and Baker et al. [17] employ a goal distance based action selection mechanism to reflect rational agents that select actions to decrease the distance to the goal. Beside the usage of the exact goal distance, the literature employs different approximations [93, 100].

The goal distance (and its approximations) provide knowledge about a model, independent from training data. Models of machine learning, in contrast, could provide an action selection feature that is adjusted to the actual human choice by employing training data. Inverse reinforcement learning [200, p.857] is a prominent approach to learn a policy about action selection from training data. Other training-based approaches employ SVMs or neural networks [204] to provide goal distance estimates.

To summarise, several approaches to implement a probabilistic action selection framework exist, they range from the application of training free features (e.g. goal distance or specificity) to approaches that employ methods of machine learning. A common framework that allows combinations of different features, however, is missing.

CCBM realisation A powerful mechanism to combine different factors, independent from their nature, are log-linear models [23]. The integration of different factors f_k into the action selection function γ via a log-linear model is given by:

$$\tilde{\gamma}(a_t \mid g_t, a_{t-1}, s_{t-1}) = \exp\big(\sum_{k \in K} \lambda_k f_k(a_t, g_t, a_{t-1}, s_{t-1})\big) \tag{3.7}$$

$$\gamma(a_t \mid g_t, a_{t-1}, s_{t-1}) = \frac{1}{Z}\tilde{\gamma}(a_t \mid g_t, a_{t-1}, s_{t-1}) \tag{3.8}$$

$$Z = \sum_{a \in A} \tilde{\gamma}(a_t \mid g_t, a_{t-1}, s_{t-1}) \tag{3.9}$$

Here, Z is the normalisation constant, $f_k : \mathcal{A} \times \mathcal{G} \times \mathcal{A} \times S \to \mathbb{R}$ the features and $\lambda_k \in \mathbb{R}$ their corresponding weights.

Beside the fact that human action selection is a topic under current investigation, the statistical model used throughout this thesis implements several action selection features from the set of goal distance based features. The following features are used within the investigations on the influence of goal distance based action selection to the recognition performance:

f_δ : The probability of an action is inverse proportional to the goal distance of the resulting state. Thus, actions that decrease the goal distance are preferred. The exact goal distance is computed by an exhaustive process. The algorithm of Dijkstra [63] is used to find the shortest path to the goal for each node.

$f_{\bar{\delta}}$: An approximation of the goal distance that only considers states that were reached by the actions actually executed during the experiment. This feature is an example for training based mechanisms to action selection.

f_h : An approximation of the goal distance based on the experimental description. Single consecutive steps are extracted and used for goal distance determination (see Table 6.6 for a specific task script).

f_π : The action's precondition that allow to restrict the applicability of that action to a subset of the state space can be interpreted as an "enabling" feature.

Note that for the enabling feature f_π, the weight λ_π is set to $-\infty$, which results in a probability of 0 for actions with preconditions not satisfied by the current state. A change to this weight would change the transition matrix from sparse to full.

3.1.5. Observation Model

Beside the definition of the system model, the framework of Bayesian filtering (see Section 3.2) requires the specification of an observation model (also known as sensor model). The observation model is used to incorporate knowledge about the uncertainties of the sensors by providing the PDF $p(Y \mid X)$, where Y represents the sensor data and X the system's state.

Action and state observation model The statistical model, introduced above, allows two hidden variables to be observed, the action A and the environment state S. For each of them a corresponding observable node is introduced: the Z_t represents observations about the action A_t, whereas the W_t node models observations about the state S_t. The observation model is given as follows:

$$\begin{aligned} p(Y \mid X) &= p(Y \mid A, S) \\ &= p(Z, W \mid A, S) \end{aligned}$$

The graphical model in Figure 3.1 renders Z and W as conditional independent, which essentially allows to represent the observation model as a combination of two separate sub-models: the action observation model $P(Z \mid A)$ and the state observation model $P(W \mid S)$.

$$p(Y \mid X) = p(Z \mid A)p(W \mid S)$$

The action observation model allows to incorporate sensors that directly observe actions, such as wearable sensors. Typical instances of such sensors are IMUs. This was also shown by the analysis of the related work on the domain of activity recognition in Section 2.1.2. The state observation model, in contrast, allows the usage of sensors that observe the state of the environment. Localisation sensors such as GPS or PIR sensors are typical examples for sensors that provide observation about the state. As discussed in Section 2.2, most approaches focus on one observation model. A general approach, however, has to provide both sub-models. Furthermore, providing an interface for both types of observation is a prerequisite for R5.2 as it allows to exchange the observation model without changes to the system model. The usage of state observation is, for instance, illustrated in Experiment X1 and Experiment X3. Action observations are used in Experiment X2.

Independent and identically distributed observations One important aspect of the observation model is that the statistical model is assumed to be stationary [165, p.589]. This means that the observations depend on the state (or action) only, rather than on the duration of the action. Consequently, for each state (or action) the observation have to be independent and identically distributed (i.i.d.). For human behaviour, the assumption of stationarity of the observation model often does not hold, as an action may consist of different phases. Observations in the middle of the execution of an action might be more typical for the action. When considering the action of taking an object, for instance, first the arm has to be moved to the object, then the object has to be grasped and finally, the arm has to be moved back. As a result, observations at the start of an action differ from observations in the middle of the execution. Additionally, the beginning or the end are influenced by the previous or the following action, resulting in a "mix" of actions. The resulting effect is that from the beginning the probability of an action class (given the observations) increases, is stable in the middle of the execution, and decreases at the end. This is clearly a violation of the requirement of i.i.d. samples and represents a potential drawback, especially for actions with long durations. Additionally, the correct detection of action transitions is impacted, which consequently results in lower recognition performance.

One solution to this problem is to incorporate temporal knowledge about the action's process into the observations [158]. The use of hierarchical models, where different sub-models are used to represent the temporal structure of actions and thus the temporal change of observations [176] can be seen as generalisation. Another solution, albeit being impractical for online recognition, is to scramble the observations of the action class. A within action class scrambling requires the sensor data to be preprocessed and annotated beforehand. This will prevent that the observation probability increases at the beginning and decreases and the end of an action.

3.2. Inference Algorithms

Objective of the inference is to estimate the state sequence of the dynamic system under observation. Section 3.1 introduced a DBN that allows to estimate the sequences of states from noisy and ambiguous sensor data. The implementation of the Bayesian filtering algorithm strongly depends on the representation of the individual sub-models and the belief state. Depending on the sub-model, specialisations as HMMs or Kalman filters [214] can be applied, allowing optimal solutions of the above equations. Those methods allow the belief state to be represented by either a parametric form, as in the case of Kalman filters, or by explicitly enumerating all possible states as in the case of HMMs.

Approximate filtering Accurate handling of the belief state is a prerequisite for effective state sequence estimation, as only accurate knowledge about the current time-step's state density $p(X_t \mid y_{1:t})$ allows to correctly predict further state changes. However, in order to allow specialised methods for exact computation to be applied, the representation of the state space has to be restricted. Large state spaces, for example, prevent an exact representation of the belief state and thus prevents the application of exact methods. In fact, the large, possibly infinite, state space, generated by computational action languages requires an approximate representation of the belief state. Consequently, approximate filtering methods have to be used instead. If, in general, the exact representation or computation is infeasible, the density $p(X_t \mid y_{1:t})$ is approximated by a simpler density $\hat{p}(X_t \mid y_{1:t})$. An alternative to the application of approximate methods is a simplification of the state space. Section A.3 gives an overview how the state space can be reduced in order to apply exact filtering methods such as HMMs.

The following sections describe the inference methods used within the framework of CCBM throughout this thesis. Based on the graphical model (see Section 3.1), it is illustrated how inference is done in CCBM.

3.2.1. The Particle Filter

In contrast to exact methods for inference, the PF [13] approximates the belief state by a set of weighted samples. Particle filters are widely used for estimating states in complex systems [124]. They have also been successfully used for recognising activities from uncertain sensor data [169, 212, 43, 97].

Belief state representation In the PF, the density $\hat{p}(X_t \mid y_{1:t})$ is approximated by:

$$\hat{p}(X_t \mid y_{1:t}) = \sum_{i=1}^{N} \omega_t^{(i)} \delta(X_t = x_t^{(i)}) \tag{3.10}$$

$\delta(X = x)$ represents the Dirac delta function at x. A set of weighted samples $\langle x_t^{(i)}, \omega_t^{(i)} \rangle$, namely particles, with $\sum_{i=1}^{n} \omega_i{=}1$ is used to represent the density. Here, one particle represents the tuple $X{=}(S, A, D, U, G)$.

The standard PF, as usual for Bayesian filtering, works by employing predict and update cycles. The prediction uses a proposal function to generate a new state from the current state. Here we use the bootstrap PF [68], where the system model is utilised as a proposal function ($x_t^{(i)} {\sim} p(X_t \mid x_{t-1}^{(i)})$). During update, sensor observation data is used to compute the observation density $p(y_t \mid x_t)$ by making use of the observation model. This step leads to updated weights and requires a normalisation to be executed afterwards.

To prevent the PF from degenerating, an artificial reduction of variance is introduced by resampling [68] the most likely particles. While this reduces the diversity of the particles, it simultaneously focusses on the most likely states. Resampling is usually performed when the effective number of particles $N_{eff}{=}(\sum_{i=1}^{N} \omega_i^2)^{-1}$ drops below a threshold [67].

Algorithm 1 illustrates PF based inference in CCBM. The predict step first samples d_t to determine whether the current action a_{t-1} should have been terminated in the elapsed interval $(v_{t-1}, v_t]$ or not. If the action a_{t-1} terminates, a new action is sampled according to the action selection function $\gamma(a_t \mid g_t, a_{t-1}, s_{t-1})$. The sampled function a_t is applied to the past state s_{t-1} resulting in the new state s_t. In the update step, both observation models, the action observation model $p(z_t \mid a_t)$ and the state observation model $p(w_t \mid s_t)$ are used to update the particle weights. Resampling is performed in order to ensure better particle distribution. Here, systematic resampling [65] is employed, as it has low computational requirements. The complexity of forward filtering in the PF is $\mathcal{O}(N\,T)$. $\mathcal{O}(1)$, for one time-step with fixed N.

Particle filters for continuous state spaces Particle filters are designed to estimate the joint density of state sequences in continuous state spaces but often also employed in categorical state spaces (see for instance [37, 149]). The probability of one state is represented by the density of samples in the vicinity of this point. Particles are distributed to occupy different points in the state space. A "distance" is used to control the influence of particles to the probability density. This "distance-based" probability works well for continuous state spaces, where a distance can be defined naturally. This allows particles to "smoothly" wander to the correct state. In contrast, categorical state spaces effectively prevent the use of such distance measure. Different points in the state space represent different states without any meaningful distance. Additionally, the frequent use of resampling leads to equal particle weights. This essentially means probabilities are not represented by weights but rather by numbers of particles. *"The main difficulty for standard particle filters is that they will quickly force all of the particles to be the same or nearly the same as the most likely particle. In addition, the propagation mechanism causes most of the particles to follow very similar paths through the network."*,([212]). The effect of "particle clinging" is a direct result of this property.

Algorithm 1 The particle filter forward filtering algorithm

$t \leftarrow 1$
for $i \leftarrow 0; i < N; i \leftarrow i + 1$ **do**
    sample $g^{(i)} \sim p(g)$ ▷ Sample goal
end for
while sensor data available **do**
    read (v_t, w_t, z_t) from sensor data
    for $i \leftarrow 0; i < N; i \leftarrow i + 1$ **do** ▷ Iterate all particles
        sample $d_t^{(i)} \sim \tau(T \leq v_t \mid a_{t-1}^{(i)}, u_{t-1}^{(i)}, T > v_{t-1})$ ▷ sample action termination
        $g_t \leftarrow g_{t-1}$
        if $d_t^{(i)}$ **then** ▷ Action a_{t-1} terminates
            sample $a_t^{(i)} \sim \gamma(A_t \mid g_t^{(i)}, a_{t-1}^{(i)}, s_{t-1}^{(i)})$ ▷ Sample new action
            $s_t^{(i)} \leftarrow a_t^{(i)}(s_{t-1}^{(i)})$ ▷ Resulting state
            $u_t^{(i)} \leftarrow u_{t-1}^{(i)}$ ▷ Starting time
        else
            $a_t^{(i)} \leftarrow a_{t-1}^{(i)}$; $s_t^{(i)} \leftarrow s_{i-1}^{(i)}$; $u_t^{(i)} \leftarrow u_{t-1}^{(i)}$ ▷ Action continues
        end if
        $\tilde{\omega}_t^{(i)} \leftarrow \omega_{t-1}^{(i)} \cdot p(z_t \mid a_t^{(i)}) \cdot p(w_t \mid s_t^{(i)})$ ▷ Update particle weight
    end for
    for $i \leftarrow 0; i < N; i \leftarrow i + 1$ **do**
        $\omega_t^{(i)} \leftarrow \tilde{\omega}_t^{(i)} / \sum_{j \in N} \omega_t^{(j)}$ ▷ Normalise weights
    end for
    RESAMPLE() ▷ Perform resampling
    $t \leftarrow t + 1$
end while

3.2.2. The Marginal Filter

Particle filters for categorical state spaces As briefly discussed, the PF is inappropriate for tracking categorical state spaces. However, it is the favoured approximate filtering technique (see Table 2.2) for AR even in discrete state spaces. It is, for example, used by Nguyen et al. [169] to recognise behaviours within an office environment. In the PF each particle samples one successor, by selecting one applicable action for the represented X state according to the action selection function γ. Consequently, each particle could easily contain the list of all past X states, including the S state s_t, the action a_t, and the starting time u_t. In fact, the PF estimates the joint probability $p(X_{1:t} \mid y_{1:t})$ instead of the marginal probability $p(X_t \mid y_{1:t})$. Here, computing the marginal probability is sufficient, as for each time-step t we are only interested in the current X state $x_t = (s_t, a_t, g_t, u_t, d_t)$.

Marginal filtering in the literature Klaas et al. [124] were the first to use the marginal filtering. The key idea of the marginal PF [124] is to create the marginal distribution by summing up the probabilities of duplicated X states before sampling successor states. During sampling, for N particles N successor particles are sampled. This leads to duplicated X states being sampled, which is no issue in continuous state spaces as used by Klaas et al. [124]. Another observation of their approach is that, as in the standard PF, a sampling-based prediction is used. This typically results in less likely transitions being to be omitted, even if the sensor data might recover the probability. The complexity of the marginal PF in $\mathcal{O}(N^2 T)$, which is $\mathcal{O}(N^2)$ for one filter step and $\mathcal{O}(1)$ if N is assumed to be constant.

Algorithm 2 The marginal filter forward filter algorithm

$m_0 \leftarrow$ NewMap
for all $g \in \mathcal{G}$ **do**
    Put$(m_0, ((s = s_0, g = g, a = \mathit{initAction}, u = 0), w = p_g))$ ▷ Initialise
end for
$t \leftarrow 1$
while sensor data available **do**
    read (v_t, w_t, z_t) from sensor data
    $m_t \leftarrow$ NewMap
    for all $(x = (s, g, a, u), \omega) \in m_{t-1}$ **do**
        $p_d \leftarrow \tau(T \leq v_t \mid a_{t-1}, u_{t-1}, T > v_{t-1})$
        Put$(m_t, (x, \omega \cdot (1 - p_d)))$
        for all $a' \in \mathcal{A}$ **do**
            $\omega \leftarrow p_d \cdot \gamma(a' \mid g, a, s)$
            $s' \leftarrow a'(s); u' \leftarrow t$ ▷ Apply action
            if $(x = (s', g, a', u'), \omega') \in m_t$ **then**
                Update$(m_t, (x, \omega' + \omega))$ ▷ Merge particles
            else
                Put$(m_t, (x = (s', g, a', u'), \omega))$ ▷ Insert State
            end if
        end for
    end for
    for all $(x = (s, g, a, u), \omega) \in m_{t-1}$ **do**
        Update$(m_t, (x = (s, g, a, u), \omega \cdot p(z_t \mid a) \cdot p(w_t \mid s)))$ ▷ Update weight
    end for
    Prune(m_t); Normalise(m_t)
    Delete$(m_{t-1}); t \leftarrow t + 1$
end while

Shi et al. [212] introduced the D-Condensation filter, which is particularly designed for categorical state spaces. They propose a discrete version of PF, where, instead of sampling, all successor states are computed for all current X states. Shi et al. [212] introduced marginal filtering to cope with large belief states that result from the usage of a duration model. A state is represented by the set of all actions, each of them being active or not. Thus, in each time-step, the successor states (at most 2^k) will be computed from one state with k actions, as each action can stay active or transit to inactive. If duplicate states are created during prediction, they are merged and their probabilities summed up. This is similar to marginalising in continuous state spaces as done by Klaas et al. [124]. In order to restrict the maximal number of tracked states Shi et al. [212] select the N most likely states. This technique is called pruning and is implemented by means of beam search. The D-Condensation algorithm is only applicable efficiently if the number k of actions is very low, as it requires to compute at most 2^k successor states before pruning. Otherwise, the prediction step might require $\mathcal{O}(N^2)$ memory, which is infeasible for large k. The complexity of the D-Condensation filter is $\mathcal{O}(N\,T)$, $\mathcal{O}(N)$ for a single filtering step.

With respect to the LTS, as discussed in Section 2.1.4, the D-Condensation filter would represent the complete state space of the LTS. While this is obviously infeasible for infinite state spaces, it requires high amounts of memory even for small LTSs. This is due to the assumption that all k actions may be active at the same time, while in the LTS only one step can be active at each time-step.

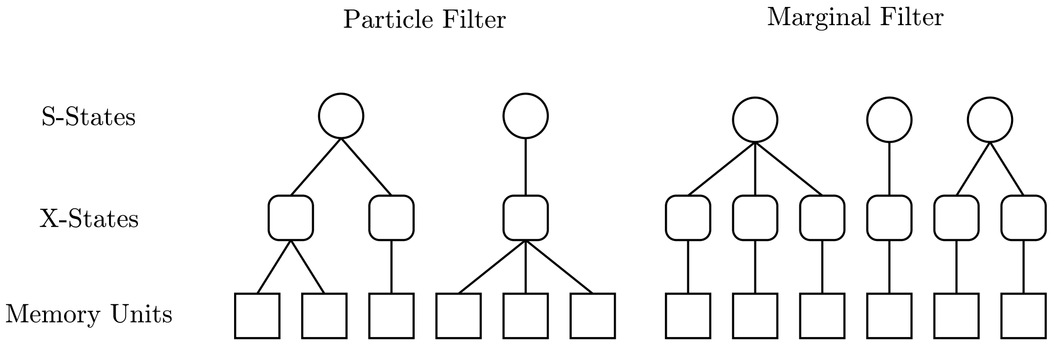

Figure 3.4.: Schematic illustration of the S state, X state and representation unit assignment in Particle Filter and Marginal Filter. The PF uses several representation units to track the same X state, whereas in the MF only one particle is required to represent one X state. As illustrated, this results in more X states (and S states) being tracked by the same number of representation units in the different approximate algorithms. This illustration was previously published in [172].

Marginal filtering in latent infinite LTS The MF translates the ideas of the D-Condensation filter and the marginal PF to the categorical state space provided by the LTS. For this reason an explicit representation of the LTS state in the form of X states is utilised. In contrast to the D-Condensation filter, which is developed to handle durative actions in comparatively small state spaces, the marginal filter employs the concept of marginalisation to restrict the number of LTS states to be tracked. Additionally, due to the causal modelling approach, the branching factor b of the LTS is much more limited as in the D-Condensation filter (2^k at most). Starting with the initial state distribution, for each state the successor states are computed. The number of successor states is thereby restricted by bN, which essentially describes the number of actions that can be executed in a particular LTS state. Due to incremental exploration of the state space, the marginal filter has to maintain only a limited sub-set of the potentially infinite state space. This allows the marginal filter to apply the concepts of the D-Condensation filter to infinite LTSs.

Belief state representation The MF uses a set of X states to approximate the marginal filtering density by maintaining the density $\hat{p}(X_t \mid y_{1:t})$ with finite support $supp(\hat{p})$. Correspondingly, $\hat{p}$ can be represented by a set of pairs of X states $x^{(i)}$ and weights $\omega^{(i)}$. By ensuring identical X states to be merged, the value of $\hat{p}(x_t)$ is computed by summing over all trajectories that arrive in state x at time t. Figure 3.4 illustrates the difference in tracking X states and S states by the marginal and the PF. As it can be seen from the figure, the number of X states that can be tracked is higher for the MF even though the number of particles does not differ.

Inference in the marginal filter Similar to the PF, inference is done by executing two steps, prediction and update. Algorithm 2 gives a detailed description of forward filtering in the MF. During prediction, for each X state $x_t^{(i)}$, every applicable action is applied and the successor state is computed.

$$\hat{p}(X_{t+1} \mid y_{1:t}) = \sum_{x_t \in supp(\hat{p}_t)} p(X_{t+1} \mid x_t)\hat{p}(X_t \mid y_{1:t}) \tag{3.11}$$

This step typically leads to a higher number of X states. Due to the restricted branching factor, only a finite set of states is reachable in each time-step, as for each state only a finite set of successor states will be computed. The restriction of the branching factor is a direct result of the LTS generated from the causal model, which, as discussed, leads to sparse transition matrices (see Section 2.1.5). In the correction phase, for each X state $x_t^{(i)}$ the observation model is used to compute the corrected posterior distribution.

$$\hat{p}(X_{t+1} \mid y_{1:t+1}) = p(y_{t+1} \mid x_{t+1})\tilde{p}(X_{t+1} \mid y_{1:t}) \tag{3.12}$$

The support of the posterior $supp(\tilde{p}(X_{t+1} \mid y_{1:t+1}))$ is still finite. This allows the computation until this step to be exact, without a need for approximation. The sole approximation is then to restrict the number of weighted X states to be tracked. This is done by employing beam search and selecting the N most probable X states from $\tilde{p}(X_{t+1} \mid y_{1:t+1})$ to build $\hat{p}(X_{t+1} \mid y_{1:t+1})$. Afterwards the set has to be normalised to ensure $\hat{p}$ to sum to one. A more sophisticated method to pruning is discussed by Nyolt and Kirste [171].

Complexity of the marginal filter The complexity of the MF is $\mathcal{O}(N\,T)$, for one filtering step $\mathcal{O}(N)$. If N is fixed, the complexity of one filtering step reduces to $\mathcal{O}(1)$, differing only in the constant factor to the complexity of the PF. As for the pruning step, an absolute order of the states by weight is not necessary, the application of sorting can be omitted here. Thus, the pruning step can be done by selecting the N states with the largest weights, which can be accomplished in linear time [27].

In addition to efficient forward filtering, the MF allows both, estimating the smoothing density and the maximum a-posteriori path to be computed in $\mathcal{O}(N\,T)$. This is basically done by first considering only actions whose resulting state is part of the filtering density and by caching the probability of selecting an action a_t in a specific state s_t. Caching action selection probabilities based on the S state allows reusing them in all associated X states. This effectively reduces the complexity of selecting a new action from $\mathcal{O}(n_A^S)$ to $\mathcal{O}(1)$ per state, where n_A^S is the number of applicable actions in state S. The number of applicable actions per state, namely the branching factor, is constant and can be assumed to be much smaller than the overall number of actions. In addition, caching prevents the action selection function γ from being recomputed for different X states that are based on same the S state.

Marginal filtering for categorical state spaces To summarise, the MF allows for efficient approximate inference in categorical state spaces. It overcomes the drawbacks of the PF by a weight-based representation, where the representation of duplicated X states is prevented. By replacing the sampling-based prediction step of the PF by fully expanding the successive time-step, an exact prediction is allowed. By ensuring duplicate X states to be merged, an efficient representation of the belief state is enabled. Finally, by substituting the resampling step of the PF by pruning based on beam search, it is guaranteed that the most likely hypotheses about the state space are further maintained. Thus, the MF provides an improved method for inference in categorical state space with sparse transition matrices.

3.3. The Computational Causal Behaviour Model Toolbox

In order to investigate the capabilities of CCBM with respect to recognition performance, the CCBM toolbox provides an implementation of the concepts introduced thus far. It provides a collection of tools related to CCBM based inference. The purpose of the CCBM toolbox is to provide a common infrastructure to investigate different aspects of the question **IQ**. To this end, the CCBM toolbox provides a modelling language and an implementation of the inference algorithms (see Section 3.2). In the following, a brief overview of the toolbox is provided.

3.3.1. Modelling Language

Objective One central aspect of CSSMs is the computational action language, which allows to describe the system dynamics by computational means. This allows to specify a latent infinite LTS. The concept of CSSM, in general, raises no restrictions to the modelling language. Any computable form can be used to describe the actions and their interactions with the state. However, as discussed in Section 2.1.4, the application of DSLs allow for reusability. As this is one of the main objectives of CCBM, the use of a modelling language that satisfies the needs of the investigations, conducted in this thesis, is considered rather than using a general purpose programming language such as C++ [225] or LISP [207]. Additional justification of this decision is provided by Spinellis [216], who provides a list of advantages of DSLs over general purpose languages when it comes to special modelling purposes. These are for instance, "Direct involvement of the domain expert" – DSLs allow domain experts to be easily incorporated – and "expressiveness" – DSLs *"can be designed to provide the exact formalisms suitable for that domain"*,(Spinellis [216]).

General framework In order to address the needs of the investigations in this thesis, a list of requirements for the modelling language was collected. The analysis was done based on the requirements for the inference system (Section 1.2), the statistical model (Section 3.1) and the requirements analysis on human behaviour modelling by Yordanova [255]. The following requirements were considered:

L1 The modelling language has to support the paradigm of model-based specification. In particular, the modelling language has to allow the specification of the environment state by means of state features resembling properties of the application scenario under investigation. Furthermore, actions have to be modelled by their relation to the environment state. Preconditions are used to limit the number of possible states where actions can be executed in and effects describe how the execution of an action changes the state.
(This requirement is a direct result of the discussion in Section 2.1.4.)

L2 The modelling language has to support the specification of action templates. By providing parameters, an action template describes a set of actions.
(This requirement results from considerations with respect to reusability, as action templates can be reused with different variables that match the definition of the parameters.)

L3 The modelling language has to provide a certain support for multiple agents. In particular, different execution threads for different agents have to be modelled. Furthermore, the execution thread for actions has to be specified.
(This is necessary to investigate the multi-agent capabilities of CCBM with respect to reusability (R5.3) and recognition performance.)

L4 The modelling language has to support the specification of probabilistic action durations. Depending on the parameters of the action template (see requirement L2), the modelling formalism has to allow the specification of the actions' durations.
(This requirement is a result of discussions about action durations in Section 3.1.)

L5 The modelling language has to allow to bind an observation model to the behaviour model, to support both, action and state observation models. For each action template (see L2) and for each state feature of the environment state, support for specifying the interaction of observation and behaviour model has to be provided.
(This requirement is a result of the discussion about the statistical model in Section 3.1.)

L6 The modelling language should allow to separate the general parts of the model from the parts that are specific to the application scenario. This allows the general parts to be reused by adjusting application specific parameters.
(This requirement is a result from the considerations with respect to R5.1.)

Design considerations Yordanova [255], who analysed the requirements for human behaviour models, gives an overview of alternatives and highlights PDDL as favourite choice to model human behaviour. Thus, similar to other modelling languages for CSSMs used in the literature [189], the CCBM modelling language is based on PDDL [156]. In addition to the specification of human behaviour, the use of PDDL provides further advantages, as it allows for model checking [173] and automatic state space analysis without requiring full state space exploration. The latter has been used in the planning domain for e.g. the automatic extraction of landmarks [193] or reachability analysis [227].

In detail, PDDL satisfies the following requirements directly:

L1: PDDL employs the model-based specification mechanism. State features are described by means of predicates and fluents. The main difference between both is that predicates use a boolean value domain, whereas the value domain of fluents can be customised. Actions are described by preconditions and effects with respect to the state features.

L2: Actions in PDDL are described by use of action schemata with parameters. For each parameter the type is provided. During grounding for each combination of possible instantiations of the parameters, a grounded action is created. Thus, PDDL allows the specification of sets of actions by use of action schemata.

L6: A PDDL planning task consists of two parts: the domain and the problem description. The domain description can be considered as static within the same application domain. The problem specification contains a list of involved objects, an initial state and a goal formula and can thus be considered as dynamic within the same application domain. This allows domain specifications to be reused with different problem descriptions.

With respect to the choice of the modelling language, the terms “action schema”, “grounded action” and “action class” are defined as follows:

Definition 6 (action schema) *An action schema is a template with parameters. An action schema describes a set of actions by use of these parameters.*

Definition 7 (grounded action) *A grounded action is a specific instance of an action schema. Instead of parameters, grounded actions refer to specific objects.*

Within the scope of this thesis the term “action” refers to a “grounded action”.

Definition 8 (action class) *An action class describes the type of action. Typically, an action class refers to an individual action schema, but this is not necessarily the case.*

Within the scope of this thesis the term “activity” refers to an “action class”.

Realisation In order to satisfy the entire list of requirements, the CCBM modelling language introduces additional features. The slots `:agent`, `:duration`, and `:observation` are introduced to satisfy the requirements that are not directly satisfied by PDDL. Example 3.1 provides an action specification. In detail, the following requirements are addressed:

L3: The CCBM modelling language allows to set the execution slot of an action by use of the slot `:agent`. This can either be done by use of a constant or a parameter. For each object, regardless of the type, that is at least once set by any action schema, a different execution thread is created. An execution thread reflects the multi-agent concepts that are discussed in Section 3.1. For the action take in Example 3.1, each grounded action will be executed in the corresponding execution thread of the individual occupancy of the parameter `?who`. If the variable `?who` is, for instance, instantiated with three different values, three execution threads are created.

L4: To specify a probabilistic action duration for an action template, the slot `:duration` is used. The value of this slot does not specify an action duration directly, but rather

Example 3.1: The action take

The action template take as defined with the CCBM modelling language. The action take has three parameters: a person that takes something, an object that is taken, and the original location of the object. The duration of the action is given by the external function takeDuration. The action is executed in the thread of the person who takes the object. The action take can be applied whenever the number of objects that are already taken is below 2 and the both the person and the object are at the original location of the object. After the action take was executed, the number of objects is increased by 1 and the location of the object has changed to the person – the person is carrying the object. The observation probability this action schema is computed by the function takeObservation.

```
(:action take
  :parameters (?who - person ?what - object ?from - location)
  :duration (takeDuration)
  :agent ?who
  :precondition (and
                  (< (objects-taken ?who) 2)
                  (= (is-at ?what) ?from)
                  (= (is-at ?who) ?from))
  :effect (and
            (increase objects-taken)
            (assign (is-at ?what) ?who))
  :observation (takeObservation)
)
```

refers to an external function. The rationale here is, that the construction of duration models including the selection of appropriate PDFs is part of active research. To this end, the reference to an external function allows the duration model to be adjusted without changes to the behaviour model. For the action take in Example 3.1, the duration of all grounded actions that are generated from the action schema is provided by the external function `takeDuration`.

L5: The requirement to bind observation models is addressed by the slot `:observation`. As for the duration model, the value of the slot refers to an external function. Likewise, the rationale is that the observation models can thus be exchanged without changes to the model. In Example 3.1 the observation model for all grounded actions of this action schema is provided by the external function `takeObservation`. Additionally, to support state observations an `:observation` clause is introduced to the domain.

Example A.1 provides a domain and a corresponding problem description.

Compilation of behaviour models When it comes to inference based on models of human behaviour, CCBM follows the "source-to-source transformation" creational pattern [216] – the CCBM modelling language is translated to C++ [225]. The result of the compilation is a C++ representation of the environment state and all grounded actions. Example 3.2 provides the results of the compilation of the action from Example 3.1. The generated C++ code is then compiled together with an implementation of the inference algorithms. The rationale here is that CCBM could take advantage of the execution speed due to the optimised compilation toolchain, when it comes to inference. This is also an advantage over the usage of interpretation, as compilation time can be neglected in favour to inference time. The reason for this is that an inference task will be executed several times without changes to the domain or the problem specification, as usually action sequences have to be reconstructed from several observation sequences (for instance Experiment X1 executes 20 inference tasks with the same model). This

Example 3.2: Compiled representation of the environment state and the action.

For each grounded state feature the state contains a variable. The state is a result of compiling the action from Example 3.1. The preconditions and effects of the action0 (take emilia spoon drawer) refer to the compiled state representation in Example 3.2.

```
struct StateRec {
  unsigned int F0:2; // objects_taken emilia
  unsigned int F1:3; // is-at plate
  unsigned int F2:3; // is-at spoon
  unsigned int F3:3; // is-at emilia
        StateRec() {bzero(this,sizeof(StateRec));}
};
```

```
//(take emilia spoon drawer)
bool action0(StatePtr x, StatePtr x1) {
        if(x->F0 < 2 && x->F1==1 && x->F3==1) {
                *x1 = *x;
                x1->F1 = 0;
                x1->F0 = (x1->F0)+1;
                return true;
        } else {
                return false;
        }
}
```

is in contrast to traditional planning where the planner is executed once for each combination of domain and problem, which are interpreted by the planner to save time of the overall process.

3.3.2. CCBM Inference Tools

Objective of the CCBM toolbox is to allow investigations of the concept of CCBM. This includes modelling aspects as well as the inference algorithms that were introduced in Section 3.2. To this end, for both inference methods a corresponding inference tool is provided.

Particle Filter: The PF inference tool provides an implementation of Algorithm 1. Thus, given an observation sequence, the PF estimates the most likely sequence of X states by maintaining a belief by sets of particles.

Marginal Filter: The MF provides an implementation of Algorithm 2. Like the PF, the MF takes a sequence of observations and estimates the sequence of X states.

The need for these tools was also described in the workflow for developing causal models for AR, introduced by Yordanova and Kirste [258]. The inference tools are required to estimate the activity during the evaluation phase.

Another tool that was suggested by Yordanova and Kirste [258] is the plan validator. It is used in the validation phase, as it allows to check wether the behaviour model allows to represent the action sequence derived from the annotation. The need for a plan validation tool is also described in Chapter 4 to ensure annotation to be causally correct.

As discussed in Section 3.2, the inference complexity of both, PF and MF depends on the number of particles. The complexity of both the PF and the MF is $\mathcal{O}(N\,T)$, where N is the number of particles and T the length of the observation sequence. Efficient inference means to handle the states in an appropriate way, as similar, for instance, in each time-step, equal states have to be merged. Bonet and Geffner [29] suggest the usage of hash tables, as they allow to access elements in constant time. This is also implemented in the CCBM inference tools.

4

Causally Correct Annotation

Synopsis: *This chapter discusses the need for annotation sequences and the problem of causal inconsistencies in the annotation. A model-based semantic annotation by means of LTSs is proposed to overcome this problem. Finally, a workflow that allows to produce causally correct annotation is introduced.*

Chapter Sources: *Parts of this Chapter have been previously published in the following publication(s):*

- *Towards Causally Correct Annotation for Activity Recognition [92]*
- *Computational State Space Models for Activity and Intention Recognition. A Feasibility Study [133]*

The annotation of sensor datasets describing human behaviour is an important part of the AR and the PR process [26, 40]. It provides a target label for each observation in the cases where supervised learning is applied. It also serves as a ground truth for evaluating the performance of the activity or plan estimation procedure by comparing the estimated values with the annotated values. As this thesis targets at activity and context recognition, sufficient annotation labels that provide the corresponding ground truth are required. Furthermore, as CCBM exploits causal dependencies the annotation has to be causally correct. Here, a model-based (see Section 2.1.4) approach to semantic annotation of human behaviour is proposed, like for the modelling of human behaviour.

4.1. Annotation of Human Behaviour

Annotation of activities In the context of human behaviour recognition, three different types of annotation exist. In the first type of annotation, textual descriptions (or labels) are assigned

Source	sensor modality	sensors	example labels
[18]	motion sensing	accelerometers, video	walk, stand, sit
[223]	object sensing	RFID	vacuuming, ironing
[54]	smart environment	switches, device usage, location	phone call, wash hands, cook
[149]	location tracking	GPS, indoor navigation	work, sleep, car
[238]	dense sensing	reed switches, pressure mats, passive infrared	leave house, showering, preparing breakfast

Table 4.1.: Examples of annotation schemes The table lists different annotation schemes, where the choice of the label set was influenced by the sensing modalities used.

to each executed action [60, 64, 86, 238]. The objective is to manually assign a label l_i to each time-step of a time series by analysing a separately recorded video log. The resulting sequence of labels is usually called *ground truth*, as it provides a symbolic representation of the true sequence of activities. However, for the finite set $\mathcal{L} = \{l_1 \dots l_n\}$ of labels there is usually no further information besides the equality relation. This type of annotation is therefore known as *annotation of activities* as it simply provides an activity label without any further meaning [239]. Annotations such as "take-baking_pan" provide a textual description of the executed task that, however, do not contain an underlying semantic structure, although imitating a semantic meaning. There is usually no formal set of constraints that restrict the structure of the label sequences. Typically, nothing prevents an annotator from producing sequences like "put fork to drawer" → "close drawer" → "take knife from drawer". Examples of such textual labels can be found in the data provided by [192, 218, 60]. This is also the most common type of annotation of human behaviour, partially because even the assignment of non-semantic labels to the data is a difficult, time consuming and error prone task [238, 40].

In addition to absent semantic meaning, the label set and its granularity is often decided upon the expected capabilities of the sensor infrastructure used in the experiment [64, 238, 223]. For instance, temperature sensors in the showers, pressure sensors in the bed, and reed switches in the door strongly correlate with annotated activities showering, sleeping, opening/closing. Table 4.1 lists examples from the label set including the sensor modalities from the literature. While this approach seems convenient at first, it is inappropriate for two reasons: (1.) it exaggerates the discriminative capabilities of the model as the target labels are determined by the sensors capabilities and (2.) it effectively prevents the annotation scheme from being reused independently from type of sensors.

Plan annotation The second type of annotation is the *plan annotation*. Blaylock and Allen [26] divide it into goal labelling and plan labelling. Goal labelling is the annotation of each plan with a label of the goal that is achieved [4, 24]. In contrast, plan labelling provides annotations not only of the goal, but also of the actions constituting the plan, and of any subgoals occurring in the plan [19]. The latter is, however, a time consuming and error prone process [26]. The only attempts of such plan annotation are done when executing tasks on a computer (e.g. executing plans in an email program [19] or interacting with the terminal [24]). This is also reflected in activity and plan recognition approaches [189, 95] that use only simulated observations, and thus simulated annotation, to recognise the human actions and goals.

Semantic annotation The third type of annotation of human behaviour is the *semantic annotation* [123]. The term originates from the field of the semantic web. Semantic annotation is described as the process and the resulting annotation or metadata consisting of aligning a resource or a part of it with a description of some of its properties and characteristics with respect to a formal conceptual model or ontology [10]. The concept is later adopted in the field of human behaviour annotation, where it describes the annotating of human behaviour with labels that have an underlying semantic structure represented in the form of concepts, properties, and relations between these concepts [49, 201].

Model-based semantic annotation As argued in Section 2.1.4, the model-based representation provides a model of the system's state in terms of collection of state variables. The individual operations are defined in terms of their preconditions and effects on the state of the model. There have been no attempts to represent the semantic structure of human behaviour annotation in the form of model-based representation. Here, an approach to semantic annotation of human behaviour is used that employs a model-based representation of underlying semantic structure.

By providing an LTS, the set of causally correct annotation sequences is given by the sequences that can be generated by the LTS. In the following the LTS that represents the possible annotation sequences in named annotation LTS (aLTS), as it establishes a labelled transition system to ensure causal annotation correctness. More formally, consider an alphabet of labels $\mathcal{L}$ and an observation sequence of length n. Typically, the output of the annotation process is an annotation sequence $\alpha_{1:n} \in \mathcal{L}^n$. As already argued in Section 2.1.1, this annotation sequence is unrestricted in terms of possible sequences. Even impossible sequences can occur. Objective of the annotation process, proposed here, is to find a set $\mathcal{L}^* \subset \mathcal{L}^n$ such that $\alpha_{1:n} \in \mathcal{L}^*$ represent "correct" sequences. Here, the term correct sequences describe all sequences that can be generated by traversing the aLTS when starting from the initial state.

This representation allows to provide not only a semantic meaning to the labels, but also to produce plan labels and to reason about the plan's causal correctness. Furthermore, it gives the state of the world corresponding to each label and allows to track how it changes during the plan execution. This allows to generate a context annotation by "executing" the annotation sequence with respect to the aLTS. The following section describes the workflow for creating such model-based semantic annotation. Examples are taken from the Carnegie Mellon University Multimodel Activity (CMUMMAC) database [59].

4.2. Model-based Semantic Annotation for Human Behaviour

Typical AR experiments, such as the CMUMMAC [59], are the targeted group of datasets. Participants are instructed to fulfil a task such as food preparation. To ensure comparability of different repetitions, identical experimental setup is chosen for each trial. As a result, the action sequence executed by the participants can be regarded as a plan, leading from the same initial state (as chosen by the experimenter) to a set of goal states (given in the experiment instruction). Thus, an aLTS can be employed to describe the possible actions sequences.

The proposed annotation process allows to label a sequence of actions leading from an initial state to a set of goal states. If the action sequence, actually executed by the human protagonist, consists of two interleaved plans with two distinct goals (e.g. having prepared breakfast and having finished the morning routine), the goal can be defined as a conjunction of both.

In the following, the workflow steps to manually create a model-based semantic annotation are described. A graphic of the annotation process workflow is given in Figure 4.1.

Step one: Action and entity dictionary definition In the first step, a dictionary of action classes (e.g. the action class of "take-fork-drawer" is "take") and entities is created by a manual analysis of the video log. The action classes are represented by their name and a description of the action class that distinguishes it from the remaining action classes. The dictionary also contains the set of all entities observed during the experiment. All physical objects being

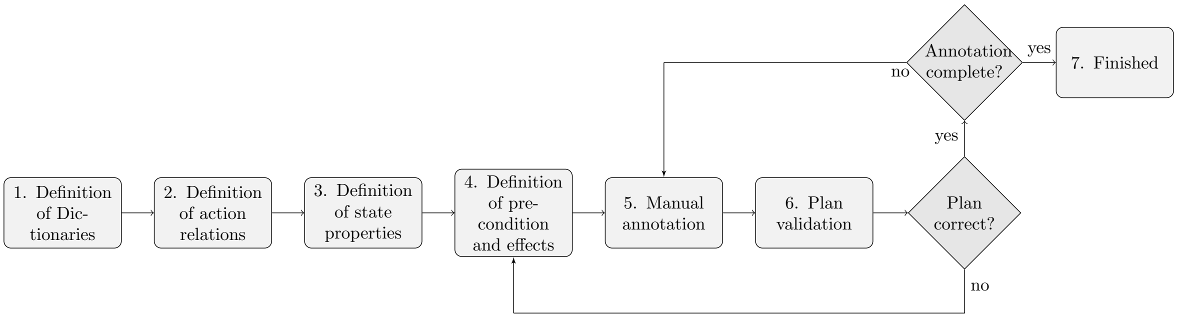

Figure 4.1.: The proposed workflow for semantic annotations of human behaviour.

	action class
a_1	take
a_2	put
a_3	walk
	...
a_n	stir

	entity
e_1	knife
e_2	drawer
e_3	counter
	...
e_m	pepper

Table 4.2.: Example of the results of the first annotation step. The table exemplifies dictionaries of actions (l.) and entities (r.) that were result of the first step of the proposed annotation process.

manipulated by the human protagonist (including the protagonist(s)) are considered as entities. The dictionary is manually created by domain experts analysing the video log. The results of the dictionary definition are the set of action classes and the set of entities manipulated during action execution (see Table 4.2).

In addition to the name, a definition is provided for each action class. The definition of action take for example is given in Example 4.1.

Step two: Definition of action relations In the second step, the action relations are defined. For each action, the number and role of involved objects is defined. In case of take, for example, an object and a location, where the object is taken from, are defined. In addition, for each object, possible roles have to be identified. A pot, for example, can be taken, filled, washed, and stirred. The result of this step is the finite set of labels $\mathcal{L} = \{l_1 = \tilde{a}_1^1, l_2 = \tilde{a}_1^2, \ldots, l_k = \tilde{a}_n^m\}$, where $\tilde{a}$ defines the syntax of the action relation a to be used for the annotation process (see Table 4.3). This set represents the set of labels to be used for the manual annotation process. To complete this step, a set of types is defined based on possible roles of all entities.

Step three: Definition of state properties Objective of the third step is to define the state space by means of state properties. Therefore, a set of state properties is defined, each as function of a tuple of entities to an entity of the domain. The state space is then defined by each combination of possible mappings of entity tuples. Finally, the subset of mappings

> **Example 4.1:** Definition of the action *take*
>
> To grab an object. The location of the object changes from the initial location to the hand of the person. The action consists of moving the arm to the object, grabbing the object and finally moving the arm back to the body.

a_1	***take* (*what*:takeable, *from*:location)**
a_1^1	take (knife, drawer)
a_1^2	take (knife, board)
	...
a_2	***put* (*what*:takeable, *to*:location)**
a_2^1	put (knife, drawer)
a_2^2	put (knife, board)
	...

Table 4.3.: Example of the results of the second annotation step. The table lists the type signature and each possible instantiation for the set of actions identified in the previous step.

f_1	**is-at (what: takeable) → location**	
f_1^1	is-at (knife) ↦ drawer	*
f_1^2	is-at (knife) ↦ board	
	...	
f_2	**objects_taken () → number**	
f_2^1	objects_taken () ↦ 0	*
f_2^2	objects_taken () ↦ 1	
	...	

Table 4.4.: Example of the results of the third step of the annotation process. The table shows a list of functions with type signatures and their instantiations. A $*$ marks functions holding in the initial state.

that holds in the initial state (start of the experiment) is marked (see Table 4.4). The result of the third step is a list of functions that define state properties. Moreover, for each entity defined in the first step, a list of state property functions and action relations is available. The specification of types from the previous step is extended by the entities' roles with respect to the state properties. Based on the type of each entity, defined by their corresponding state property functions and action relations, a type hierarchy can be created.

Step four: Definition of preconditions and effects Objective of the fourth step is to define the semantics of the actions. Using the type signature defined in the second step, action schemata are defined in terms of preconditions and effects. As illustrated above, participants' action sequences are regarded as plans. Actions are described in a PDDL-like[1] syntax, known from the domain of automated planning and scheduling. The preconditions and effects for the single action schemata are formed by domain experts. A *take* action for example requires an object to be taken, the maximal number of objects not to exceed, and, in case the location is a container that can be opened and closed, it has to be open. This requires either the plan sequence to contain an action which opens the container, or the container to be open in the initial state. Effects of the take action are that the location of the object is changed from the original location to the hand, the number of taken objects is increased, and if the object to be taken is dirty the hands become dirty too (see Figure 4.2).

Step five: Manual annotation Once the dictionary of labels is defined, the manual annotation can be performed. Here, the ELAN annotation tool [249] is used for manual annotation. An annotator has to assign labels from the defined label set to each time-step of the video sequence. The ELAN annotation tool allows to synchronise several video files and to show them in parallel.

[1]In the scope of this thesis the CCBM modelling language is chosen as action language.

```
(:action take
 :parameters (?what - takeable ?from - loc)
 :precondition (and
  (= (is-at ?what) ?from)
  (not (= ?from hands))
  ...
 :effect (and
  (assign (is-at ?what) hands)
  (increase (objects_taken) 1)
  (when
   (not (is-clean ?what))
   (not (is-clean hands)))))
```

Figure 4.2.: Specification of the annotation LTS action take. Extract of the aLTS action schema for the take action encodes preconditions and effects in the CCBM modelling language. The entities (i.e. hands) are result of the first step, the list of parameters (i.e. (?what - takeable ?from - loc)) of the second step, and the available state features (i.e. is-at, is-clean) result of the third step.

To prepare the annotation sequence for the next step, plain text is extracted from the ELAN file format. Figure 4.3 shows the ELAN tool during the annotation of a cooking task.

Step six: Plan validation Since the label sequence produced in the previous step consists of plan operators, the complete sequence can be interpreted as plan, leading from an initial to a goal state. Objective of the sixth step is to check the causal validity of the label sequence with respect to the planning domain created in the previous step. A plan validator (such as VAL [102] or the CCBM validator (see Section 3.3.2)) can be used for this task. If the label sequence does

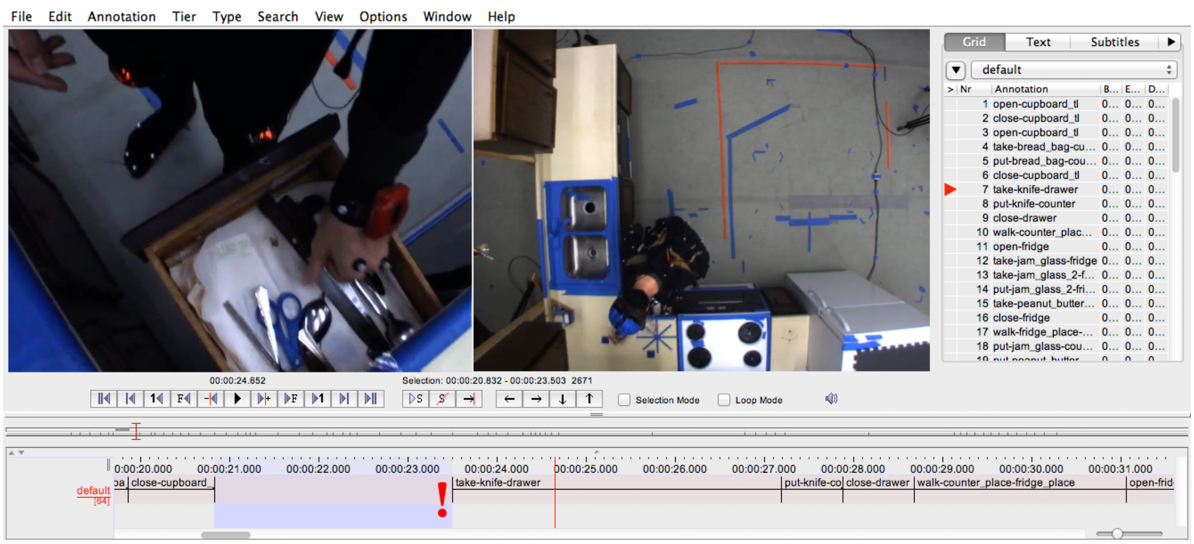

Figure 4.3.: Screenshot of the annotation procedure using ELAN. The ELAN tool used to manually create an annotation sequence from the video log of the CMUMMAC [59]. Here the action “take-knife-drawer” is being annotated. It can be seen that the annotation of the action “open-drawer” is missing. Thus, the action “take-knife-drawer” is marked as causally incorrect, as it is not possible to take an object from a closed drawer. The red explanation mark tags this problem while validating the plan. The annotator is informed about the causality issue and advised to go back to step four of the annotation process.

not fulfil the causal constraints of the planning domain, two possible reasons exist: (1.) the planning domain does not correctly reproduce the constraints of the experimental setting or (2.) the label sequence is incorrect. In case of an incorrect label sequence, step five (manual annotation) has to be repeated to correct the detected problems. In case of an incorrect domain, either the preconditions, defined in step four, have to be relaxed or the effects have to be revised. Figure 4.3 provides an example of a failed annotation sequence. The action “take-knife-drawer” cannot be executed, since the action “open-drawer” in not annotated, yet. Consequently, the plan validation fails, as the preconditions for the taking the knife from the drawer are not satisfied.

Summary The proposed annotation process has three results, (1.) the sequence of labels, (2.) the semantic structure of the labels, and (3.) an aLTS describing the causal relation of the labels. The application of CCBM requires the causal correctness of the annotation and thus a formal model of causal behaviour. Causality not available in the aLTS cannot be reliably exploited in the inference model. As consequence the LTS created for inference (inference LTS (iLTS)) has to be a refinement of the aLTS. Otherwise, the iLTS will not be able to differentiate between certain states and actions that are discernible in the aLTS.

To summarise, in order to employ the causal structure of human behaviour during inference, already the annotation of the test and training data has to be causally correct. Furthermore, reasoning about actions, goals and contextual information requires the annotation to contain such knowledge. The proposed process allows the creation of a causally correct semantic annotation of human behaviour. The process was used to provide such annotation for the second (Section 6.2) and the third experiment (Section 6.3).

5 Methods

"Prediction is very difficult, especially about the future." – Niels Bohr

SYNOPSIS: *This chapter's aim is to provide an overview of the methods used for the experimental investigations on the capabilities of CCBM. After collecting research questions by analysing the question **IQ** and the requirements, a brief discussion about the usage of empirical data is provided. Then the procedure for executing and evaluating the experiments is introduced. Typical evaluation measures are discussed briefly and finally, appropriate measures are selected.*

CHAPTER SOURCES: *Parts of this Chapter have been previously published in the following publication(s):*
- *Computational State Space Models for Activity and Intention Recognition. A Feasibility Study [133]*
- *Plan Synthesis for Probabilistic Activity Recognition [131]*
- *Where are My Colleagues and Why? Tracking Multiple Persons in Indoor Environments [132]*

The previous chapters introduced the general concept of CSSM and presented CCBM as one instance to reconstruct human behaviour from sensor data. CCBM was developed to provide a reusable mechanism to infer the activity, the plan, and the context simultaneously. The topic of this chapter is a discussion of the methodologies that are used to demonstrate the effectiveness of CCBM with respect to question **IQ**. To this end, in the first step a list of research questions is derived from question **IQ** and the requirements, stated in Section 1.2.2. The discussion then addresses the need for empirical data. Afterwards, the general experimental design, which was the basis for all experiments, is introduced. Finally, measures for performance estimation are discussed.

5.1. Research Questions

In Section 1.2.2, a list of requirements for the inference system was derived. This section now aims at translating these requirements into research questions that can be answered by conducting experiments and analysing the results. To summarise, the following five requirements were collected: (1.) Uncertainty – handling sensor observations (R3), (2.) Online – allowing online inference (R2), (3.) Reusability – allowing the behaviour model to be reused (R5), (4.) Latent infinity – handling possibly infinite state spaces (R4), and (5.) Plan – recognise the plan and the goal of the human protagonist (R1).

Chapter 3 illustrated that these requirements are addressed by considering ideas from the literature that partially satisfy them. A model-based description allowing for reusability, similar to PDDL as used by Ramírez and Geffner [188] is used to generate a possibly infinite LTS graph. A statistical model, similar to the one described by Liao et al. [149], was introduced for inference in this graph to cope with uncertainties inherent to sensor data. Finally, the framework of Bayesian forward filtering was applied to allow online inference of the plan of the human protagonist. Consequently, CCBM satisfies the stated requirements and allows to answer the question **IQ**. Besides to satisfying the stated requirements, an inference system for CSSMs has to be able to achieve recognition rates at the same level as standard methods.

By using a computational action language, as described in Section 3.3.1, it is very easy to produce models with very large state spaces. The reason for this is the ability for generalisation of the computational representation. A model considers all action sequences that achieve the same objective rather than concentrating on an explicit enumeration of action sequences (see Section 2.1.4 for a detailed discussion). However, from the viewpoint of statistical inference, a large state space is first of all not an asset but a liability. A larger state space, allowing more variance in the action sequences, might achieve weaker recognition performance than a smaller, potentially more biased state space. This can be considered as an instance of the bias-variance trade-off [88, p.158].

Objective of the experiments is to provide evidence that, albeit providing a rich state space, CCBM allows to achieve recognition rates at the same level as baseline classifiers. Thereby it is shown that CCBM satisfies the requirement for uncertainty by using noisy and ambiguous sensor data produced by observing human protagonists. As discussed in Section 1.2.2, three aspects of reusability R5 are targeted by CCBM. R5.1 is addressed by developing a model for a specific scenario of an application domain and applying the same model to a different scenario. At the same time the model has to achieve good recognition rates[1]. Exploiting action and state observation (Z and W component of the statistical model in Figure 3.1) while recognising activities based on the same causal behaviour model allows to demonstrate the satisfaction of R5.2. Finally, to provide evidence for the satisfaction of R5.3, the multi-agent capabilities have to be investigated.

To a address the above issues, they were rephrased as research questions. More precise, the following research questions were raised:

RQ1 Is it possible to simultaneously estimate activities, context information, and the goal from location data with CCBM models of similar complexity as related approaches with recognition rates at the same level as baseline classifiers?
(addressed requirements: Plan, Uncertainty)

RQ2 Is it possible to reconstruct the action sequences of multiple cooperative agents with a CCBM model with recognition rates at the same level as baseline classifiers?
(addressed requirements: Plan, Reusability)

[1]Good recognition rates means that the recognition performance should not be significantly worse than for a trained baseline classifier.

Requirement	RQ	Experiment X1	Experiment X2	Experiment X3
Uncertainty	R1	■	□	□
	R4	□	■	□
	R5	□	■	□
	R6	□	■	□
	R7	□	■	□
Online	-	■	■	■
Reusability	R2	■	□	□
	R3	■	□	□
	R7	□	■	□
	R8	□	□	■
Latent Infinity	R4	□	■	□
	R9	□	□	■
Plan	R1	■	□	□
	R2	■	□	□
	R4	□	■	□
	R5	□	■	□

Table 5.1.: Overview of the experiments and the research questions. For each requirement the list of research questions, which are targeted are listed. For each experiment the targeted research questions are marked with ■.

RQ3 Is it possible to reuse a causal behaviour model that was created for one specific application domain in a different scenario within that application domain?
(addressed requirement: Reusability)

RQ4 Is it possible to achieve successful state estimation using CCBM models of everyday activities with large state spaces (containing hundreds of millions of states)?
(addressed requirements: Plan, Uncertainty, Latent Infinity)

RQ5 Which modelling factors (duration model, action selection heuristics, inference algorithm, etc.) are relevant for achieving a good performance in CCBM-based inference?
(addressed requirements: Plan, Uncertainty)

RQ6 Is it possible to achieve good recognition rates for fine grained AR from wearable sensors by means of CCBM?
(addressed requirement: Uncertainty)

RQ7 Is it possible to exploit the different components of the observation model of the statistical model without further changes to the causal behaviour model used to generate it?
(addressed requirements: Uncertainty, Reusability)

RQ8 Is it possible to create a causal behaviour model that can be reused for different numbers of agents?
(addressed requirement: Reusability)

RQ9 How is the recognition performance of a CCBM model influenced by an increased state space resulting from an increased number of agents?
(addressed requirement: Latent Infinity)

To answer these questions, a series of experiments, each targeting multiple questions, was conducted. Table 5.1 gives an overview of the experiments, the focussed research questions and the addressed requirement(s).

The remainder of this chapter addresses the methods being used in order to assess the performance of the recognition.

5.2. Empirical Data

Reasons for empirical data From the discussion of related work (Section 2.2), it can be seen that most studies make use of simulated data and/or direct observation. Especially in the domain of PR, only few researchers use sensor data from empirical studies (e.g. Bui et al. [37], Liao et al. [149], Sadilek and Kautz [202]). Some researchers even state that the use of simulated data is favoured due to issues of behaviour annotation [26]. While the use of simulated data has several advantages (e.g. reproducibility, almost unlimited data), here we focus on sensor data from empirical studies.

The use of empirical trial data was chosen in favour to simulated data for the following reasons:

- The use of simulated data (presumably from the same model that is used for inference) will exaggerate accuracy and overestimate the effect of action selection heuristics. Additionally, it will guide research on action selection heuristics in the wrong direction, as the simulated action sequence will always fit the heuristic's assumptions, independent from the actual heuristic. Action sequences, generated from human behaviour require the heuristic to reproduce human action selection.
- Evaluating model behaviour with respect to sensor data obtainable in real settings requires to have such data available for use as observations.
- Albeit not necessary for the model development, samples of real-world behaviour provide a good starting point for model construction in order to develop models of realistic structural complexity with respect to everyday behaviour.

Sufficient sample size Regarding the sample size of the empirical data collection, it has to be noted that the purpose of the experimental validation is a comparison of CCBM to standard methods in realistic scenarios. Therefore, relative comparisons of the methods' recognition results are required, rather then absolute performance rates. To this end, a convenience sample of volunteers was considered sufficient to demonstrate the capabilities of CCBM in comparison to baseline classifiers.

Empirical data for model comparison The main purpose of the empirical samples is to provide data for model comparison. With respect to designing the causal model of the CCBM, a large dataset is not required for parameter training. Furthermore, the causal model is not subject to the $\sqrt{n}$ law regarding the standard error of a parameter estimate. A single example at the symbolic level is sufficient to infer a causal link. In fact, this is one of the main advantages of CSSM-like approaches as they allow even rare action sequences to be modelled without the need for training data.

Evaluation with respect to empirical data With respect to the evaluation, typically, leave-on-out cross validation is applied to prevent models from overfitting. However, for creating the causal model of the CCBM, leave-one-out cross validation is infeasible, as it would require the number of model engineers with identical qualification being equal to the sample size. Thus, the causal model is developed based on the entire dataset. In order to not disadvantage the baseline models relative to the CCBM, they were also created by use of the complete data. Thus, both, the CCBM and the baseline performance can be expected to be exaggerated in absolute terms due to overfitting. However, as we focus on a comparison in favour of absolute performance this exaggeration is not an issue. If CCBM can be shown to achieve similar recognition rates as the baseline classifier, then this justifies an investigation of the capabilities on a larger scale. This should also include the application of leave-on-out cross validation for the model development (e.g. by learning models [263]).

5.3. Experimental Procedure

In order to answer the nine research questions RQ1 – RQ9(see Section 5.1), a series of experiments (X1 – X3) is conducted. The models, which are base of the investigations in the experiments, are constructed by use of a common development process for human behaviour models that was introduced by Yordanova and Kirste [258]. The workflow targets the development of CSSMs in particular. The process includes six phases that have to be executed subsequently but allows for iterative refinement:

YK1 **Analysis**: The problem domain is analysed, a data recording experiment is conducted and the recorded data is annotated. This step also includes a detailed analysis of the sensor data and a preparation for further processing. According to the process introduced in the previous chapter, the aLTS is created in this phase.

YK2 **Design**: A modelling solution including action durations and action selection heuristics is selected according to the needs of the application scenario. The contextual information that is of potential interest has to be identified during the second phase. Additionally, decisions upon the duration modelling and action selection have to be made.

YK3 **Implementation**: An initial human behaviour model is implemented. The duration model and the action selection heuristics are implemented according to the decisions of the previous phase. During this phase multiple duration models can be implemented and tested for their influence to the model performance.

YK4 **Validation**: The model is validated with respect to the actual user behaviour recorded during the first phase. Here, all parts of the dataset have to be considered to avoid potential overfitting. Based on the separation of single aspects of the model (e.g. the different sub-models, see Section 3.1) it is checked whether the model is able to explain the human behaviour that was recorded during the experiment.

YK5 **Evaluation**: The model is validated with respect to the discriminate power, namely the recognition performance. Typical performance metrics are selected to assess the discriminate power of the model. Furthermore, the results are compared to that of baseline classifiers and their statistical significance is proved.

YK6 **Documentation**: A documentation is created that contains all design decisions including their cause.

At each point during the process it is allowed to go back to a previous phase in order to refine modelling decisions. In the following, each step of this workflow is described in detail with respect the experiments X1 – X3.

YK1 – Analysis Objective of the first phase (YK1) is to select an experimental setting that is appropriate to investigate a research question. For this purpose, during the analysis phase, first, a subset of research questions was selected. Afterwards, an application domain was selected based this subset.

For Experiment X1, for instance, research questions RQ1, RQ2, and RQ3 were selected. The application domain "meeting of three persons" was chosen, as it allows to answer these questions. A public available dataset [121] was selected, as it allows an investigation of most selected questions. To allow an investigation of RQ3 – reusability with the same application domain, the dataset was then extended by conducting a second data recording experiment in the same application domain. An ADL scenario was selected for Experiment X2 and an indoor tracking scenario for Experiment X3. The considerations which were the basis for these decisions are described in the respective sections. The results of the data recording experiments are made publicly available.

To assess and compare the recognition performance for each dataset, two baseline classifiers were selected. One non-temporal and one temporal classifier (see Section 2.1.1). Since reasonable recognition rates were shown previously for the dataset of Experiment X1 by Giersich [79], a temporal baseline classifier was considered sufficient to demonstrate potential recognition performance. Furthermore, Experiment X1 itself was considered as baseline experiment. For each experiment, the baseline classifiers were selected and trained on the entire dataset. Section 5.2 provides a discussion of the reasons for not applying Leave-One-Out cross validation.

For Experiment X2 and Experiment X3, the introduced annotation process (see Chapter 4) was applied in order to produce causally correct semantic annotation. A re-annotation of the first dataset was omitted as the annotation seemed reliable and the publication contained no video information that could have been used.

YK2 – Design Objective of the second phase is to make design decisions about the contextual information that should be included in the model. Furthermore, the set of necessary action selection features and duration models have to be selected.

Based on the aLTS, context information that was found to be of potential interest during the annotation process (see Chapter 4) was selected. Moreover, context information, that is not part of the aLTS can be chosen. This includes, for instance, the location of persons and objects, or the set of objects currently handled by a person.

To allow an investigation of the influence of the duration model to the overall recognition performance, two different duration models were created for each experiment. A parametric model, that employs a parametric PDF to reflect possible actions durations and an empirical duration model, which restricts possible action durations to the durations actually occurred in the dataset.

With respect to the action selection model, a goal distance based action selection was selected for all experiments.

YK3 – Implementation and YK4 – Validation During the implementation phase, the decisions made in the design phase have to be implemented. This applies to the causal model as well as to the duration and action selection models.

The inference LTS The aLTS is designed to allow a validation of the annotated action sequences and to provide additional semantics by means of LTS states. However, the development of the aLTS does not target at good recognition rates, which is the design goal of the iLTS. For this purpose, the preconditions and effects of the causal model that describes the iLTS have to be more specific.

The iLTS was created based on the aLTS by a two-stage procedure. In the first step, a *feasible solution*[2] for the behaviour model was constructed iteratively based on the aLTS. In the second step this solution was refined to the iLTS by use of modifications to the actions.

In general two possible approaches can be thought of when analysing the annotated action sequences with respect to the iLTS - a sequential and a parallel approach. During the former, the iLTS is developed by considering only one annotation sequence at once – the model is adjusted to that plan. The latter, in contrast, focusses on all sequences in parallel. Thereby the model is sequentially adjusted to the different phases of the annotated scenario. The advantage of the parallel analysis is that it allows the model developer to concentrate on specific phase of the scenario rather than on the specific characteristics of the individual plans. For this reason, the parallel analysis of the annotation chosen.

[2]Feasible solution means that a first version of iLTS is developed that contains more specific preconditions and effects than the aLTS, albeit being able to validate the annotated plan sequences.

The feasible solution was created by applying the following steps:

1 The initial model was set to $\mathcal{A}_0 := \emptyset$.
2 The model $\mathcal{A}_s(s = 0, 1, \dots)$ was applied on all action sequences in parallel to identify the smallest t where action $a_{i,t}$ of dataset i fails at time t. (t has to be smaller then the length of the action sequence i). This step has two possible outcomes:
  1 If the action $a_{i,t}$ is not contained in $\mathcal{A}_s$, the action $a_{i,t}$ was added to $\mathcal{A}_s$ with preconditions and effects based on the aLTS action.
  2 If the action $a_{i,t}$ is in $\mathcal{A}_s$, but it's preconditions are not met, either the precondition of $a_{i,t}$ were relaxed, or the effects of a preceding action were extended, such that the precondition of $a_{i,t}$ were met.
3 If no failed action could be found (t equal the plan length of the longest action sequence): $\mathcal{A}_s$ is the feasible solution, as all action sequences are valid with respect to $\mathcal{A}_s$.

In order to limit the number of plans, the final iLTS was then created by further refinement of the feasible solution by addition preconditions. This final step was executed by manually analysing each action schema of the feasible solution. After each change to an action schema all action sequences were validated by use of the iLTS.

The duration model Regarding the creation of the duration model, the following method was applied. For the parametric model, a set of candidate distributions were fitted to the observed durations and the best fitted, according to the likelihood, was selected as duration distribution. The set of candidate distributions was Cauchy, exponential, gamma, geometric, lognormal, negative binomial, normal, Poisson, and Weibull. The function `fitdistr` of the package `MASS` [241] of the R programming environment [184] was used to fit the distributions. The empirical duration model was created by using the empirical CDF of the actions' durations in the action sequences of the dataset.

The process of fitting the duration models was adjusted for the individual experiment, due to their specific properties. The number of actions for Experiment X2, for instance, was found to be larger than for the other experiments. For this reason, evaluation was done based on the action class rather than on the specific action. Consequently, as the baseline classifiers were trained on the action class, also the duration model for Experiment X2 was created with respect to the action class.

The action selection model As determined in the design phase, the action selection model was based on the goal distance. For all experiments the goal distance was computed by an exhaustive process.

As Experiment X2 targeted at the investigation of the influence of the different parameters on the overall recognition performance, additional goal distance based heuristics were considered. The action selection feature $f_{\bar{\delta}}$ – the core distance – was selected as training based goal distance heuristic, as it only considers states that actually occurred during the experiment. Furthermore, f_d – the recipe distance – was considered, as it represents an alternative goal distance that was created by analysing the experimental instruction and the actual action sequences (see Table 6.6). Moreover, also the respective weights λ were adjusted to allow a detailed analysis of their influence.

YK5 – Evaluation In order to assess the performance of the model, for each experiment the inference tool created by the CCBM toolbox was applied to the observation sequence. Several different performance measures (e.g. accuracy) were then assessed and compared with the corresponding results of the baseline classifiers. The accuracy was considered as main

Factor	Description
Target	the targeted filtering distribution (e.g. forward ($p(x_t \mid y_{1:t})$), smoothing ($p(x_t \mid y_{1:T})$), or MAP-sequence ($\arg\max_{x_{1:T}} p(x_{1:T} \mid y_{1:T})$))
Model	the model that is used to represent the temporal relations of actions (e.g. non-temporal baseline classifier, HMM, or CCBM)
Mode	the inference algorithm used to reconstruct the action sequence within CCBM (e.g. MF or PF)
Observations	the observation model
Distance	the different action selection features for goal distance approximations (e.g. f_δ, $f_{\bar{\delta}}$, or f_h)
Weight	the weight λ_k of the different action selection features (e.g. 0 or 1)
Duration	the duration models (e.g. parametric or empirical duration model)
Dataset	the dataset used for evaluation (given by the identifier of the respective dataset)
Trial	the trial used for evaluation (given by the identifier of the respective trial)

Table 5.2.: Modelling factors and their meaning for experimental configurations. For each modelling factor a description and possible levels are given. Depending of the objective of the experiment the subset of considered modelling factors and their levels differ.

performance criterium, as the review of the literature revealed that it is the most common measure. A discussion of evaluation methods is provided in the next section.

To assess the capabilities of CCBM with respect to the recognition performance and the different reusability aspects, a variation of different modelling factors was used within the experiments. Table 5.2 gives an overview of all modelling factors that were considered in the experiments. Note that depending on the objective of the respective experiments only a subset of these factors is considered within each experiment. A detailed description of the levels used for evaluation of the experiments is provided in description of the respective experiment.

5.4. Evaluation Methods

In the previous section, the general experimental procedure, the experimental analysis is based on, was described. This section provides an overview of different evaluation measures used for quantitatively evaluating the performance.

5.4.1. Different Estimation Tasks

In general, the result of an inference task is a probability distribution $p(X_t \mid y_{1:t})$ for each element in the observation sequence $t \in T$. The structure of this X_t differs depending on the applied classifier. While for the baseline classifiers, the X_t usually represents a simple variable (e.g. the current action), when applying the statistical model, depicted in Figure 3.1, X_t represents a composite variable. In this case X_t consists of five variables: (1.) A_t – the current action, (2.) S_t – the current state of the environment, (3.) G_t – the goal to be followed, (4.) D_t – the termination flag of the current action, and (5.) U_t – the start time of the current action. Since the goal of this thesis is to estimate the action sequence, the goal, and contextual information, the variables A_t, G_t and S_t were selected as estimation targets.

Depending on the actual estimation task, the probability distribution of interest is generated by marginalising over $p(X_t \mid y_{1:t})$. The goal probability $p(G_t \mid y_{1:t})$, for instance, is computed by summing up the probabilities of X_t which represent equal values for G_t for each time slice. The result is a categorical PDF where the number of categories it determined by the number of goals. This is similar to the concept of statistical model checking [142], where the probability that a system S satisfies a property ϕ is determined by multiple simulations. Statistical model checking uses a threshold on the probability to decide whether the system satisfies the property

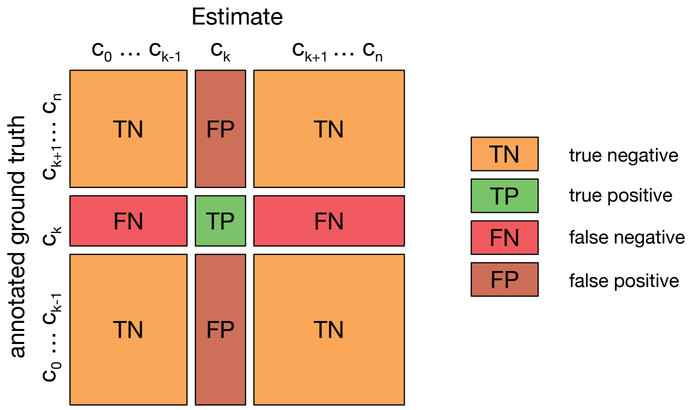

Figure 5.1.: Confusion matrix for multi-class classification. The confusion matrix of a classification with n classes. When considering the class k $(0 \leq k \leq n)$, the four different classification results can be obtained: true positive (green), true negative (orange), false positive (brown), and false negative (red).

or not. Here, likewise, a point estimate is determined by selecting the most probable state (or action) for each time-step t.

Due to the difference in the meaning of the three variables (A_t, S_t, and G_t), different evaluation methods are chosen. In the following these methods are discussed with respect to the respective estimation task.

For the multi-agent scenarios, evaluation was based on each single agent rather than on the combination. Both the estimates and the truth of all agents were strung together and then evaluated as in the single agent case. As a result, the evaluation provides the average performance of all agents.

5.4.2. Evaluation of Action Recognition

Objective of the action recognition is to estimate the current action a_t for each time-step t. Since this objective agrees with that of the domain of AR, evaluation methods from this domain are applied. Following, a brief discussion about the evaluation methods is provided.

Standard measures for AR As discussed in the Section 2.1.2 confusion matrix based measures – the accuracy in particular – were found to be the predominant performance measures (see Table A.2) in the domain of AR. Figure 5.1 illustrates a confusion matrix for the multi-class situation with n classes. The confusion matrix gives the amount of (mis)classifications for each class. The point estimates are collected into the confusion matrix $C := (c_{ij})$, where c_{ij} is the number of time-steps where the class was actually i and class j has been estimated. In general, the confusion matrix provides four types of classification results (each of them coded by a different colour in Figure 5.1) with respect to one classification target k:

- true positives (tp) – the class was estimated while it actually happens (c_{kk})
- true negatives (tn) – the class was not estimated and it did not happen $(\sum_{ij \in N \setminus \{k\}} c_{ij})$
- false positives (fp) – the class was estimated, but did not happen $(\sum_{i \in N \setminus \{k\}} c_{ik})$
- false negatives (fn) – the class was not estimated but did happen $(\sum_{i \in N \setminus \{k\}} c_{ki})$

The first two represent correct classification, while the last two represent classification errors.

Several measures can be calculated from the confusion matrix each of them providing insights on different aspects of the classification. Typically, these measures are defined for the two-class classification problem [34]. As we consider the multi-class classification, these measures are computed as the weighted mean of the class-wise measures. The following confusion matrix based measures are used in the scope of this thesis. Consider N time-steps that have to be classified into M classes.

Accuracy: The accuracy provides the amount of correctly classified time-steps by relating the number of correctly classified to the overall number of time-steps. The accuracy is the predominant measure in the domain of AR. The formula for the accuracy is provided in Equation 5.1.

$$\mathsf{Accuracy} := \frac{\sum_{i=0}^{N} c_{ii}}{\sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij}} \tag{5.1}$$

Sensitivity: The sensitivity (often called recall) represents the classifier's ability to correctly identify a given class. It is provided by the amount of truly positive predicted time-steps (tp) related to the number of time-steps where the class actually happens (tp + fn). The formula is given in Equation 5.3.

$$\mathsf{Sensitivity}_{class} := \frac{tp_{class}}{tp_{class} + fn_{class}} \tag{5.2}$$

$$\mathsf{Sensitivity} := \frac{\sum_{i=0}^{M} \mathsf{Sensitivity}_i * (tp_i + fn_i)}{\sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij}} \tag{5.3}$$

Precision: The precision represents the classifier's certainty of correctly predicting a given class. The precision relates the amount of truly positive predicted (tp) time-steps to the number of time-steps where the particular class was predicted (tp + fp). The precision is determined as given in Equation 5.5.

$$\mathsf{Precision}_{class} := \frac{tp_{class}}{tp_{class} + fp_{class}} \tag{5.4}$$

$$\mathsf{Precision} := \frac{\sum_{i=0}^{M} \mathsf{Precision}_i * (tp_i + fn_i)}{\sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij}} \tag{5.5}$$

F1-Score: F1-score provides the classifier's ability to predict a given class. The F1-score it determined by considering the classifier's precision and sensitivity. Equation 5.7 provides the formula for F1-score.

$$\mathsf{F1\text{-}score}_{class} := \frac{2\ tp_{class}}{2\ tp_{class} + fn_{class} + fp_{class}} \tag{5.6}$$

$$\mathsf{F1\text{-}score} := \frac{\sum_{i=0}^{M} \mathsf{F1\text{-}score}_i * (tp_i + fn_i)}{\sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij}} \tag{5.7}$$

Drawbacks of the accuracy Albeit being predominantly used in the literature on AR (see Table A.2), the accuracy suffers from several drawbacks: (1.) the accuracy is prone to classification due to chance, (2.) the accuracy does not incorporate the temporal sequence, and

(3.) the accuracy is based on point estimates. While the first two issues directly apply to the domain of AR, the last issue is of interest only, if the categorical PDF is used for the evaluation. In order to cope with these drawbacks, the evaluation within this thesis employs performance measures that target these drawbacks in addition to the accuracy. These measures are briefly discussed in the following.

Cohen's κ Ben-David [22] describes that the accuracy simply counts correct and incorrect classification results but does not compensate for classification due to chance. To counteract this issue, Ben-David [22] suggests the usage of Cohen's κ [51] instead. Cohen's κ *"attempts to correct the degree of agreement by subtracting the portion of the counts that may be attributed to chance"*,(Ben-David [22]). Cohen's is computed by $\kappa(C) := \frac{p_0 - p_c}{1 - p_c}$, where p_0 represents the accuracy and p_c represents a factor that is based normalised marginal probabilities.

Sequence alignment In order to assess the performance of the classifier with respect to the temporal sequence of actions, measures that are sensitive for such are applied. Measures that are based on the confusion matrix ignore the temporal sequence, as it is constructed by "counting" classification results. Thus, they can assign the same value to causally different sequences. Consider, for instance, the sequence (on, off, off). Consider furthermore the following two hypothetical estimates e_1 = (on,on,off) and e_2 = (on,off,on). Both estimates result in the same accuracy (2/3), although their causal consequence differs. Additionally, the estimate e_2 consists of three actions while the true sequence and e_1 consist of two actions, as the sequence (on, on) can be regarded as one action with a duration of two time-steps. Consequently, e_1 can be regarded as better estimate than e_2, as it reflects the same causal sequence, but mistakenly overestimates the duration of the first action. Measuring the sequence alignment provides a well established mechanism. Within this thesis, the Levenshtein edit distance [145] and the dynamic time warping (DTW) distance [82] are used. A more detailed discussion on performance evaluation in the domain of AR is provided by Ward et al. [245].

5.4.3. Evaluation of Contextual Information

A third issue when using confusion matrix based performance measures is that they rely on point estimates for both the true and the estimated state sequence. This is typically the case in AR, as the true sequence is directly taken from the annotated action sequence.

Ambiguous annotation When it comes to contextual information that is not annotated directly, but provided as additional semantic information as it is the case of the approach for causally correct annotation introduced in Chapter 4, the annotation has to be generated by "executing" the annotated action sequence within the aLTS. As described in the experimental procedure, the iLTS is developed on base of the aLTS. As long as the aLTS is unambiguous with respect to the iLTS, the resulting state sequence is deterministic. However, if additional contextual information of potential interest is identified during the development of the iLTS, the aLTS is unable to provide unambiguous annotation.

Consider, for instance, the action sequence (eat, drink, eat, clean). Now consider the task of estimating if eating has been finished. The aLTS is ambiguous with respect to this information and the execution results in a probabilistic truth. This was the case in Experiment X2 in Section 6.2. As the confusion matrix does not allow the probability to be incorporated, measures that are based on it are unable to cope with this uncertainty. Consequently, measures that consider the entire categorical PDF of states have to be applied.

Effect Size	Cohen's d		η^2	Vargha-Delanye A		Spearman's ρ	
No effect	0		0	.5		0	
Small	>.2	<-.2	>.0099	>.56	<.44	>.1	<-.1
Medium	>.5	<-.5	>.0588	>.64	<.36	>.3	<-.3
Large	>.8	<-.8	>.1379	>.71	<.29	>.5	<-.5

Table 5.3.: Overview of different effect size measures and their interpretation. Original source for interpretations: d [52, pp.24–27], η^2 [52, pp.284–288], A [240], ρ [52, pp.79–80].

The Jensen-Shannon distance The Jensen-Shannon distance (JSD) provides a measure of distance between two PDFs. Consequently, the JSD can also be applied for point estimates, as they can be interpreted as peaked distributions, where all values are 0 except for one. The JSD is employed to measure the distance between the probabilistic truth P and the estimate Q. The JSD is a metric that is defined as the square root of the Jensen-Shannon divergence [150].

5.4.4. Evaluation of Goal Recognition

When it comes to the evaluation of goal prediction, Blaylock and Allen [26] suggest the usage of convergence, convergence point, and precision. Convergence is a binary value that states if the prediction converged to the correct goal at the end of observation sequence. If the prediction converged, the convergence point gives the first time-step that predicted the correct goal without any changes afterwards. The convergence point is reported as the relative amount of the observation sequence that has been processed after the point of convergence. A convergence point of 0 means that the correct goal was predicted from the beginning of the observation sequence. The third measure to assess goal prediction performance is the precision. The precision determines the number of time-steps, where the correct goal was predicted.

5.4.5. Assessing the Size of Effects

According to the publication manual of the American Psychological Association [14, p.34], a presentation of statistical results should be accompanied by reporting of effect sizes. To this end, Ellis [70] distinguishes two types of effects: (1.) differences between variables and (2.) relationships between variables. Accordingly, the results that are reporting within this thesis are provided with a corresponding measure of the effect size.

With respect to differences of variables, two different measures are used, depending on the distribution of the values of them. Like the decision, whether to report the mean or the median of values, the Shapiro-Wilk normality test [211] was employed. The effect size of differences of normally distributed values was then reported by using Cohen's d [52]. Otherwise, the Vargha-Delanye's A [240] was employed. To allow a common interpretation of these effect size measures, Table 5.3 provides an interpretation guideline, compiled from the literature.

Regarding the presentation of effect sizes concerning the relationship between two variables, Spearman's rank correlation ρ was used. Spearman's ρ quantifies *"strength and direction of a relationship between two variables"*,(Ellis [70]). The assessment of the relationship of multiple variables was done by applying an repeated measures analysis of variance (rANOVA). Here, the generalised η^2 [175] was employed to provide the size of the effect.

The effect sizes are computed by use of the package "effsize" [234] for the R programming environment [184].

6
Experiments

"If you torture the data long enough it will eventually confess." – Ronald Harry Coase

SYNOPSIS: *The previous chapter introduced a list of research questions. This chapter's aim is now to answer these questions. For this reason three experiments are conducted, each of them targeting a sub-set of the research questions. Based on the analysis of the results of the experiments, the effectiveness of CCBM is demonstrated. Furthermore, it is highlighted that CCBM satisfies the requirements for an inference system that were derived earlier.*

CHAPTER SOURCES: *Parts of this Chapter have been previously published in the following publication(s):*

- *Computational State Space Models for Activity and Intention Recognition. A Feasibility Study [133]*
- *Plan Synthesis for Probabilistic Activity Recognition [131]*
- *Towards Creating Assistive Software by Employing Human Behavior Models [129]*
- *Where are My Colleagues and Why? Tracking Multiple Persons in Indoor Environments Krüger et al. [132]*
- *Evaluating the Robustness of Activity Recognition using Computational Causal Behavior Models [128]*

This chapter presents experimental evidence that CCBM, introduced in Chapter 3, allows the simultaneous recognition of the activity, context information and the goal of a (team of) human protagonist(s) from noisy and ambiguous sensor data. To this end, three different experiments (X1, X2, and X3) were conducted and their results are presented. The experiments aim at answering the research hypotheses stated in Chapter 5. Objective of the experiments is to show that CCBM allows to achieve good recognition performance. Each experiment utilises CCBM to recognise activities, context and/or the goal of the involved participants.

In Experiment X1, a typical team meeting situation is analysed by use of indoor location information. Aim of this experiment is to recognise the activity of each person during the meeting as well as to conclude the common goal of the team, namely the jointly agreed agenda, from the estimated sequence of activities. Furthermore, the first experiment serves as demonstration of the principal feasibility of the proposed approach. For this reason a scenario of similar complexity as related work was selected. This experiment addresses the requirement R5.1 – reusability within the same application domain. To this end, the experiment consists of two trials with different trial layout to assess the approach's capabilities to address this aspect of reusability.

In Experiment X2, activities of daily living [113], taking place within a kitchen environment, are reconstructed from noisy sensor data. The participants were instructed to execute typical kitchen tasks while being observed by wearable sensors. While the first experiment serves as baseline check, the second experiment investigates the capabilities of CCBM in more detail. The effects of the different parameters on the recognition performance are investigated. Moreover, the requirement R5.2 is targeted by demonstrating that different observation models can be used without further changes to the behaviour model.

The experiment Experiment X3 is concerned with the problem of simultaneous identification and localisation of persons in partially observed environments. Binary sensors are used to reconstruct the paths of varying numbers of persons in an indoor environment. This experiment illustrates how prior knowledge, encoded in the behaviour model, can be used to solve the identification problem. By varying the number of involved participants, this experiment allows an examination of the reusability objective R5.3, namely whether a model could be reused for different numbers of involved agents. Finally, the effect of increasing the number of agents on the recognition performance is investigated.

In the following, the experiments are discussed in more detail.

6.1. Experiment X1: Three Person Meeting

The first experiment serves as a baseline experiment to assess the basic feasibility of CCBM with respect to activity and plan recognition. Objective is the reconstruction of the plans of three persons during a meeting from location data. The reconstruction of a meeting situation from location data has been chosen for the following reasons:

- The recognition of meeting activities is a well studied domain. It has been done for example by Giersich and Kirste [80], Dai et al. [56] and Kim et al. [118].
- At the team level, a meeting situation is rather simple, but by looking at the level of the single protagonist, it shows increased complexity. This, on the one hand, allows recognition to be done by simple models such as HMMs, but, on the other hand, also justifies the use of more complex approaches such as CCBM.
- It provides a cooperative situation where multiple involved persons are trying to achieve a common agreed goal: following an agenda. The team level plan, which is typically given by an agenda, is refined by the person level plans. Consequently, the meeting situation covers all necessary aspects that are targeted in this thesis. The simultaneous recognition of the activities of multiple agents, additional context information, and the plan of the cooperative team from noisy and ambiguous sensor data.
- The use of location data for activity and plan recognition can be regarded as baseline check as it is often chosen in the literature to assess the performance of novel approaches. As has been discussed in Section 2.2, all reviewed related approaches, if using noisy sensor data at all, use location data to illustrate their feasibility (e.g. Bui et al. [37], Liao et al. [149], and Sadilek and Kautz [202]).
- Finally, since meetings are part of the everyday working life, potential subjects do not

need any introduction to the domain. This allows the data collection to be reproduced easily while changing only some aspects such as removing parts of the agenda or changing the planned time for single agenda items.

In the following the experimental setting is described in detail.

6.1.1. Objective

Experiment X1 aims at providing a comparison of the capabilities of Computational Causal Behaviour Models with related work. This is done by addressing a scenario of similar complexity (according to the identified dimensions of complexity (CD.1 – CD.3)) as other relevant work in the domain of activity and plan recognition from noisy sensor data (see Section 2.2).

With respect to CD.1, current studies on instances of CSSM use state spaces of at most 70,000 states, while targeting up to ten action classes (CD.3). Additionally, the maximum plan length (CD.2) of 20 actions limits inference complexity.

Experiment X1 investigates whether CCBM is able to handle inference tasks with recognition performances at the same level as established methods. To this end, an experiment with the following properties is conducted:

1 To compare the capabilities of CCBM to that of the related work, a trial setting of similar complexity as the related work was selected. For this purpose, a causal model, which generates an LTS of similar size of the related work, was used.

2 To assess the multi-agent capabilities of CCBM, a trial setting where multiple agents interact cooperatively was selected.

3 To demonstrate the reusability objective R5.1, a causal model was constructed for one trial setting and reused for another trial setting in the same application domain. The same model achieves good recognition rates in both trial settings.

Experiment X1 targets at answering the research questions RQ1, RQ2, and RQ3. In order to address these research questions, the first two questions were combined and reframed into one hypothesis allowing a framework of statistical testing to be applied.

H.X1 A suitable parametrised CCBM for multiple agents allows the reconstruction of the action sequence while simultaneously recognising the correct goal with similar recognition performance as a baseline classifier.

Research question RQ3 is not translated into a research hypothesis. Here, it is considered sufficient to demonstrate reusability of the causal model. For this reason, a causal behaviour model is created to address one application domain. It is then shown that the behaviour model can be reused in another trial setting of the same application domain without further changes. A detailed description of both trial settings and the differences is given below.

6.1.2. Trial Setting

For Experiment X1, the data of two trials is analysed, each of them is described in the following. Both trials are three person meetings that were held in the Smart Appliance Lab at the Mobile Multimedia Systems Group at the University of Rostock. Figure 6.1 provides a map of the Smart Appliance lab. The room is equipped with the Ubisense Ultra Wide Band indoor positioning system (represented by the red circles), where the position of each person wearing a tag can be estimated. During both meetings the position of all participants was recorded using this localisation system. Two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ were created from the data of the two trials.

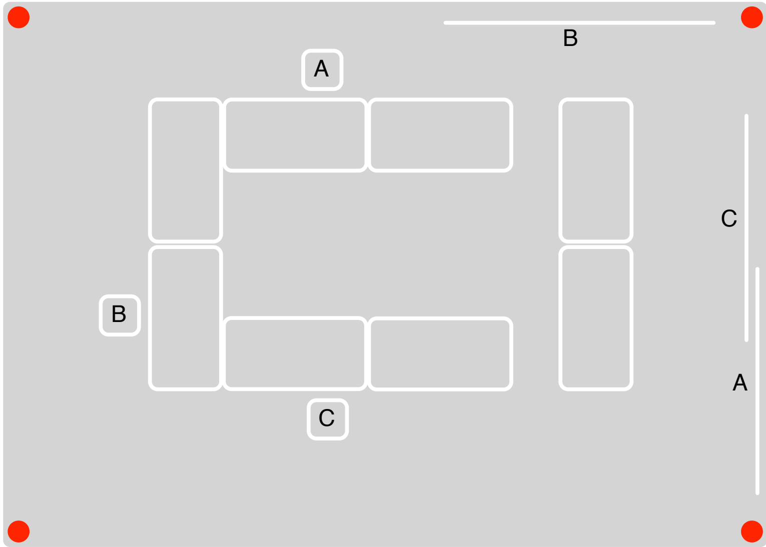

Figure 6.1.: Overview of the Smart Appliance Lab. For each person (A, B, and C) the corresponding seat and the presentation stage are illustrated. Tables are represented as rectangles, chairs as squares and presentation screens as lines. The letters highlight the chair and the presentation stage for each person. The red circles represent the Ubisense indoor tracking system.

Agenda item	compliant	non-compliant				
1st presentation	A	A	B	B	C	C
2nd presentation	B	C	A	C	A	B
3rd presentation	C	B	C	A	B	A
Frequency in $\mathcal{D}_1$	8	3	1	2	3	3

Table 6.1.: Different types of agendas including their frequencies. The last row gives the actual frequency of each possible agenda in dataset $\mathcal{D}_1$.

Trial task $\mathcal{D}_1$ Dataset $\mathcal{D}_1$ contains the location data of three persons during 20 staged meetings. Additionally, an annotation of the activities of each persons and the overall meeting phase (team activity) is included. The dataset and a detailed description is publicly available [121] and has been used in previous investigations on team tracking (e.g. by Giersich [79]). As it is not the purpose of this work to investigate recognition performance in absolute numbers but rather in comparison to baseline classifiers (see discussion in Section 5.2), it was not necessary to collect empirical data from real meetings. A scripted meeting with shortened action durations was therefore considered to satisfy the study objectives.

For each of the 20 meeting sequences three participants (A, B and C) were selected from the overall group of four (three male and one female) at the age of 26–43. One agenda was generated prior to the first meeting and two types of scripts were generated from this agenda. The first being a script compliant to the agenda, the second a non-compliant one. Each meeting consists of three presentations followed by a group discussion. Each participant was assigned a particular seat and presentation stage. An overview of the different orders of presentation including their frequencies is given in Table 6.1. For the compliant agenda, the meeting was scheduled as follows: Once the team has entered the room, person A moves to the stage in order to give the presentation. At the same time, person B and C move to their seats to listen to A's presentation. The presentation of A is scheduled for 60 s, the presentation of B and C for

Team	A	B	C
Enter	at door	at door	at door
PrepareA	move to stage	move to seat	move to seat
PresentA	present	listen	listen
PrepareB	move to seat	move to stage	sit
PresentB	listen	present	listen
PrepareC	sit	move to seat	move to stage
PresentC	listen	listen	present
Prepare	sit	sit	move to seat
Discuss	discuss	discuss	discuss
Exit	move to door	move to door	move to door

Table 6.2.: Overview of the compliant meeting. For each agenda item of the team the corresponding agent actions are listed.

90 s and 60 s, respectively. After the first presentation is finished, A moves to the corresponding seat while B moves to the stage. Person B starts the presentation. Afterwards, C moves to the stage and B back to the seat. After C has finished the presentation, C moves back to the seat and a group discussion, scheduled for 60 s, starts. Finally, all persons stand up from their seats and leave the room together. The sequence for the compliant meeting is also given in Table 6.2.

Trial task $\mathcal{D}_2$ Dataset $\mathcal{D}_2$ also addresses a meeting situation in the lab location. Objective was to provide another dataset that meets the same requirements as the first one, but comprises simple changes to the experimental configuration. In contrast to the first dataset that consists of data from 20 staged meetings, the second dataset features only one meeting. Additionally, the meeting was not staged. For the second dataset, a seminar with three students was observed. During the seminar, each student gave a presentation, while the other students were allowed to ask questions during the presentation. Neither the order, nor the duration of the presentation was scheduled beforehand, but was left up to the students. The final agreed order of presentations was: (1.) Presentation of C, (2.) Presentation of B, and (3.) Presentation of A. Similar to the dataset $\mathcal{D}_1$, the location of all participants was recorded during the meeting. The durations of the presentations of A, B, and C were 16 min, 17 min and 18 min, respectively. The duration of the overall seminar was 52 minutes. In addition to the different durations of dataset $\mathcal{D}_1$, the participants of the seminar shared a common presentation stage. Due to the possibility of questions during the presentation, a group discussion was not necessary during the seminar. The group discussion was therefore skipped.

Sensor data and preprocessing Both datasets were created by use of the Ubisense Ultra Wide Band indoor positioning system [236]. Before the start of recording, each person was equipped with a tag according to its role (A, B, or C). The Ubisense system is event-based, in the meaning that it signals changes of the position of the tag. Whenever a person (equipped with a Ubisense tag) moves, the Ubisense system captures the position and records it. The resulting dataset consists of a sequence of events, which is not necessarily alternated through the participants (i.e. the sequence might contain n events from the same tag consecutively). This means that there is not necessarily a position update for each person at each time-step.

To ease further processing, a preprocessing step was executed. For each location event in the event stream, a data row was created that contains the time of the event and the position data for each agent. In addition the data of each agent was extended by a “seen” flag signalling whether the position was recently updated or carried over from the last time slice. An extract of the sensor data from the first meeting of the first dataset can be seen in Table B.1. While this preprocessing step was already executed in the published dataset ($\mathcal{D}_1$), it was still necessary for the second dataset in order to adjust the format of the sensor data accordingly.

Observation model As described in Section 3.1, purpose of the observation model is to assign the probability of the observation to each state $p(y \mid x)$. Since each activity can be assigned to a specific location in the room, location based activity recognition can be applied. For this purpose, a state observation model has been chosen that makes use of the S component of the statistical model (see Figure 3.1). Therefore, seven different locations of interest were identified from the room layout: (1.) the door, (2.) stage and seat of person A, (3.) stage and seat of person B, and (4.) stage and seat of person C. A position was considered to be at one of these locations, if it fits in a circle of a location-specific diameter. If no location was chosen, the default location *"elsewhere"* was assumed. The meaning of the location *"elsewhere"* is that the position is somewhere in the room, but not at one of the preselected locations. This geometrical observation model was chosen from a set of different observation models, due to the results of a preceding analysis, which showed it to allow good recognition performance beforehand [131]. The probability of the sensor data signalling a specific location, when the person is actually at this position is set to .92. Again, this value is result of a preceding analysis. Consequently, the probability for each state is calculated by combining the observation probabilities for the different agents $p \in \mathcal{P}$ as in equation (6.1).

$$p(y \mid x) = \prod_{p \in \mathcal{P}} p(y_p \mid x_p) \tag{6.1}$$

$$p(y_p \mid x_p) = \begin{cases} 1, & \text{seen flag not set} \\ .92, & \text{position within range for state x and seen flag set} \\ .08, & \text{otherwise} \end{cases}$$

6.1.3. Experimental Setup

Baseline classifier HMMs were used as baseline classifiers to compare their recognition capabilities to CCBM, as they were found to be the predominant classifier in AR. These models were created from the complete set of training data in the following way. For each agenda (see Table 6.1), the unique set of team states was analysed from the training data. Here, a team state is just the combination of agent states (e.g. if all agents are sitting the corresponding team state would be sit.sit.sit). For each agenda a transition matrix was created by supervised learning for each team state sequence. Note that depending on the actual meeting sequence, the dimensions of this transition matrix might vary. To combine these transition models to a common model, a start state was introduced. The transition probabilities from the start state to the single sub-models were computed from the prior probability of each agenda. The initial state probabilities were set to zero, except for the start state, the probability of which was set to one. The resulting model consists of 146 states, with varying number of states per sub-model. A graphical representation of the constructed transition matrix is given in Figure 6.2. This type of HMM is also known as joint HMM in the literature [42]. Purpose of the so constructed HMM is to provide a baseline classifier for simultaneously recognising the agent's activity and the final agreed agenda of the team. Here, completing the agenda is interpreted as the common team goal.

The application of Bayesian filtering algorithms to the model results in a probability distribution over possible states. Each state of the constructed HMM is a 4-tuple with the following elements: 1. action of A, 2. action of B, 3. action of C, and 4. agenda of the meeting. The probability of an agenda being followed by the team is calculated by marginalising over all states

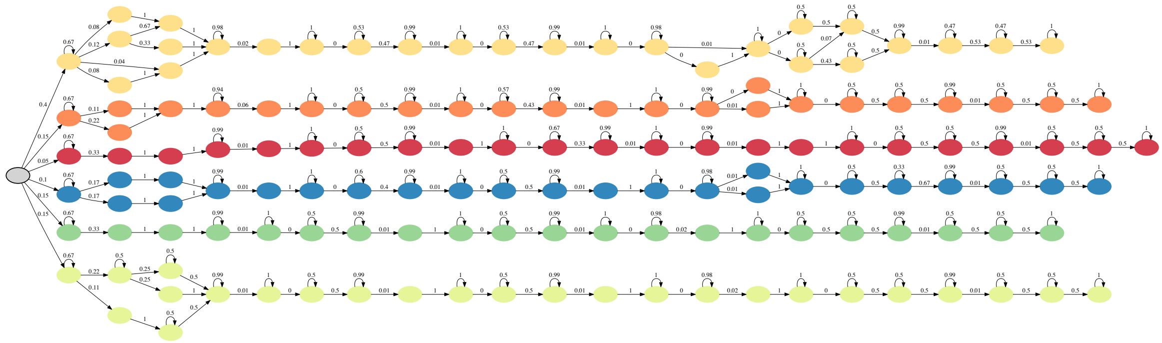

Figure 6.2.: Graphical representation of the transition matrix of the baseline HMM for the meeting experiment. Different colors represent different agendas. Grey represents the initial state. Note that values are rounded, which means that values of zero actually represent very small values.

with the same agenda. The probability of the agent activities is estimated in a similar way. For each possible agent activity, the probability of the activity being executed by the agent is estimated by marginalising over those team states that comprises the same activity for this agent (e.g. summing up the probabilities of all states that contain the activity *sit* at the first position of the team state). In this way, each filtering run provides a probability distribution of activities per agent and a probability distribution of agendas.

Causal behaviour model The causal behaviour model was created by following the model development process provided in Section 5.3. In fact, the meeting model used within this thesis is an adapted version of the meeting model introduced by Yordanova [256]. To support the distinction of different agendas it was extended by a mechanism to track the agenda. Six team goals were introduced, each of them reflecting one possible agenda of the three participating protagonists (all permutations of three items). Figure 6.3 illustrates the action *start-presentation*, where agenda tracking was implemented. Note that this mechanism is specialised for the case of three agents and does therefore not allow to generalise for different numbers of protagonists. A more flexible approach was implemented for the third experiment in Section 6.3.

Model characteristics To provide an idea of the model with respect to the related work, in the following the size of the causal model is reported. The causal model consists of 16 action schemata of ten activity classes, which when grounded result in 78 grounded actions. The state space if formed by 88 state features, 29 describing the state of one of the agents, respectively. An exhaustive exploration of the state space revealed an overall number of states of 23,717 per goal. By taking also the six different goals into account, the number of states to be considered during inference is 142,302 states. The minimal number of actions to be applied in order to reach a goal state from the initial state is 48. This goal distance is obviously equal for each goal. Since the agenda to be finished is a team goal, each agent is required to perform actions in order to reach it. Thus, the goal distance here represents the number of actions to be performed of all agents.

Duration model As described in the overall experimental procedure, two duration models were created, a parametric and an empirical duration model. Due to the high number of training samples per grounded action, it was possible to train the durations specific to the

```
(:action start-presentation
  :parameters (?p - person ?s - stage ?n - (number 1 3))
  :agent ?p
  :duration (present (person-id ?p))
  :precondition (and (not (has-presented ?p))
    (at ?p ?s) (idle ?p)
    (forall (?p1 - person)
      (imply (not (seated ?p1)) (= ?p ?p1)))
    (or
      (and (= ?n 1)(= (first) none))
      (and (= ?n 2)(and (not (= (first) none))(= (second) none)))
      (and (= ?n 3)(and (not (= (first) none))(not (= (second) none))
            (= (third) none)))
    ))
  :effect (and
    (is-presenting ?p) (not (idle ?p))
    (when (= ?n 1) (assign (first) ?p))
    (when (= ?n 2) (assign (second) ?p))
    (when (= ?n 3) (assign (third) ?p))
    (forall (?a - activity) (and
      (when (= ?a presenting)(isActive ?p presenting))
      (when (not (= ?a presenting))(not (isActive ?p ?a)))))
    ))
```

Figure 6.3.: The action start-presentation. The action *start-presentation* comprises a mechanism for tracking the order of the agendas, which are defined by the different goals.

grounded action in favour to the action classes.

For the empirical duration model, the frequencies of each duration was also considered in the duration model.

For the parametric duration model, from the set of probability density functions listed in Section 5.3, the following distributions where selected as best fitting: (1.) gamma distribution, (2.) Weibull distribution, (3.) normal distribution, (4.) lognormal distribution, and (5.) Cauchy distribution Table B.2 gives an overview of the selection distributions including their parameters. For the second dataset, the creation of an empirical duration model was omitted due to the low number of training samples. Instead the duration of all action was assumed to be normally distributed. Since the purpose of the second dataset is not to show high recognition rates but reusability within the application domain, the selection of appropriate duration distributions was not an issue.

Reusability aspect R5.1 In addition to comparing the performance of CCBM to that of the baseline classifier, this experiment aims at demonstrating reusability within the same application domain. For this reason, the dataset $\mathcal{D}_2$ was considered. By achieving high recognition results when using the second dataset $\mathcal{D}_2$ the causal model can be shown to be reusable. For this reason only application-specific parameters were adjusted. The parameters to be adjusted are: (1.) the duration model, (2.) the observation model, and (3.) the goal state. Obviously the duration model from the first dataset cannot be reused. The durations of all actions differ by an order magnitude. Due to the freedom of choice in the seat of each student during the second trial, the observation model has also to be adjusted. The missing team discussion phase has to be removed from the agenda, and thereby from the goal state. Objective of this experiment is to show that the causal model can be reused without further adjustments. Consequently, no action definition has been changed.

Factor	Level	Comment
Target	f	filtering distribution $p(x_t \mid y_{1:t})$
Model	HMM	HMM transition matrix
	C	CCBM model
Mode	M	Marginal filter
	P	Particle filter
Distance	f_δ	True goal distance, complete state space
Weight	Lλ	$\lambda_\delta = 1$
Duration	τ_c	continuous parametric duration models
	τ_d	discrete duration models based on empirical distribution function
Dataset	$\mathcal{D}_1$	short meeting sequences from Kirste [121]
	$\mathcal{D}_2$	long meeting sequence

Table 6.3.: Factors and levels for the Meeting experiment.

Experimental procedure To assess the performance of the meeting model on $\mathcal{D}_1$, multiple filtering runs were performed. For each filtering run different modelling factors (see Table 5.2) were changed to assess their influence on the recognition performance. The factors to be changed for the different runs are:

1 Model: describes the type of model is used to represent temporal relations: HMM and CCBM
2 Mode: describes the inference algorithm used for CCBM: MF and PF
3 Distance: gives the goal-based action selection heuristic: f_δ
4 Weight: gives the weight of the action selection heuristics: $\lambda_\delta = 1$
5 Duration: gives the duration model: parametric (τ_c) and empirical (τ_d).
6 Dataset: gives the dataset: $\mathcal{D}_1$ and $\mathcal{D}_2$

The factors Model and Mode were combined to a common factor Method. Method consists of the levels: HMMf, CMf, and CPf, with the meaning HMM forward filtering, CCBM marginal filtering, and CCBM particle filtering, respectively. Forward filtering was considered sufficient to assess the model performance in all cases. Table 6.3 provides an overview of all factors including their possible levels. The following configurations were chosen from the list of factors:

(HMMf, $\mathcal{D}_1$): Baseline evaluation of exact inference for trained temporal classifier. Obviously, due to the different temporal relations, the HMM trained on dataset $\mathcal{D}_1$ cannot be used for dataset $\mathcal{D}_2$
(1 configuration)

({CMf,CPf}, $\{\tau_c, \tau_d\}$, $\{\mathcal{D}_1, \mathcal{D}_2\}$): Analysis of the influence of selected CCBM factors to the overall performance.
($2 \times 2 \times 2 = 8$ configurations)

Each of the configurations is applied to all meeting sequences of the particular dataset.

Experiment execution Objective of each run was to simultaneously estimate (1.) the current activity of each agent, (2.) the team goal, and (3.) context information about each agent. As context information, here, we consider whether an agent is currently seated or not. The resulting output consists of a probability distribution over all possible values for each estimated target. Each of them were created by marginalising over all particles. For the PF, to compensate sampling or resampling based issues each run was repeated 20 times and consolidated afterwards by merging the estimated densities. Overall, 20 runs of the HMMs, 400 runs of the PF, and 20 runs of the MF were executed. For each run (consolidated in case of PF) the

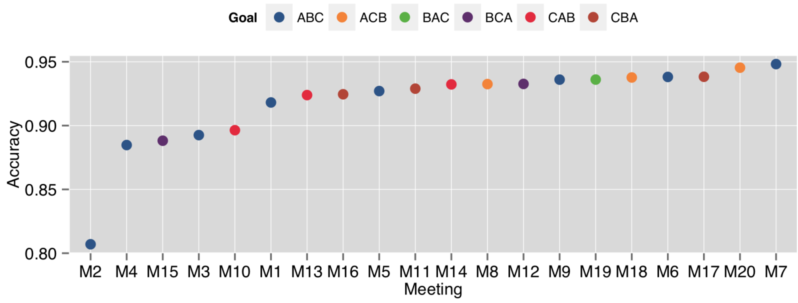

Figure 6.4.: Overview of the accuracies of the baseline HMM for the meeting experiment. Meetings are sorted according to their accuracy.

performance was evaluated. Additionally, the variance that was introduced by the random number generator was assessed. A detailed description of performance measures is given in Section 5.4.

6.1.4. Results

Baseline classifier for $\mathcal{D}_1$ In the following, first the results for the first dataset ($\mathcal{D}_1$) are presented. Regarding the performance of the AR, the HMM achieved a median[1] accuracy of .93 (with interquartile range $IQR = .91 - .94$). The simultaneous recognition of the agenda achieved a median precision of .75 ($IQR = .6 - .99$). The goal recognition converged for each meeting sequence. The median convergence point was .27 ($IQR = .25 - .59$), which means that the agenda was correctly recognised after processing 27% of the observation data. Interestingly, for the sixth meeting sequence convergence was achieved at the first time-step, giving a precision of 1. Figure 6.4 gives an overview of the baseline (HMMf) recognition results. No significant correlation between accuracy of AR and precision of goal recognition could be observed (ρ=$-$.036, S=1378, p=.88). This is is also illustrated in Figure 6.8. Figure 6.7 illustrates the estimated goal PDF for M17. The probability of each goal at each time-step is given. The convergence point (cp=.59) is marked as white triangle.

CCBM action recognition for $\mathcal{D}_1$ Considering the results of the CCBM, an overview of the results for ({CMf,CPf},$\{\tau_c,\tau_d\}$,$\{\mathcal{D}_1\}$) with respect to activity is given in Figure 6.5. For parametric timing, the median recognition accuracy of CMf was .92 ($IQR = .92 - .93$). CPf achieved an AR accuracy of .92 ($IQR = .92 - .93$) in median. This represents small effects in comparison to the recognition of the baseline classifier (Vargha Delaney's A=.64 for CPf and A=.59 for CMf). Furthermore, the difference to the baseline classifier was neither significant for CMf nor for CPf. By using empirical timing, the MF achieved a median accuracy of .93 ($IQR = .92 - .93$) and the PF of .93 ($IQR = .91 - .93$). Both effects were negligible (A=.55 for CPf, A=.54 for CMf). A detailed overview of all recognition results is provided in Table 6.4.

CCBM goal recognition for $\mathcal{D}_1$ With respect to recognising the goal, CCBM achieved a median precision of .96 ($IQR = .81 - 1$) for the MF (CMf) and .98 ($IQR = .79 - 1$) for the PF (CPf). As observed for the HMMs, by applying CCBM, all meeting sequences converged to the correct goal. The median point of convergence was .08 ($IQR = 0 - .24$) for CMf. CPf converged to the correct goal in median after 2% ($IQR = 0 - 22$)% of the observation sequence.

[1]The median accuracy is reported here, as the accuracies were not found to be normally distributed.

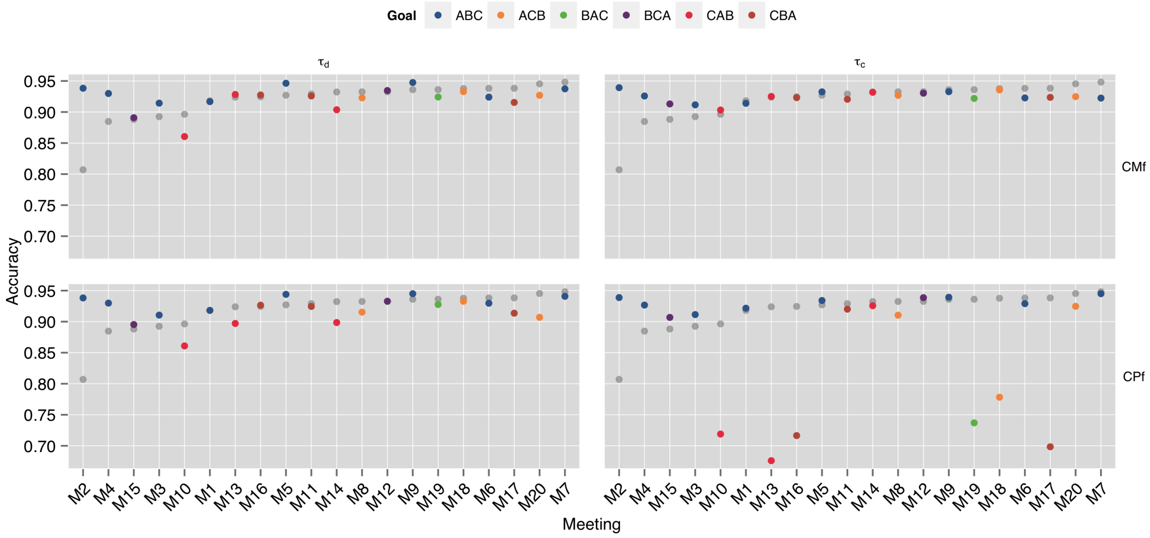

Figure 6.5.: Overview of the results of all CCBM configurations for the meeting experiment. Single meetings are sorted according to their baseline accuracies (given in grey).

Duration	Mode	Target	V	p_V	M_V	p_{SW}	A
τ_c	CMf	Activity	91.00	.61	-0.00	$<$ .001	0.59
τ_d	CMf	Activity	93.00	.67	-0.00	$<$ .001	0.54
τ_c	CPf	Activity	61.00	.1	-0.02	.003	0.64
τ_d	CPf	Activity	76.00	.46	-0.00	$<$ .001	0.55
τ_c	CMf	Goal	171.00	.002	0.12	$<$ .001	0.26
τ_d	CMf	Goal	168.00	.004	0.12	$<$ .001	0.26
τ_c	CPf	Goal	184.00	$<$ .001	0.16	$<$ .001	0.26
τ_d	CPf	Goal	173.00	.002	0.12	$<$ .001	0.28

Table 6.4.: Performance comparison of different CCBM configurations with corresponding HMM configuration. Wilcoxon signed rank tests were used for comparison. p_{SW} gives the p-value for the Shapiro-Wilk normality test, A gives the Vargha Delaney effect size.

Figure 6.6 provides a plot of the convergence points for the different classifiers. Additionally, the relation between the achieved point of convergence and the convergence point of the true action sequence is shown. The true convergence points are computed by analysing the true action sequence. Each sequence contains two possible convergence points, when the first or the second presenter moves to the stage. The first point is selected, when the prior goal probability allowed to conclude the correct goal from this point (i.e. the prior goal probability for the correct goal is higher than for the other remaining option), the second otherwise.

As can be seen from the plots, significant correlations between the achieved and the true convergence exist for the CCBM configurations (Spearman's $\rho = .87$, $S = 176$, p$<$ .001 for CPf, $\rho = .94$, $S = 79.3$, p$<$ .001 for CMf).

The probabilities for each goal at each time-step with all methods for M17 is depicted in Figure 6.7. The goal recognition precision is provided by the bottom line, which represents the most likely goal for each time-step.

Comparison to baseline Both, precision and convergence point signal a significant increase of the goal recognition performance, when comparing to the baseline classifier. The median

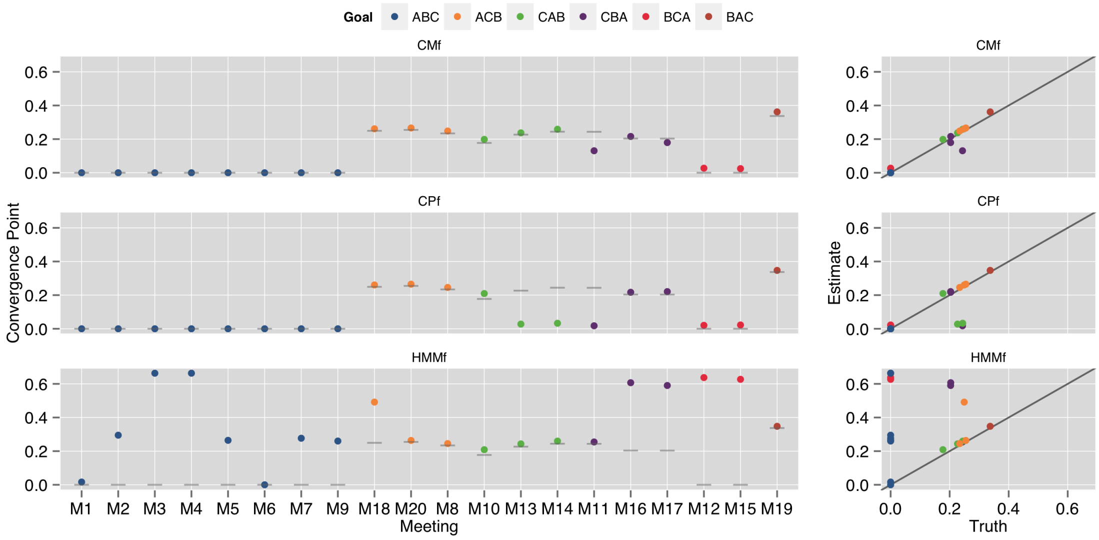

Figure 6.6.: Convergence points for the baseline HMM and the τ_c-based CCBM configurations. The goal is represented by the color of the point. Left: For each meeting sequence the convergence point is provided. The grey line highlights the earliest possible convergence point according to the prior goal probability and the annotation. Right: The relation of achieved convergence point and minimal convergence point according to the annotation is given. The grey line signals perfect goal recognition. Points above this line represent late convergence, points below this line represent premature convergence.

	prior	HMM	CMf	CPf
ABC	0.40	0.88	1.00	1.00
ACB	0.15	0.74	0.74	0.74
CAB	0.05	0.77	0.86	0.91
CBA	0.10	0.53	0.86	0.85
BCA	0.15	0.37	0.97	0.98
BAC	0.15	0.65	0.64	0.65

Table 6.5.: Mean precision per goal for each classifier. The column prior gives the prior probability of each goal.

precision increased by 21pp ($V_{(20)} = 19$, p=.002) when comparing CMf to HMMf. A large effect (Vargha and Delaney's A=.26) was observed. The median convergence point decreased by 19pp ($V_{(20)}$=179, p $<$ .001). Again, the effect size (A=.84) was found to be large.

For CPf, differences of 23pp ($V_{(20)} = 6$, p=$<$ .001) and 25pp ($V_{(20)}$=182, p $<$ .001) for precision and convergence points were observed. Large effect sizes were observed for both (A=.26 for the precision and A=.86 for the convergence point). A significant correlation of AR accuracy and goal recognition precision could be observed for CPf (ρ=.66, S=448, p=.001). This relation was not observed for CMf.

Figure 6.8 gives illustrates the relation of AR accuracy and goal recognition precision for CCBM and the baseline HMM. An overview of the mean goal recognition performance per goal is given in Table 6.5. It could be observed that for each goal the mean goal recognition precision of CCBM is at least as high as the precision of the HMM. One exception is the precision of the CMf for the goal BAC, which is 1pp below the precision of the HMM.

For empirical timing (τ_d), a median precision of .98 ($IQR = .76 - 1$) for CPf and .95

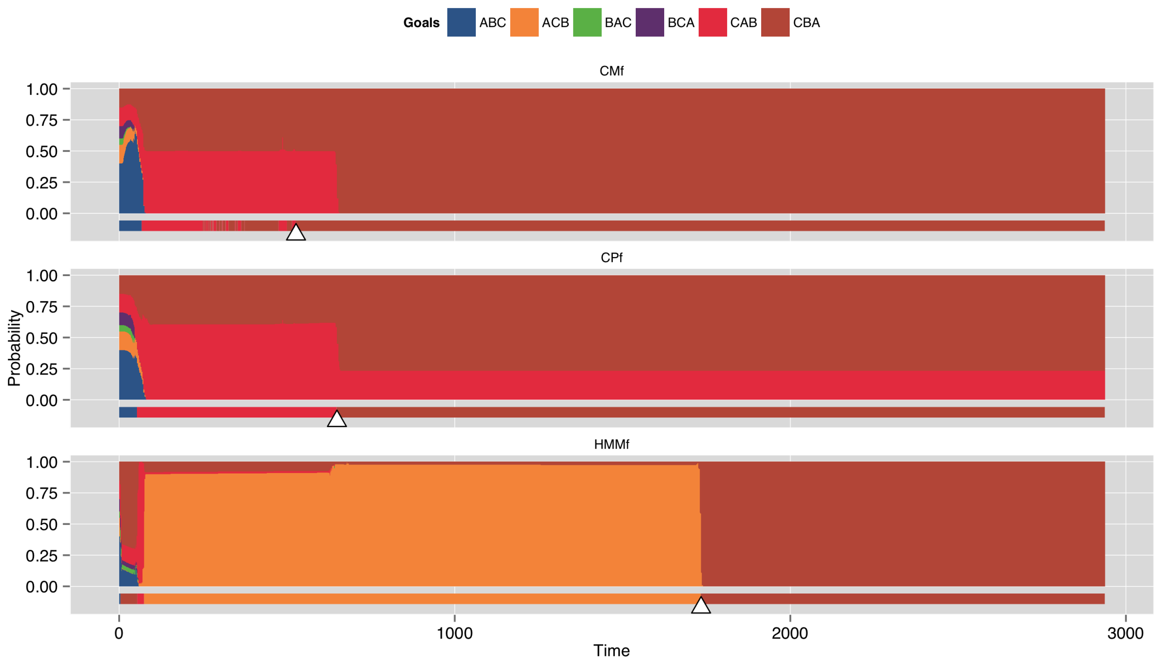

Figure 6.7.: The goal PDF for M17 of the HMM baseline classifier and τ_c-based CCBM classifiers. Each goal is represented by a different color. For each time-step the probability of each goal is provided by the size of the bar. The bottom line illustrates the most likely goal for each time-step. The white triangle marks the convergence point.

($IQR = .79 - 1$) for CMf was observed, no difference in median to τ_c. Furthermore, no difference (in median) could be observed for the convergence point.

Alternative performance measures In addition to the accuracy, performance measures sensitive to the causal sequence of the action sequence were calculated. With respect to the normalised DTW, for the sequence of actions estimated by the HMMf a median distance of .022 ($IQR = .005 - .033$) was observed. CMf and CPf estimated action sequences with median difference of .024 ($IQR = .019 - .027$) and .014 ($IQR = .01 - .071$). The median Levenshtein edit distance of the estimated action sequence with the true sequence was .066 (.069, .068) for the HMMf (CMf, CPf). No significant differences were found for both.

Context recognition Concerning the recognition of contextual information, Figure B.1 provides a graphical illustration of the probability of each agent being seated or not. From the figure, the three phases, presentation of C, presentation of B, presentation of C can be seen. As the dataset $\mathcal{D}_1$ was not annotated with the process described in Chapter 4, a context annotation is missing. Thus, assessing the performance for estimation contextual information is not possible. However, Figure B.1 exemplifies the capabilities of CCBM to estimate contextual information.

CCBM action recognition for $\mathcal{D}_2$ Concerning dataset $\mathcal{D}_2$, CMf achieved an accuracy of 1 for τ_d and .99 for τ_c. The PF (CPf) achieved an accuracy of 1 for τ_d and .99 for τ_c.

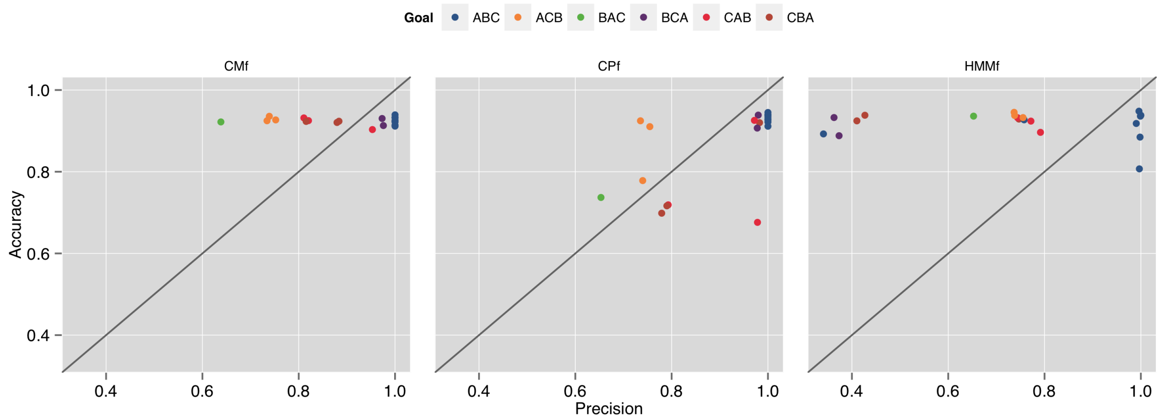

Figure 6.8.: Relationship of activity recognition and goal recognition of CCBM and baseline classifiers. CCBM plots were based on τ_c-duration models. The grey line highlights perfect positive correlation.

Influence of the random number generator To assess the influence of the random number generator on the recognition performance of the PF, 20 different random seeds were used for initialisation. The variance of the accuracies that was introduced by the seed is $1.26 * 10^{-6}$ ($IQR = 5.68 * 10^{-7} - 4.58 * 10^{-6}$). To compare, the variance that was introduced by the different meetings is $3.75 * 10^{-4}$ ($IQR = 3.75 * 10^{-4} - 3.75 * 10^{-4}$). The variance of the accuracies of the different meetings is significant ($W_{(20)} = 380$, p=< .001) larger (A=.95) than the variance of the different seeds.

6.1.5. Discussion

The following section aims at answering the research questions stated above. To do this, in the first step, the research hypothesis H.X1 is investigated. Afterwards, the remaining research question (R3) is answered.

Complexity of the causal model Objective of the first experiment was to show that CCBM is able to handle inference tasks for scenarios with same complexity as related approaches (see Section 2.2). As discussed, the maximum state space size handled by related approaches was 70,000 states. The maximal plan length was 20 steps and the approaches were used to distinguish up to ten classes. For the first experiment, a model with state space size of $6 \times 23,717 = 142,302$ states was created. This exceeds typical state space sizes by the factor two. The number of classes to be distinguished (see Table 6.2) for the scenario was ten per agent. The plan length for each agent was ten. By also considering that the agents are independent and not fully synchronised the actual plan length increases. However, the problem was considered to be similar to the problems discussed in the literature as it does not underestimate the complexity of the model of the current state of the art. Like all related approaches that incorporate real sensor data, inference based on location data was chosen.

Hypothesis H.X1 Hypothesis H.X1, which states that CCBM allows the simultaneous recognition of the user's activity and goal at the same level as trained baseline classifiers has to be accepted (see Section 6.1.4). In fact, CCBM is able to recognise the activities of each agent at the same level as the trained HMM. This result is also confirmed by considering performance measures sensitive for causality (DTW and Levenshtein distances). No significant difference was observed. Albeit not considering the causal sequence of actions in general, examining the transition matrix of the trained HMM (see Figure 6.2) reveals that for each single goal a strong sequence is enforced. This sequence is rarely interrupted. The HMM consists of 146 states, resulting in $146 \times 145 = 21,170$ degrees of freedom, at least theoretical. Thus, it is obviously overfitted. While nevertheless CCBM was able to achieve recognition rates at the same level, this indicates that the problem of reconstructing meeting sequences from location data can be solved easily. On the other hand it confirms location-based AR as baseline test for model feasibility.

Goal recognition When it comes to goal recognition, CCBM outperformed the baseline classifier (see comparison to baseline in previous section). Estimating the correct goal within the HMM was basically selecting the sub-model, which was best fitted to the sequence of actions (including timing of actions by self transitions). Thus, action sequences following the same goal but started with different durations did not necessarily result in the same recognition rates (see Figure 6.4). CCBM was, for instance, able to recognise the, most likely goal (ABC) consistently good. The HMM in contrast did not show such consistency. Table 6.5 gives an overview of the mean goal recognition precision for each goal for each classifier. This provides evidence for the superiority of goal distance based action selection in contrast to the trained transition probabilities of the HMM in this scenario.

Influence of the duration model The impact of the duration model to the goal recognition as well as the AR was neither significant for CMf nor for CPf. Although increasing inference complexity, the results indicate that a parametric representation of the action durations can substitute an empirical representation. Additional benefit with respect to reusability is provided by the increase of flexibility. However, additional research is required, covering for instance the generation and evaluation of parametric duration models by use of cross validation.

Influence of the random seed A comparison of the variances introduced by the different sources (meeting sequence or seed) showed that the influence of random number generator used in the PF is small. The influence of the different meeting sequences is at least one order of magnitude larger than that of the initialisation seed. Moreover, the variance that was introduced by the seed itself is only small. As a result, the compensation of the influence of the random number generator seed can be omitted in further experiments.

Reusability aspect R5.1 Concerning the third research question (RQ3), it was found that the causal model of human behaviour developed for the dataset $\mathcal{D}_1$ could successfully be reused for dataset $\mathcal{D}_2$. The basic model, containing the causal constraints of the meeting domain was reused without changes. The only changes that were necessary affected the duration and the observation model. The results indicate that CCBM allows to reuse the behaviour model within the same application domain.

Summary In summary, Experiment X1 showed that CCBM is indeed able to reconstruct action sequences in problems of the same complexity as addressed in the related work. CCBM performs on average at the same level as standard methods, provides earlier convergence, and allows for reusability. It has been proven that the use of multiple interacting agents, albeit increasing inference complexity (due to action and state space size increase), allows to achieve high recognition rates. Additionally it has been shown that CCBM allows to simultaneously recognise the goal, contextual information as well the action sequence from location data.

Furthermore, it has been demonstrated that a causal model created for a specific inference task can be reused within the same application domain. Based on these baseline results, the experiment in the next section exploits the capabilities of CCBM in a domain with a complexity exceeding the current state of the art by several orders of magnitude.

6.2. Experiment X2: Kitchen Task

Experiment X1 illustrated the basic feasibility of CCBM to simultaneously recognise the activity, context information and the plan from noisy location data. It has been shown that CCBM is able to handle scenarios with the same complexity as CSSMs have been used for in the literature. Based on this result, the second experiment now aims at demonstrating the capabilities of reconstructing action sequences in a more complex scenario. To do this, a typical kitchen task has been chosen that focusses on the sequence of meal preparation, consumption and kitchen cleanup. This scenario was selected based on the following considerations:

- Meal consumption can be considered as relevant ADL, while meal preparation and cleanup are relevant instrumental ADLs [6]. A sequential combination of both represents a typical episode in every day living. Providing assistance for (instrumental) ADLs is therefore an important application domain of assistive systems.
- Meal preparation is an activity that is used in the Kitchen Task Assessment [20], a procedure used as functional measure for assessing the level of cognitive support required by persons suffering from cognitive decline. The work of Serna et al. [209] provided evidence that erroneous behaviour, especially in this setting, can be modelled by use of action languages.
- Kitchen activities are frequently used as trial settings for AR (see Table 2.2). It has for example been addressed by Rohrbach et al. [196] and Ramírez and Geffner [188].
- Albeit being trivial from the viewpoint of execution by healthy humans, the task provides a non-trivial causal structure. This includes situations where several actions can be executed in any order, resulting in high branching factors (e.g. several objects (plate, knife, spoon, pot, ...) have to be cleaned.) On the other hand, the task provides phases that have to be executed in strict causal sequence (e.g. water has to be filled in the pot in order to cook). The latter point makes the usage of causal models, especially CCBMs, meaningful.

A detailed description of the experiment, including objectives, the trial setting and results are provided in the following sections.

6.2.1. Objective

The related work use relatively small studies to show the effectiveness of CSSMs. Researchers of CSSM-like approaches addressed in different studies problems tracking at most 70,000 states (CD.1), distinguishing at most ten action classes (CD.3) and action sequences not longer as the maximal plan length of 20 (CD.2). No study addressed the limits of all complexity dimensions simultaneously. Objective of Experiment X2 is to investigate the capabilities of CCBM in a setting with a problem size that is larger than that of the related work. In detail, the following objectives were targeted by Experiment X2:

1 While Experiment X1 demonstrated that the CCBM approach is able to address problems of similar size as related work, objective of Experiment X2 is to extend the problem size by extending each of these complexity dimensions (CD.1 – CD.3) at the same time, while still being able to achieve good recognition rates.

2 Table 3.1 introduced different sub-models, each of them providing parameters that influence recognition. Experiment X2 aims at investigating the influence of these parameters and to show that these factors indeed allow flexible configuration to achieve good performance. To this end, the influence of each parameter on the recognition rate has to be examined.

3 In the literature, it has been stated that *"wearable sensors are not suitable for monitoring activities that involve complex physical motions and/or multiple interactions with the environment"*,([46]). The choice of wearable sensors allows an investigation of this statement with respect to more refined system models, as generated from the causal description.

4 Reusability aspect R5.2, namely reusability regarding the observation model, is investigated. This is done by exploiting both components of the observation model, action observation $p(z \mid a)$ and state observation $p(w \mid s)$ (see Section 3.1) without further adjustment to the causal model.

Experiment X2 therefore addresses the research questions RQ4 – RQ7: The research questions RQ4 and RQ5 are reframed into hypotheses, allowing the research questions to be answered, if proved to be true.

H.X2.1 A suitably parameterised CCBM for a typical activity of daily living will achieve the same accuracy on average on a given estimation task as a standard baseline classifier built from training data when applied to the same task.

H.X2.2 All CCBM modelling factors and their interactions have significant effects on the average accuracy achieved in state estimation using the CCBM model.

The research questions RQ6 and RQ7 are answered by adjusting the experimental procedure accordingly. By selecting wearable sensors as data source, and answering the research question RQ4, research question RQ6 will be answered too. Likewise, the last research question (RQ7) is not translated into a hypothesis. Similar to the investigation of R5.1 in Section 6.1, it is considered sufficient to demonstrate reusability by achieving reasonable recognition rates. In the following, the trial setting is described.

6.2.2. Trial Setting

The trial of Experiment X2 addressed a typical meal time routine. A detailed description of the trial setting is provided below.

Trial task During the trial, participants had to accomplish the following four main tasks: (1.) Prepare meal, (2.) set table, (3.) consume meal, and (4.) clean up and put away utensils. A more detailed schedule of the task sequence is provided in Table 6.6.

Neither absolute motion trajectories nor absolute action duration were of relevance (see discussion in Section 5.2). For this reason, participants were instructed to shorten some actions to bound the overall experiment duration. Additionally, this decreases the influence of single actions to the overall recognition performance. Consider, for example, an action that can easily be recognised (e.g. cooking or eating) with long duration, lasting 50% of the experiment duration. An overall recognition rate of 50% would easily been achievable. However, this overestimates the overall recognition performance. As physical environment, a simplified motion capturing environment was used where some of the kitchen utensils (e.g. the stove) were replaced by physical props (cp. [110]). A symbolic map of the spatial structure of the trial domain and the involved domain objects is given in Figure 6.9.

The trial execution was accompanied by an experimenter, who presented the experimental task verbally to the participants and explained the physical layout, the props, and their

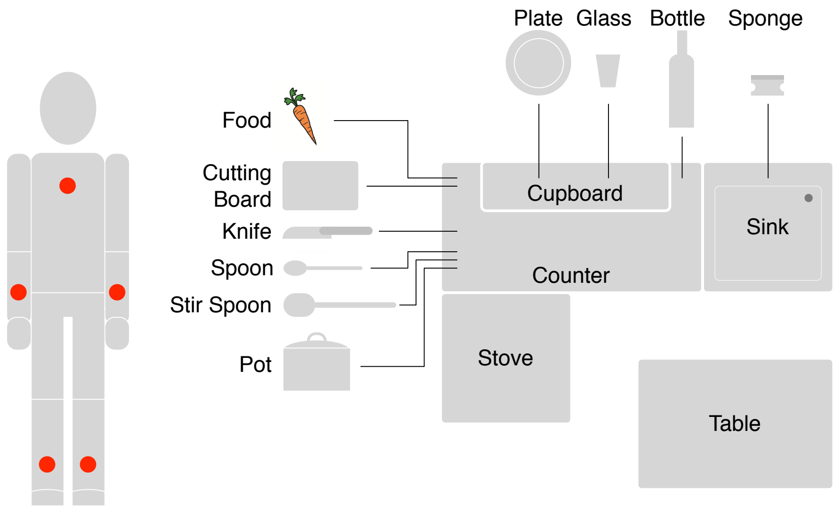

Figure 6.9.: The physical trial setup of the kitchen experiment. Left: Instrumentation of the participants. The red circles represent the placement of each sensor. Right: Conceptual spatial layout as viewed from above. The lines represent location of involved object at start of the trial.

use. After the introduction to the trial, participants were instrumented with the wearable sensor equipment. The participants were selected sequentially without any specific order. The participants were monitored by the experimenter during the trial. In case of questions, the experimenter indicated next steps of the task. The sequence of actions was up to the participants, as long as the causal dependencies allowed the execution. The entire trial was recorded on video.

Subjects Regarding the overall number of participants, seven subjects (six male and one female) were considered sufficient to detect relevant effects on accuracy. The rationale here is that if CCBM can not be proven to be inferior at this level, then this justifies to spend the effort on a larger scale experiment. Although there is no direct data on how much experimental data is required for building successful causal models, a weak argument can be found in the domain of usability research, where it is established that in interactive software five to seven subjects are sufficient to identify most usability problems – most situations where system behaviour does not meet user expectations [170].

6.2.2.1. Sensor data and preprocessing

Sensor setup The XSens MVN motion capturing equipment [195] was used for recording sensor data. It consists of wearable IMUs, each of them recording three axis acceleration and angular rates. Besides the initial motivation of answering research question RQ4, this sensor setup was preferred to other setups (e.g. radio-frequency identification (RFID) labelling [180], cameras [2], or multi-modal setups [251]) for the following reasons:

- The use of wearable sensors is dominant in the field of AR (see Section 2.1.2 and Bulling et al. [40]). The main purpose of this work is to combine AR from low-level sensors with context and plan recognition. The choice of wearable sensors, therefore, seems expedient.
- This sensor setup is used in several experiments by various researchers [139, 59, 60], enabling an easy adaption of the CCBM method to other available datasets in the future.
- Wearable sensors do not require the instrumentation of the environment. Consequently,

Step	f_h	Task	State predicate(s) if task not fulfilled
1	14	Clean hands	– (no state predicate with lower step count matches)
2	13	Get food to sink	not clean food, holds food, not at sink
3	12	Clean food	not clean food, holds food, at sink
4	11	Cut food	clean food, not food prepared
5	10	Turn stove on	food prepared, not cooked, stove off
6	9	Cook food	food prepared, not cooked, stove on
7	8	Turn oven off	hungry, cooked, stove on
8	7	Finish setting table & Sit down	hungry, cooked, stove off, not seated
9	6	Consume meal	hungry, stove off, seated
10	5	Get up	not hungry, seated
11	4	clean kitchen utensil	not hungry, not seated, 4 kitchen utensil dirty
12	3	clean kitchen utensil	not hungry, not seated, 3 kitchen utensil dirty
13	2	clean kitchen utensil	not hungry, not seated, 2 kitchen utensil dirty
14	1	clean kitchen utensil	not hungry, not seated, 1 kitchen utensil dirty
15	0	done	

Table 6.6.: Task script of the kitchen trial For each step the distance values of the recipe distance (f_h) and state predicate(s) that hold before this step are provided.

they are a technically and economically feasible choice for everyday environments. Furthermore, the ubiquitous existence of IMUs in smart phones, smart watches, and fitness tracker allows to draw conclusions about CCBM's suitability for daily use.

- IMUs are not influenced by environmental factors such as lighting conditions. Additionally, they allow a identifying observation of the subjects, preventing identification problems as in the case of sensor carpets or similar environmental sensors (see Section 6.3). The latter point becomes relevant in real-world applications.

The participants were instrumented with five IMUs, fixed at lower legs, lower arms, and upper back. The positions are given by the red circles in Figure 6.9. These sensor locations were chosen to be compatible with sensor data available from other experiments (e.g. the CMUMMAC database [60]). For each sensor three axis acceleration and angular rates were recorded, with a sampling rate of 120 Hz, resulting in an overall datastream of $5 \times 6 = 30$ signals.

Preprocessing As typical in the domain of analysis of wearable data, the data stream has to be preprocessed, following the standard pipeline as for instance outlined by Bulling et al. [40]. At first, a window-based segmentation with a window size of 128 samples and an overlap of 75% was performed for each signal of the stream, resulting in a frame rate of 3.75 Hz. Secondly, for each signal, the following six features were extracted: (1.) mean (2.) variance, (3.) skewness, (4.) kurtosis, (5.) peak, and (6.) energy. Result of this step was a $6 \times 30 = 180$ dimensional feature space at a frame rate of 3.75 Hz. Afterwards, a principal component analysis was applied to reduce the number of dimensions of the feature space. A k dimensional observation space was then constructed by selecting the k dimensions of the transformed feature space with the largest eigenvalues. The values of k were selected by Fibonacci probing.

The requirement for independent and identically distributed observations was not satisfied. As described in Section 3.1.5, this can result in inferior recognition rates. To estimate the influence of this effect, a scrambling of observation data, was used. The dataset that was result of the scrambling operation was published in [134].

Observation model Oko The observation model $p(z \mid a)$ was then constructed by use of this k dimensional observation space. For sake of simplicity, all actions a of an action class $c = class(a)$ were assumed to share the same observation model, resulting in $p(z \mid a) := p(z \mid c)$. For each action class, the observation model distributions were represented as multivariate normal distributions (k dimensions) with unconstrained covariance matrix Σ_c, $p(z \mid c) := N(z \mid \mu_c, \Sigma_c)$. The parameters μ_c and Σ_c were computed by taking all observations into account that were annotated with action class c. Leave-one-out cross validation was omitted: for each action class, the observation data of all subjects was considered for observation model construction (see Section 5.2 for a brief discussion).

Observation model OL With respect to research question RQ7 (the use of different observation models), a second observation model was constructed. While the IMU-based observation model exploits the Z and A components of the statistical model (see Section 3.1), objective of the second observation model was to exploit the W and S components, allowing to demonstrate the reuse of the causal model with different observation models.

A location-based model was constructed by extracting the location of the participant and the food from the aLTS model by stepwise execution. An overview of the locations of each domain object is given in Table B.4. The conditional probabilities were set according to:

$$p(y \mid x) = \begin{cases} 1, & \text{if observed locations match locations in } x \\ 10^{-6}, & \text{otherwise} \end{cases} \tag{6.2}$$

Clearly, the second observation model provides more precise information than sensor-based observations, as they are always correct with respect to the annotation. However, as the second model was constructed only for demonstrating the capability of reuse regarding observation model exchange (R5.2), the use was considered legitimate.

6.2.3. Experimental Setup

Baseline classifier As in the first experiment, an HMM was used as baseline classifier. In contrast, as all subjects follow the same goal, a standard HMM was chosen for this experiment. For each action class, an HMM state was created, resulting in 16 states. The transition matrix was, similar to the first experiment, computed by supervised learning of the annotation sequence. A quadratic discriminate analysis (QDA) was selected to provide an additional baseline model, without temporal information. The QDA was computed based on the multivariate normal distributions used as observation model and the prior class probabilities.

Causal behaviour model The causal behaviour model was constructed by applying the development process described in Section 5.3. The iLTS was developed based on the aLTS, which was created by applying the annotation process described in Chapter 4. During the annotation process, 16 action classes and 18 entities were identified. The action classes are given in Figure B.2, the entities in Table B.5. In all, 82 ground actions have been identified to be used during the annotation. The annotations for subject S1 are provided in Table B.7, to exemplify the complexity of the task.

Model characteristics The resulting model consists of 99 ground actions, generated from 44 action schemata of 16 action classes. 18 state features were used to model 14 domain objects and their properties. The median branching factor was $\tilde{b} = 5$ (with interquartile range $IQR = 3 - 6$). The exhaustive exploration of the complete state space revealed 1.47×10^8 states. The minimal number of actions to be sequentially applied to the initial state in order to reach the goal state is 44. The maximal number of unique actions to be applied was 66.

Duration model Experiment X2 is based on smaller set of action sequence samples as Experiment X1, which means that also the number of duration samples per action is smaller than for the other experiments. For this reason, it was assumed that all actions sharing the same action class also share the same action duration distribution. Thus, the action duration model was created based on the action class, rather than based on the specific action. Two different models were selected to determine the effect of possible infinite support on the inference complexity and the resulting recognition performance.

The empirical model was created by counting the number of steps per action instance from the annotation sequence. To avoid overfitting due to the low number of samples, the empirical distribution was created by considering only unique duration samples.

The parametric model was created by applying the process described in Section 5.3. From the list of durations to be tested, ten classes were found, four Weibull, two gamma and four lognormal distributions were fitted. The remaining six classes were found to be fit by a common lognormal distribution. An overview of the selected distributions is given in Table B.8.

Action selection model To detect the effect of the action selection heuristics on the recognition performance, different goal driven heuristics were selected:

f_δ: the complete goal distance
f_h: the recipe distance
$f_{\bar{\delta}}$: the restricted goal distance

For a detailed description of these action selection features, see Section 3.1.4.

To investigate the impact of each single heuristic including their weight values (see Section 3.1.4) different weight values were tested. The weight values to be used were created by exponential probing $\lambda_k = -(2^k), k \in \{0, \ldots, 4\}$. The value of $\lambda_k = 0$ effectively disabled the action selection heuristics resulting in uniform action selection.

Reusability aspect R5.2 As discussed in Section 6.2.1, Experiment X2 targets the demonstration of the reusability of the causal model with different observation models, each using a different variable of the statistical model. For this reason, two different observation models were constructed. The observation model Oko employs sensor data from wearable sensors and thus provides observations of actions by using the variable Z of the statistical model. The observation model OL, provides information about the location of entities. It provides observations of states by using the variable W of the statistical model. The proof of R5.2 is done by demonstrating the use of both observation models without any change to the causal behaviour model.

Experimental procedure To investigate the influence of the modelling factors to the overall recognition performance, several modelling factors were subject to variation. Before evaluating the research hypotheses, a reduction of configurations was strived by selecting the relevant observation models. For model comparison, and thus the evaluation of research hypothesis H.X2.1, using the best parameter set would be sufficient. Developing an understanding for the influence (answering H.X2.2), however, requires systematic parameter change and thus, a multi-factorial experimental design. An overview of all factors, including possible values, is given in Table 6.7. The following factors were subject to change:

1 Model: describes which kind of model is used to represent temporal relations: QDA, HMM, or CCBM.
2 Target: describes the mode of filtering: forward filtering, smoothing, or MAP-estimate.
3 Mode: describes the filtering algorithm used for CCBMs: MF or PF

Factor	Level	Comment
Target	f	filtering distribution $p(x_t \mid y_{1:t})$
	s	smoothing distribution $p(x_t \mid y_{1:T})$
	v	MAP-sequence $x_{1:T}^{MAP} := \arg\max_{x_{1:T}} p(x_{1:T} \mid y_{1:T})$
Model	QDA	(no system model)
	HMM	HMM transition matrix
	C	CCBM model
Mode	M	Marginal filter
	P	Particle filter
Observations	Oko	IMU data using $k \in \{5, 8, 13, 21\}$ principal components
	Oks	IMU data, scrambled
	OL	Locations (categorical)
Distance	f_δ	True goal distance, complete state space
	$f_{\bar{\delta}}$	True goal distance, restricted state space
	f_h	Heuristic goal distance, using recipe
Weight	Lλ	$\lambda \in \{0, 1, 2, 4, 8, 16\}$
Duration	τ_c	continuous parametric duration models
	τ_d	discrete duration models based on empirical distribution function

Table 6.7.: Factors and levels for experimental configurations.

4 Observations:gives the applied sensor model: location-based or IMU (original or scrambled) with varying k.
5 Distance: gives the goal-directed action selection heuristic: exact, recipe or core.
6 Weight: gives the value of λ_i: $\lambda_i = -(2^k), k \in \{0, \ldots, 4\}$
7 Duration: represents the duration model: parametric or empirical.

From this list the factor Method is created by combining Target, Model, and Mode. The values of Method are: QDA, HMMf, HMMs, CMf, CMs, CMv, CPf, where for instance "HMMs" means an HMM model with smoothing distribution as estimation target, "CPf" a CCBM with PF and forward filtering distribution as target, and "CMv" a CCBM with target MAP-sequence computed using the Viterbi algorithm. Finally, the factor "Subject" with levels "Si", where $i \in 1 \ldots 7$, represents the seven datasets being available for experiments.

From the set of potential factor combinations, the following configurations were selected:

$(\mathsf{QDA}, \mathsf{O}k\mathsf{o})$: Baseline evaluation that serves as demonstration of the discriminative power of the observations without incorporating temporal relations.
(4 configurations)

$(\mathsf{HMM}\{\mathsf{f}, \mathsf{s}\}, \mathsf{O}k\{\mathsf{o}, \mathsf{s}\})$: Baseline temporal analysis to investigate the impact of temporal relation, different observation models and scrambling. Observation models for further analysis were selected based on the results.
($2 \times 4 \times 2 = 16$ configurations)

$(\{\mathsf{CMf}, \mathsf{CMs}, \mathsf{CMv}, \mathsf{CPf}\}, \{\mathsf{O5s}, \mathsf{O21s}, \mathsf{OL}\}, f_{\{\delta, \bar{\delta}, h\}}, \mathsf{L}\lambda, \tau_{\{c,d\}})$: Complete CCBM analysis to evaluate the influence of different configurations on the overall performance. From the HMM analysis the scrambled observation models O5s and O21s were selected for analysis. OL described the location-based observation model. As in the first experiment, comparison of the different values of Mode is only based on forward filtering.
($4 \times 3 \times 3 \times 6 \times 2 = 432$ configurations).

As all configurations were applied to the data from each of the seven participants, the overall number of configurations is $(432 + 16 + 4) \times 7 = 3164$.

Experiment execution The analysis of the influence of the random seed to the overall recognition accuracy of the PF in Experiment X1 showed no significant influence. For this purpose, also for Experiment X2 a prior analysis of the influence was conducted. Depending on the results of this analysis, it is decided whether the application of multiple runs in order to compensate random seed based performance variations can be omitted or not. The analysis is done on the configuration $(\mathsf{CPf}, \mathsf{O21s}, f_\delta, \mathsf{L1}, \tau_c)$ with 20 different random seeds.

Concerning the evaluation of the different configurations, the methods discussed in Section 5.4 were applied. Thus, performance comparisons were based on the accuracy with respect to the action class. The target action class was preferred to the more expressive target ground action for two reasons. Firstly, by using observations that were computed with respect to the action class, the QDA would, due to missing temporal information, not be able to disambiguate ground actions sharing the same action class. Secondly, in order to provide the HMMs a sufficient amount of training data, states were selected to represent action classes. Consequently, the HMM would also not be able to distinguish different ground actions of the same action class.

In addition to the action class, the estimation of contextual information was targeted. Since the annotation process established in Chapter 4 provides a database allowing the extraction of such context information, the ground truth for three different context information was used. By an analysis of the application domain, the following three situations were identified as being of potential interest in a real world scenario:

- "Danger" – a potentially dangerous situation exist (i. e., the stove is on): $\mathsf{on}(\mathsf{stove})$.
- "Eaten" – the protagonist has finished the dinner: $\neg\mathsf{hungry} \wedge \neg\mathsf{thirsty}$.
- "Success" – the protagonist has finished the complete routine (and may be engaged with additional cleanup): $\neg\mathsf{hungry} \wedge \neg\mathsf{thirsty} \wedge \mathsf{clean}(\mathsf{plate}) \wedge \mathsf{clean}(\mathsf{glass})$.

6.2.4. Results

This section presents the results of the analysis of the Experiment X2. Each of the following sections targets one of the two research hypotheses H.X2.1 and H.X2.2. First, the performance of the baseline models is reported and afterwards compared to the results of the CCBM. Based on this analysis, hypothesis H.X2.1 is accepted. In the second step, the results of all configurations are presented and subjected to an analysis of influence of the different factors.

6.2.4.1. H.X2.1 – Model Comparison Results

In the following, the results of the baseline classifiers (QDA and HMM) are presented and compared. Afterwards, the results of the CCBM-based configuration are described and compared to those of the baseline classifiers. If not stated differently, performance is denoted as accuracy (see Section 5.4 for further explanation).

Baseline performance Regarding the baseline classifiers, the best mean[2] accuracy of the QDA was .64 with confidence interval ($CI_{.95} = .6, .68$). HMM forward filtering (HMMf) achieved as best mean accuracy .66 ($CI_{.95} = .62, .71$) with original observations. The use of scrambled observation resulted an accuracy of .73 ($CI_{.95} = .69, .77$), an increase of 7pp. For all baseline configurations, the best results were achieved with O21, by use selecting $k = 21$ principal components. Additionally, HMM smoothing consistently outperformed HMM forward filtering with a mean increase of 3.98pp (paired t-test, $t_{(55)} = 11.6$, $p < .001$). This difference was increased for scrambled data, resulting in an increase of 5.57pp ($t_{(27)} = 15.9$, $p < .001$). While

[2] The mean accuracy is reported here, as the distribution of accuracies was found to be normal by applying the Shapiro-Wilk test of normality.

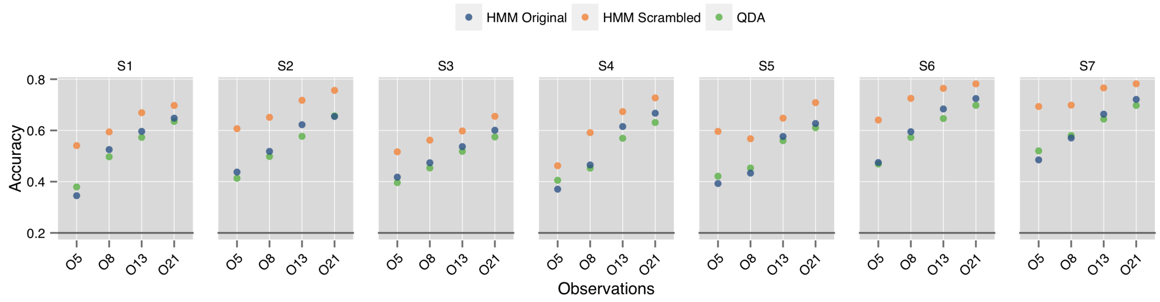

Figure 6.10.: Accuracies of the baseline classifier per subject. Different numbers of principal components in the observation model have been used. An accuracy of .2 (solid grey line) is achieved by selecting the action class with highest prior probability.

the effect for original was only small (Cohen's d =-.49), the effect for scrambled data was found to be large (d =-.94). The results of all baseline classifiers for all configurations are depicted in Figure 6.10.

The use of scrambled observations for HMM forward filtering consistently outperformed the use of original observations for all numbers of principal components k. In median, an increase of 9.89pp (Wilcoxon signed rank test, $V_{(28)} = 406$, $p < .001$, Vargha-Delaney $A = .23$) could be observed. While the accuracy of HMM forward filtering compared to QDA improved by 11.7pp for scrambled observation ($V_{(28)} = 406$, $p < .001$, $A = .18$), the increase was only 1.67pp ($V_{(28)} = 302$, $p = .023$, $A = .46$) for original observations. As discussed in Section 3.1.5, this difference is due to the violation of the assumption of independence of the observation. The observation probability depends on the position of the observation within the process of the action. Figure B.4 illustrates this difference for original and scrambled observations. Consequently, further analysis of CCBM performance focused on scrambled data.

Finally, with respect to the number of principal components k, it was found that the models using $k = 21$ consistently performed best (see Figure 6.10). In contrast, the models based on $k = 5$ principal components had the lowest performance. Accordingly, further analysis of CCBM results was restricted to these extremes.

Influence of the random number generator To assess the influence of the random number generator on the recognition performance of the PF, 20 different random seeds were used for initialisation. The variance of the accuracies that was introduced by the seed is $2.23 * 10^{-3}$ ($IQR = 8.32 * 10^{-4} - 6.92 * 10^{-3}$). The variance that was introduced by the different subjects is $1.51 * 10^{-2}$ ($IQR = 1.51 * 10^{-2} - 1.51 * 10^{-2}$). The variance of the accuracies of the different subjects is significant ($W_{(20)} = 143$, p=< .001) larger (A=.97) than the variance of the different seeds. Consequently, the compensation of the influence of the random number generator was considered as unnecessary and was therefore omitted for the further analysis.

H.X2.1 – Comparison between CCBM and baseline An overview of the results of all CCBM configurations is provided in Figure 6.11.

Configuration selection for comparison For each configuration, resulting from the combinations of (Mode ={CMf,CMs,CMv,CPf}) × (Observations = {O5s, O21s, OL}) × (Distance = $\{f_\delta, f_{\bar{\delta}}, f_h\}$) × (Weight = {L0, L1, L2, L4, L8, L16}) × (Duration = $\{\tau_c, \tau_d\}$), the results of the

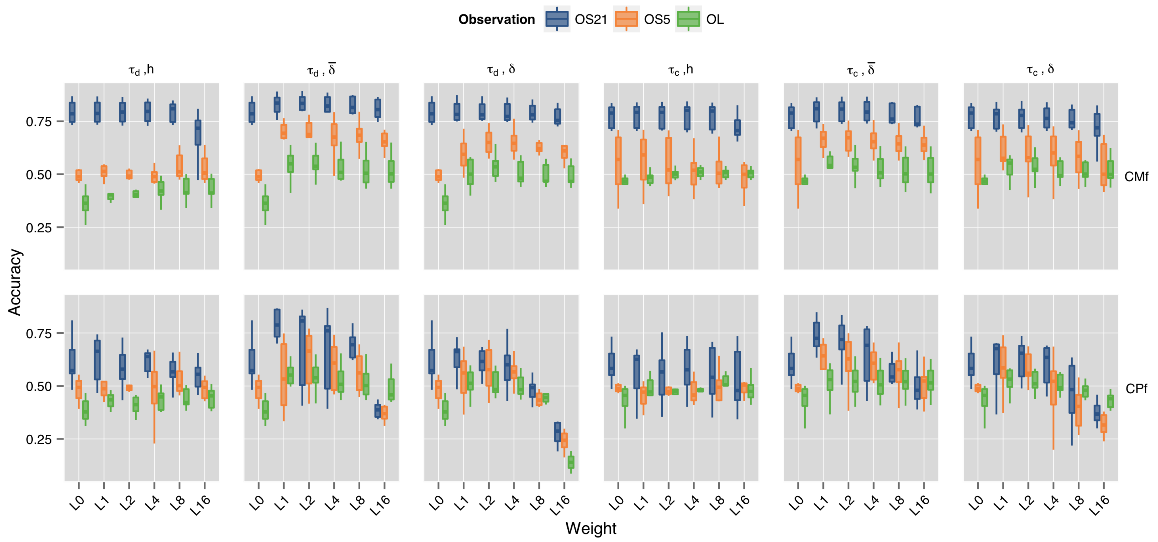

Figure 6.11.: Boxplots of the accuracies of all CCBM configuration for the kitchen experiment.

seven subjects are summarised. From the boxplots in Figure 6.11, influences of single parameter combinations (e.g. CMf consistently outperformed CPf) can be seen. A detailed analysis of these influences if provided in Section 6.2.4.2. From this complete set, all configurations based on (Mode = {CMf, CMs}) × (Observations = {O21s}) × (Weight = {L1}) × (Duration = $\{\tau_c\}$), were selected for comparison with baseline classifiers. These configurations were selected for using the least information from the training data (e.g. τ_c instead of τ_d), employ the least restrictions to the state space (e.g. f_δ instead of $f_{\bar{\delta}}$) and utilise the most rational action selection (e.g. L1 instead of L0, see [17, 188]) from the overall set of configurations. As discussed in Section 6.2.3, using the best single configuration in order to prove hypothesis H.X2.1 is considered sufficient. Here, we focus on both, forward filtering (CMf) and smoothing (CMs).

Testing hypothesis H.X2.1 With respect to forward filtering, CMf showed a significant increase of 3.63pp ($V_{(7)} = 27$, $p = .031$) in median when compared to HMMf. This represents a medium effect ($A = .27$). Furthermore, CMs exhibits an increase of 6.78pp ($V_{(7)} = 28$, $p = .016$, $A = .2$) in contrast to HMMs.

Both results support the hypothesis H.X2.1. Moreover, they suggest that with the use of a suitable parameter configuration, CCBM allows to improve the recognition performance with respect to baseline classifiers.

A detailed comparison of other configurations based on Mode = {CMf, CMs} is listed in Table B.3. In addition, to the overall performance, the performances of the single action classes are shown in Figure 6.13 and Figure B.5. There was no significant difference in the overall per-class performance of QDA, HMM, and CCBM. Furthermore, it can be seen from these figures that the three action classes TAKE, PUT, WAIT, provide problems to the recognition in general.

Cohen's κ An analysis of Cohen's κ for all Modes gives additional evidence for the superiority of CCBM to the baseline classifiers. The results show highly significant difference in κ. Table 6.8 lists the κ values, and the differences for selected configurations. Note that, according to Landis and Koch [138], the κ statistics of CMs ($\kappa \geq .81$) signals "almost per-

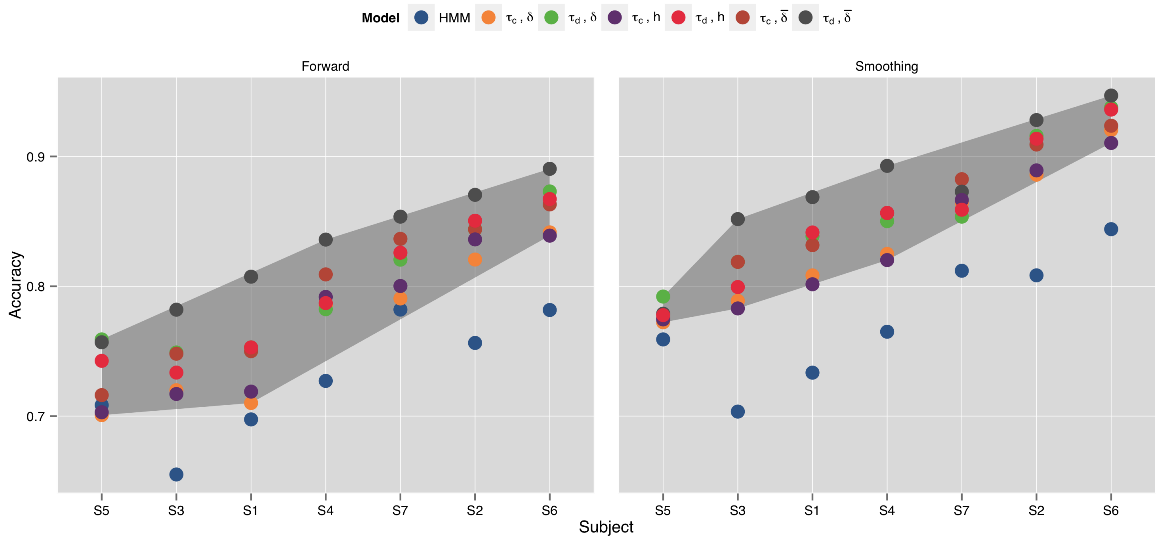

Figure 6.12.: Accuracy comparison of selected CCBM configurations to HMM, by subject and filter method. The CCBM configurations use (CM,O21s,L1). The grey area highlights the hull of the CCBM based configurations. (Subjects are sorted by median performance in all configurations.)

fect" agreement, while the QDA ($\kappa \leq .6$) results in "moderate" agreement, only. The other configurations (HMMf, CMf, and HMMs), ($.61 \leq \kappa \leq .8$) achieve "substantial" agreement.

6.2.4.2. H.X2.2 – Configuration Factor Analysis

After having presented the results regarding H.X2.1 by comparing CCBM to baseline classifiers, in the following, the effect of different parameters of CCBM on the overall performance is analysed. First, a general overview of the influences and parameter interactions is provided. These observation are later detailed with respect to Mode and Distance.

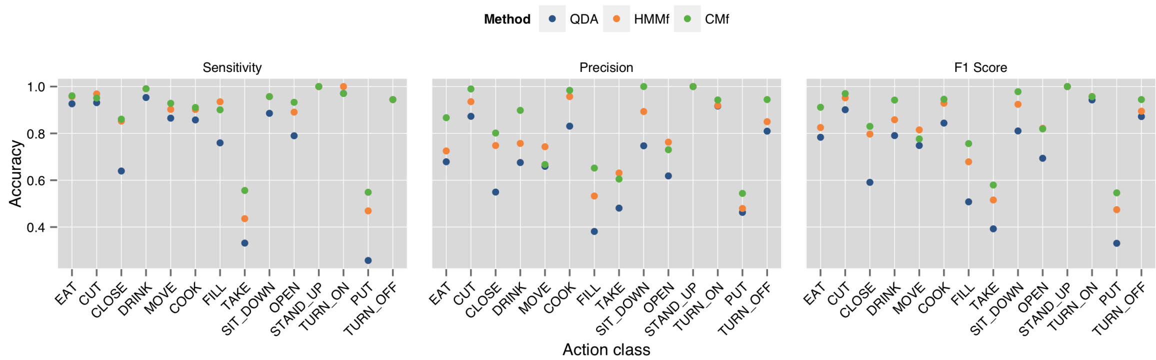

Figure 6.13.: Confusion matrix based performance measures per class. Detailed accuracies for the configuration (OS21s, f_δ, L1, τ_c).

	κ	CM − HMM	Accuracy	CM − HMM
QDA	.6		.65	
HMMf	.7		.73	
CMf	.74	.042	.77	.037
HMMs	.75		.78	
CMs	.82	.072	.84	.064

Table 6.8.: Cohen's κ and overall accuracies for selected configurations. CM − HMM represent the difference between κ and Accuracy values, respectively, for CCBM and corresponding HMM ($n = 6647$, $p < .001$ in both cases). The configuration (O21s,f_δ,L1,τ_c) was used for CCBM.

H.X2.2: Configuration factor effects Figure 6.11 suggests that several effects of CCBM configuration factors on Accuracy are present. In the following, an analysis of these effects using rANOVA is presented. Table B.9 provides a complete overview of the results of the rANOVA. Effect sizes were determined by the generalised Eta-squared effect size measure [175]. Here, the discussion focusses on significant effects of at least medium size, $\eta^2_G \geq .0588$ (see Table 5.3)

Main effects Significant main effects (all $p < .001$) for Observations ($\eta^2_G = .61$), Mode ($\eta^2_G = .34$), Weight ($\eta^2_G = .18$), and Distance ($\eta^2_G = .14$) were found. No significant effect for Duration has been observed. The strongest effect was found for Observation, which caused an increase of 20.5pp ($CI_{.95} = 12.9, 28$) from OL to O21s (Cohen's d =-3.63). Employing the MF (mode marginal) increased the Accuracy by 9.81pp ($CI_{.95} = 7.96, 11.7$) in contrast to the PF (d =-2.05). The factor Distance caused an increase of 4.99pp ($CI_{.95} = 2.58, 7.4$) from f_δ (Complete) to $f_{\bar{\delta}}$ (Restricted) (d =-.98). The use of f_h (Script), in contrast, resulted in an decrease of the Accuracy by 1.57pp ($CI_{.95}$ = -.55, 3.68) ($d = .34$). A moderate nonzero Weight improved the recognition performance, the difference between L0 and L2 was 8.41pp ($CI_{.95} = 6.07, 10.8$) (d =-1.21). The negligible difference ($d = .004$) between L1 and L2 was not significant ($t_{(6)}$ =-.049, $p = .96$). Interestingly, a significant drop from L2 to L16 by 8.39pp ($CI_{.95} = 5.78, 11$) ($d = 1.69$) was observed.

Interaction effects With respect to interactions of factors, the following significant (all $p < .001$) effects were found: Mode × Observations ($\eta^2_G = .24$), Distance × Weight ($\eta^2_G = .1$), Mode × Weight ($\eta^2_G = .089$), and Observations × Weight ($\eta^2_G = .079$). A graphical representation of these effects is given in Figure 6.14.

The Marginal mode gained more from better observations than the Particle mode. While there was only a small (d =-.32) difference in the Accuracy at OL of 1.55pp ($CI_{.95} = .54, 2.57$) (although significant ($t_{(6)} = 3.75, p = .01$)), the Marginal mode clearly exceeded (d =-2.96) the Particle mode at O21s by 20pp ($CI_{.95} = 15.7, 24.3$).

With respect to Distance there was a significant difference between L0 to L1 for both Complete (d =-1.41) and Restricted (d =-2.24) with an increase of 6.5pp ($CI_{.95} = 3.97, 9.02$) and 11.3pp ($CI_{.95} = 8.28, 14.4$), respectively. The same tendency for Script (d =-.12) was not significant ($t_{(6)} = .8, p = .45$). Increasing the Weight further to L16 caused a large ($d = 3.19$) performance decrease of 15.2pp ($CI_{.95} = 13.3, 17.1$) for Complete. The same tendency ($d = 1.44$) could also be observed for $f_{\bar{\delta}}$ (Restricted), with a decrease of 8.69pp ($CI_{.95} = 4.87, 12.5$).

The main reason for this tendency is the interaction of Particle and Complete, which caused a large ($d = 5.94$) decrease of 26.7pp ($CI_{.95} = 23.4, 30.1$) for L16. For Restricted a large ($d = 1.44$) decrease of 15.4pp ($CI_{.95} = 8.43, 22.4$) for L16) was observed when using Particle. For Marginal the medium ($d = .65$) effect was significant only for L16 ($t_{(6)} = 5.76$, $p = .001$) for Complete.

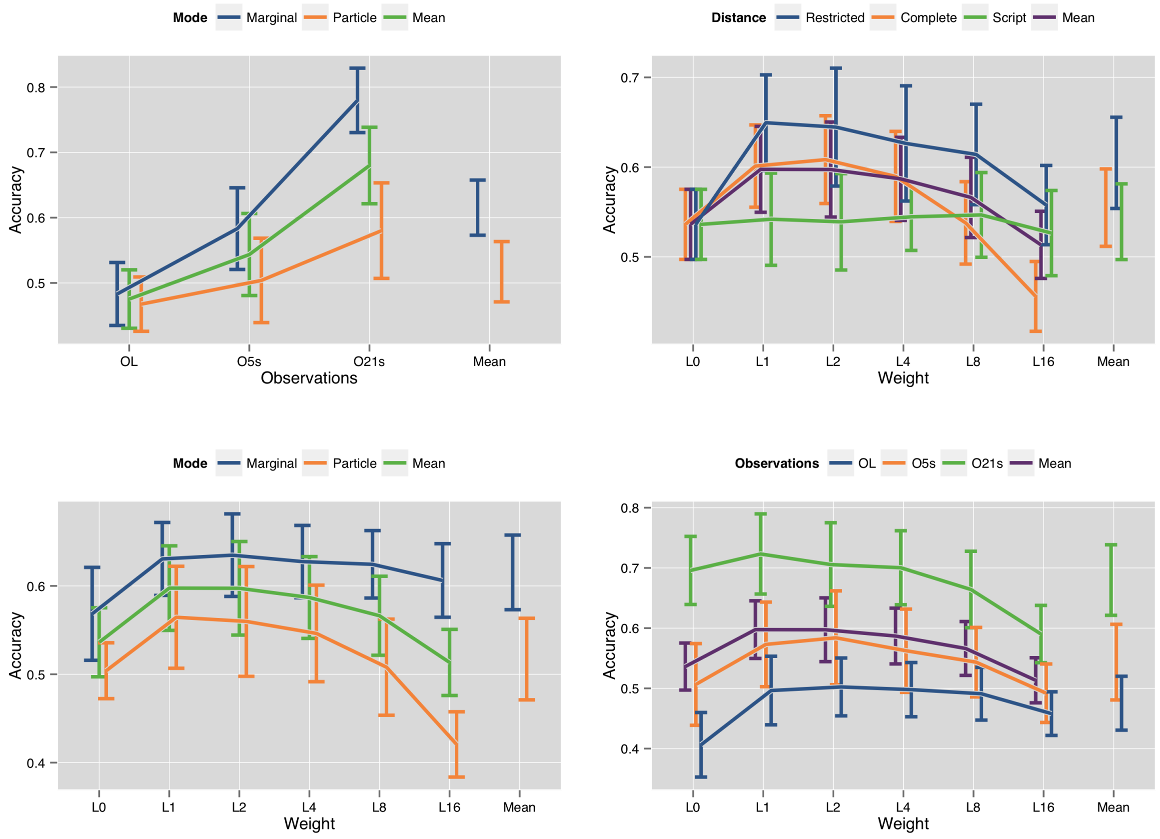

Figure 6.14.: Interaction plots for the significant interactions. Error bars give the 95% confidence intervals due to *between subject* variance. Effect comparisons are based on *within subject* differences.

Only small and non-significant effects ($d = .17, .21, .4$ for L4, L8, and L16) were observed for for Restricted ($p = .06, .49, .21$). The effect ($\eta^2_G = .077$) of the interaction Mode $\times$ Distance $\times$ Weight is also illustrated in Figure B.6

The effect of Script is, as can be seen from Figure B.7, only significant in the interaction of Observations $\times$ Distance $\times$ Weight ($p = .001, \eta^2_G = .018$). For O21s, Script had only a large and significant effect for L16 ($d = .12, .29, .098, .44, 1.09$, $p = .55, .13, .70, .06, .03$ for all five non-zero weights). For OL, in contrast, Script had only a medium (d =-.52) benefit, giving an increase of 3.04pp ($CI_{.95} = .66, 5.42$), at L1 and medium to large effects (d =-.65,-1.14 for L2 and L16), which increases to up to 5.58pp ($CI_{.95} = 2.98, 8.19$) at L16. All effects were found to be significant ($p = .020, .003, .001, .001, .002$ for the non-zero weights). It could thus be observed that with weaker observation models even less perfect distance models begin to show a positive effect.

6.2.4.3. Understanding the effect of Mode

One explanation for the superiority of the Marginal mode is that the MF is able to maintain more states than the PF, as it represents state probabilities by weights rather than sample counts (cp. Section 3.2). To analyse this, the number of LTS states (elements of S) as well as the number of inference states (elements of X) were counted for each step in each filter run. The

	CMf	CPf	ratio (CMf/CPf)
SpU	0.10	0.001	83.18
XpU	1.00	0.008	127.15
XpS	9.77	6.947	1.42
#S	15000.00	786.500	18.42
#X	1090.50	106.750	10.43

Table 6.9.: Median SpU and XpS values and ratios, across all runs of the kitchen experiment.

numbers obtained were compared with the number of representation units (n_U) available to the respective filter, giving the quantity "LTS state per representation unit" (SpU) and "inference state per representation unit" (XpU).

Table 6.9 gives the median values across all runs. As can be seen, the MF clearly makes much better use of the available representation resources. The numbers show the MF to be 80 to 125 times more efficient than the PF considering representation unit use. Concerning XpU, the ratio is always 1:1 for the MF. The row "XpS" gives the number of inference states per LTS state. The MF is able to maintain more inference states (more variations in starting times and action under execution) per LTS state. #S and #X give the median values for the absolute numbers of states. (Note that the PF has been used with n_U^P = 100,000, while the MF used n_U^M = 10,000 and n_U^M = 20,000.)

6.2.4.4. Understanding the effect of Distance

To understand the effect of the Distance on the overall recognition Accuracy, linear models were fitted to predict the relative remaining time RT of a state from different goal distances ($f_{\bar{\delta}}$, f_δ, f_h). The relative remaining time for a state x_t, observed at time t of the observation sequence of length T was thereby computed as $RT(x_t) := 1 - t/T$. Furthermore, the goal distances were normalised to the interval $[0, 1]$. The state sequences were generated based on the ground truth sequences and a stepwise execution of the model. Figure B.3 provides a graphical overview of the linear models. It can be observed that all models explained a substantial amount of RT variance. The normalised goal distances are highly correlated with the true temporal sequences of the observation sequence. However, while r^2 for all methods was high, they showed a markedly different performance (cp. Section 6.2.4.2). As F tests comparing the residual variances show (column $F_{(958,958)}$ in Table B.6), the Complete and Script models had a significantly higher residual variance than the Restricted method in predicting RT. The difference between Script and Complete also was significant ($F_{(958,958)} = .76, p < .001$). This variance in the residuals seems to be an indicator for the observed effect of distance method on performance.

6.2.4.5. State Predicate Estimation

Concerning the estimation of the selected state predicates "Eaten", "Danger", and "Success", it was observed that median accuracies of .93 or better were achieved. An overview of the median results for configurations selected for comparison, including JSD is given in Table 6.10. Figure 6.16 depicts the estimated state predicate probabilities for the configuration ($\mathsf{CMf}, \mathsf{O21s}, f_\delta, \mathsf{L1}, \tau_c$). In median, forward filtering (CMf) achieved an Accuracy of at least .93, whereas smoothing (CMs) further increased this result to a median Accuracy of .97. For Eaten, even perfect (Accuracy of 1) recognition was achieved for CMs.

In addition to the Accuracy, the use of the JSD was discussed in Section 5.4. The JSD was found to be more sensitive, signalling differences even if Accuracy suggests perfect estimates.

Predicate	Target	Accuracy	IQR	JSD	IQR
Eaten	Forward	.93	(.92 – .95)	.021	(.019 – .026)
Danger	Forward	.99	(.98 – .99)	.017	(.015 – .023)
Success	Forward	.95	(.93 – .98)	.043	(.028 – .056)
Eaten	Smoothing	1.00	(1.00 – 1.00)	0	(0 – .001)
Danger	Smoothing	1.00	(.96 – 1.00)	.002	(0 – .038)
Success	Smoothing	.97	(.89 – .99)	.033	(.017 – .064)

Table 6.10.: Median values and IQR (Q1,Q3) for Accuracy and JSD for predicate estimation. Estimation is based on configurations (CM, O21s, L1).

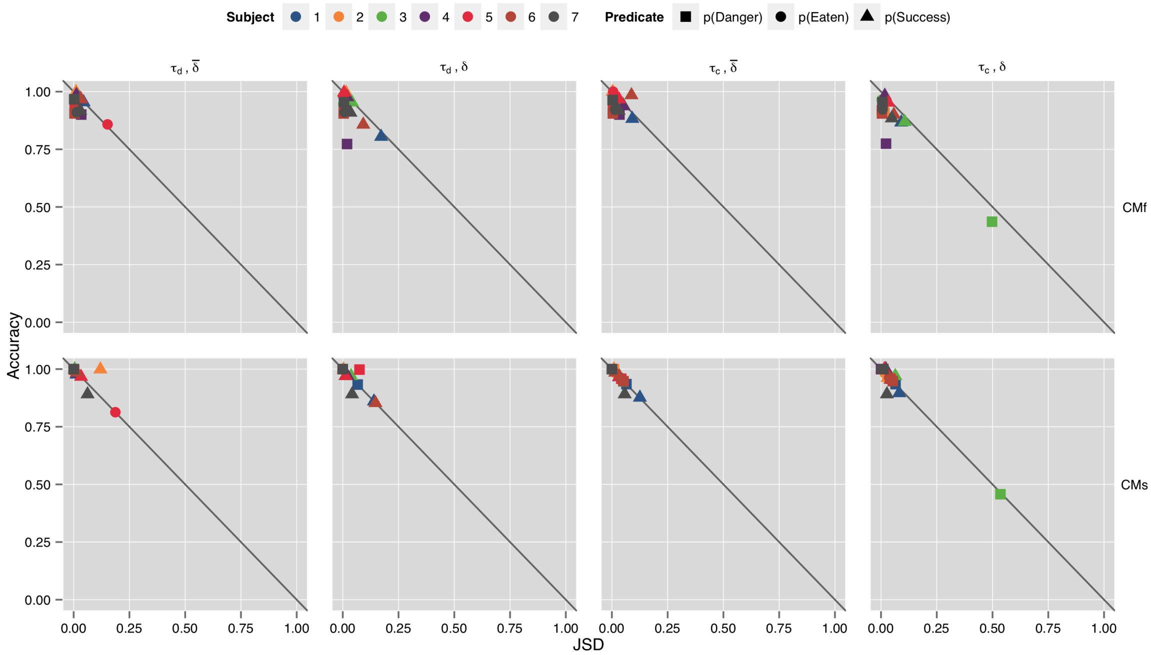

Figure 6.15.: Jensen Shannon distance and accuracies for different values for Target and Distance. Plots are based on the configuration (CM, O21s, L1). The grey line highlights perfect correlation.

This is clearly the result of considering the probability distribution instead of relying on point estimates. Figure 6.15 gives a complete comparison of the state predicate estimation of different configurations (Mode = {CMf, CMs} × Duration = $\{\tau_c, \tau_d\}$, Distance = $\{f_\delta, f_{\bar{\delta}}\}$). A significant correlation (Spearman's $\rho = -.86, S = 1.47 \times 10^6, p < .001$) between JSD and Accuracy could be observed, serving as justification for the use of Accuracy in favour to JSD.

Subject S3 represents an interesting outlier regarding the estimate of the predicate Danger (Figure 6.16). Inference mistakenly estimated the action TURN_ON, resulting in a wrong estimate of the state predicate. This recognition "error" has only low impact on the Accuracy of estimating the action sequence. But the impact of the action estimation error on (1.) the inferred situation and (2.) the subsequent action estimates in contrast, might be high. Figure 6.15 illustrates this effect graphically. Additionally, Figure 6.17 provides a comparison of the subject-specific action sequence estimates with respect to sequence alignment based performance measures. It can be seen in the plot that using measures of performance that are sensitive to the causality of the action sequence, such as the Levenshtein edit distance, reveals such effects.

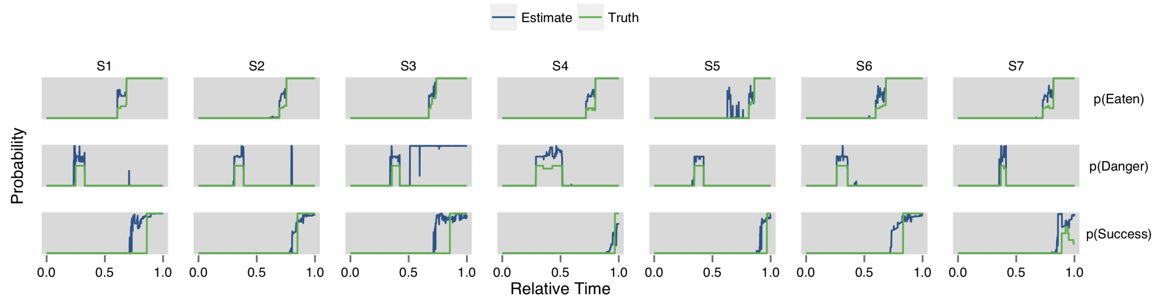

Figure 6.16.: Estimating the probability of state properties. Sample values for the configuration (CMf, O21s, f_δ, L1, τ_c), plotted for each subject individually.

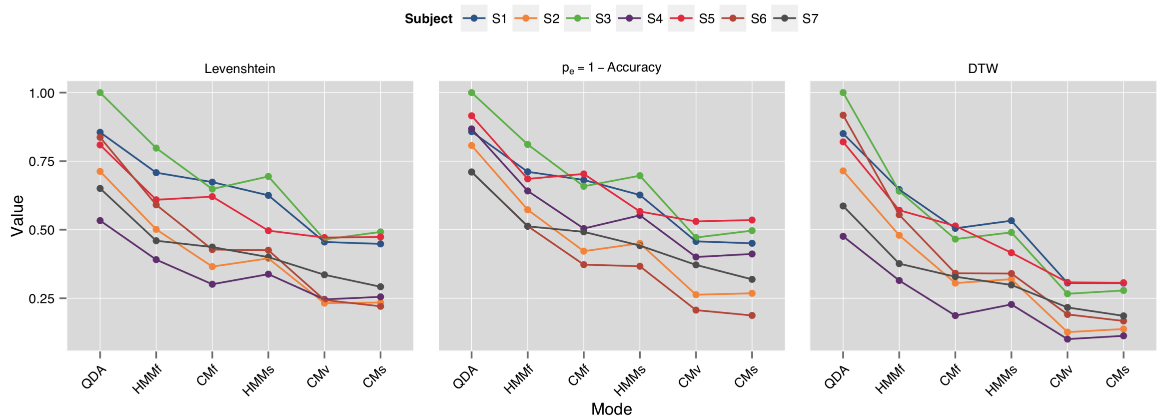

Figure 6.17.: Comparison of different performance measures sensitive for causality. Values are scaled within each measure. The coloured lines emphasise the trend for the different subjects. (smaller means better)

6.2.5. Discussion

Objective of Experiment X2 was to answer the four research questions. For RQ4 and RQ5, hypotheses were constructed that if proven to be true allow to answer the research questions. For the RQ6 and RQ7, it was considered sufficient to provide a proof by demonstration in the first place. While this does not allow to conclude general statements about the capabilities of CCBM, it provides indicators that justify further research on these questions. In the following the results and implications of Experiment X2 are summarised.

Complexity of the causal model The current state of the art on the application of CSSMs focussed on settings with maximal plan length of 20, maximal state space size of 70,000, and the distinction of maximal ten action classes. This experiment, in contrast, was set up to simultaneously exceed all of these limitations. The constructed LTS was found to have 1.47×10^8 (S) states, which increased to 6.23×10^9 inference states (X states) when also considering the start times. At the same time the plan length was increased to 91.6 in mean while focussing on 16 action classes. Thus, the experiment addressed a model that is several orders of magnitude larger than models used in comparable studies.

Hypothesis H.X2.1 Regarding the research question RQ4, CCBM has been compared to baseline models with respect to recognition rates. A QDA, which only relies on the data, and an HMM, featuring also temporal aspects, were used for this comparison. Both were based on training data only, and were allowed to overfit by incorporating the complete dataset in order to prevent a bias towards CCBM. CCBM was found to achieve recognition rates at the same level as the HMM. Moreover, CCBM outperforms the constructed HMMs even though the HMMs were allowed to overfit by training on the complete data. This was also confirmed by the results of the test of hypothesis H.X2.1. Consequently, the first research question which states whether CCBM is able to achieve recognition results at the same level as baseline classifiers, if suitably parameterised, can be confirmed. Although it was not the aim of this study to prove the superiority of CCBM in absolute numbers, the κ statistic for the action sequence estimated by CMs can be interpreted as “almost perfect” agreement (see [138]) to the ground truth. The results show that the use of rich state spaces, created from causal models, albeit increasing state space complexity, do not reduce recognition performance.

Complexity of the baseline model With respect to the HMM as baseline model, it has to be noted that the transition matrix for 16 states has $16 \times 15 = 240$ parameters to adapt to the training data. The CCBM, in contrast, has only $11 \times 2 = 22$ (eleven different duration models with two parameters, resp., see Table B.8) parameters. The 16 parameters that describe the self transitions of the HMM, however, are restricted in their meaning. All of them describe the λ parameter of the geometrical distribution. For the duration model of the CCBM, the best fitting PDF.

Hypothesis H.X2.2 The second research question (R5) concentrates on the influences of single parameters and their interactions. All parameters, except for the duration model, were found to have significant influence on the recognition performance. The test of research hypothesis H.X2.2 affirmed this. This, on the one hand, supports the design decision of providing these configuration capabilities through CCBM parameters and on the other hand suggests further research for each parameter. While the use of a parametric duration model (with possibly infinite support) increases inference complexity in contrast to an empirical one, this replacement had no significant effect on the overall performance. This is an indication for the successful approximation of the observed action durations.

Observations on factor interactions Besides this, three important observation have be made.

1 The use of the MF increased inference performance in contrast to the PF. This is due to the weight-based representation of states instead of sample-based representation. As can be seen from the analysis of related work, all CSSM related approaches use variants of the PF for discrete state space and are therefore affected by this issue. In fact, only one of the analysed studies used an approximation particularly tailored for categorical state spaces.

2 Using i.i.d. observations increases the recognition performance notably. As has been shown, sensor observations are typically not i.i.d. (see Figure B.4). For this study, this issue has been addressed by scrambling the observation of the same action class. Obviously, this is not possible in real applications. Consequently, this issue should addressed in further research by hierarchical models, such as RBPF as for instance employed by Bui et al. [37] or Liao et al. [149].

3 By using goal distance based heuristics, recognition performance could be increased. Furthermore, performance increased with the quality of the heuristics. While $f_{\bar{\delta}}$, learned from training data, achieved the best results, f_h, naively created from the experimental

script, achieved worst results. The effect of goal distance based action selected increased with decreasing quality of the observation model. With the best observations (O21s), even f_δ, created from exhaustive exploration, had no effect. All action sequences were longer than the minimal number of steps necessary to reach the goal. Consequently goal distance based heuristics underestimate the length of the remaining action sequence, which is corrected by the observation model. Again, further research on the action selection heuristics is suggested.

RQ6: Using wearable sensors for fine grained AR Research question RQ6 addressed the successful application of sensor data from wearable sensors for fine grained AR. The high recognition rates from wearable sensors by use of CCBM indicates successful application. This qualifies the statement of Chen et al. [46]. Furthermore, the use of OL, which is similar to observations generated from environmental sensors consistently performed worse than observation based on wearable sensors.

RQ7: Reusability aspect R5.2 Regarding the reusability aspect R5.2, a proof by demonstration was done, supporting the initial statement of reusability despite a changed observation model. While the Oks observation models use action observations (the Z component, see Section 3.1), the OL observation model was set up based on state observations (the W component).

As the increase of state space complexity introduced by CCBM has no negative effect, it can be assumed that the effect of introducing further state predicates has only limited effect to the recognition quality. Thus, detailed description of the environment, including involved objects enables the use of a variety of observation models. This includes state variables that are at least at modelling time only of limited interest. The introduction or exchange of new environment sensors allows the application-specific use of observations. The introduction of additional state variables, in turn, extends the amount of context information to be estimated.

Summary In summary, Experiment X2 showed that CCBM allows to successfully reconstruct action sequences from wearable sensors. Thereby, a very large state space was employed. The detection of context situations of potential interest has been illustrated. Furthermore, it has been shown that the parameters introduced to configure the statistical model underlying CCBM have significant influence to the overall recognition performance. This result supports the initial design decisions of introducing these modelling factors to CCBM. Finally, the causal model has been used with different kinds of observations, each exploiting different parts of the X state. The CCBM inference techniques have been shown to successfully cope with the large state spaces created from the causal descriptions, allowing for generalisability with respect to the application domain. CCBM allows the simultaneous recognition of the current action and contextual information in problems of higher complexity as related work. CCBM outperforms baseline classifiers while providing reusability.

6.3. Experiment X3: Indoor Localisation

Experiment X3 is concerned with the localisation of multiple persons within an indoor environment by means of anonymous binary sensors (i.e. PIR sensors). The problem of simultaneous identification and localisation, also known as data association problem [248] or track confusion problem [74], arises when several persons are observed by sensors without any assignment of sensor data to the identity of the person itself. The persons are moving within the indoor environment. Sensors without such assignment are often called anonymous, since they do not allow to conclude the identity of the person being observed from sensor data only. Additionally,

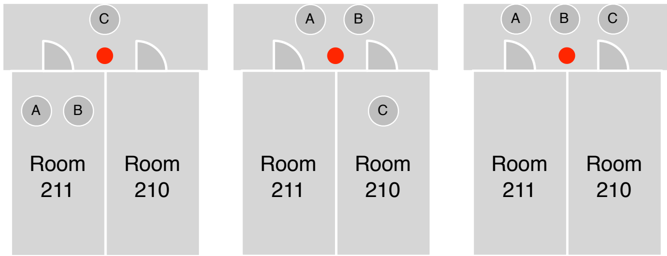

Figure 6.18.: The problem of simultaneous identification of multiple persons in partially observed environments. Only one of three rooms is observed by sensors. The red point represents the sensor. Three persons (A, B, and C) are distributed over these rooms. Three situations are illustrated which can not be distinguished by anonymous binary sensors.

binary sensors, as in the case of PIR sensors, are inherently ambiguous concerning the number of persons being observed, increasing the challenge further. Inferring the identity becomes even more challenging if not the complete environment is observed but only parts of it. This is, for instance, the case in typical office environments, where motion detectors for light control are installed in public accessible areas. The problem of localisation and identification of multiple person in a partially observed environment arises. This problem substantially differs from the problems that were addressed by the literature (i.e. Fox et al. [74] or [248]), as it introduces unobserved regions that allow multiple persons to "hide" from sensor observation. A graphical illustration of the problem is given in Figure 6.18. Imagine three rooms, of which only one is observed by an anonymous binary sensor (e.g. a PIR sensor). Imagine further, three persons (A, B, and C) who are moving within this environment. Due to the capabilities of the sensors and the environment being only partially observed, several situations exist that can not be distinguished. Three of these situations are exemplified in Figure 6.18. Obviously, this problem can not be solved without any specific knowledge about the behaviour of the involved persons. Such knowledge is called identification features. The scenario was chosen for the following reasons:

- Several researchers address problems similar to the simultaneous identification and localisation problem. Work on this topic has for instance been done by Müller and Hein [163], Fox et al. [74], and Wilson and Atkeson [248].
- The literature uses different identification features, such as identifying sensors [74] or person specific behaviour models [248] to solve the track confusion problem. The question here is, if identification features can be provided by the causal behaviour model (e.g. goals or actions specific to persons).
- Anonymous sensors, such as PIR are ubiquitous in public buildings. They are typically installed for light control or similar reasons. Additionally, PIR sensors are unobtrusive [163], allowing, for example, to be installed in office buildings [164] and home and care environments [163].
- Setting up an experiment for this problem is relatively easy, as typical public environments are already instrumented with such sensors, connected via buildings automation bus [164]. This allows results to be easily transferred to real-world applications.
- The setting allows the number of involved persons to be varied easily. An increase (or decrease) of the number of involved persons requires no change to the overall experiment. Specifically, a change to the number of persons does not require changes to the sensor infrastructure or the observation model. This allows other researchers to easily reproduce the results on a larger scale.

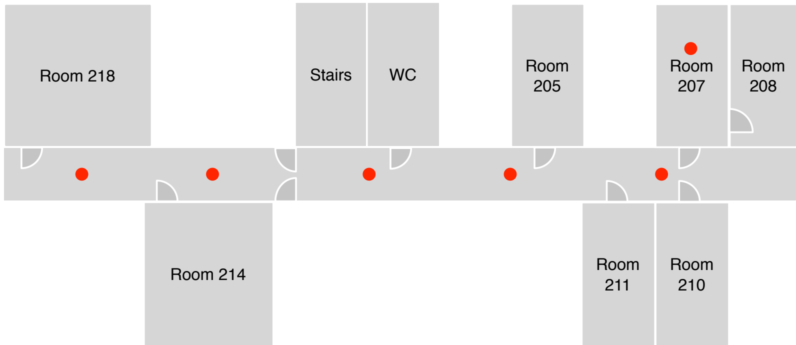

Figure 6.19.: The spatial layout of the indoor localisation trial setting. The red circles represent the PIR sensors, rooms with red circles represent observable areas, consequently.

- The setting allows an investigation of the influence of the number of persons on the state space complexity and thus on the inference complexity.
- The use of anonymous binary sensors allows to increase the complexity of the problem as with each increase of the number of persons the number of possible sensor assignments increases. This allows an investigation of the effect of an increased the state space on the recognition performance without considering potential identification effects of the sensors.

A detailed description of the experiment is given below.

6.3.1. Objective

Experiment X1 and Experiment X2 showed that CCBM allows the reuse of causal behaviour models. For this reason, two aspects of reusability were demonstrated, R5.1 and R5.2. The objective of Experiment X3 is now to investigate the capabilities of CCBM with respect to reusability with varying number of involved protagonists (R5.3). More specific, it is investigated if a causal behaviour model developed for a specific number of protagonists can be reused with a different number of persons. Furthermore, the size of the state space generated by increasing the number of persons is investigated. For this reason, a trial setting is selected were different numbers of agents act independently. A common sensorial observation, which is influenced by each person, is chosen as link between them. This experiment aims at answering the research questions RQ8 and RQ9.

For both research questions rephrasing to research hypotheses has been desisted, as a proof by demonstration was considered sufficient to demonstrate the capabilities with respect to multiple agents. The following sections describe the trial setting, the experimental setup and the results with respect to the research questions.

6.3.2. Trial Setting

In order to answer the targeted research questions, a trial was done. During the trial, different numbers (1–7) of protagonists were instructed to move through an office environment. Anonymous binary sensors were used to observe the environment, each of them signalling if at least one person is moving within the range of the sensor. In the following the trial setting is described in detail.

Trial task For the trial setting, nine rooms on the same floor, which are connected by a corridor, were selected. Figure 6.19 shows the floor plan of the trial. The sensor layout follows a standard instrumentation of office buildings for the purpose of light control [164]. The

person	office	start	goal			
			document	coffee	meeting	location
A1	room 214	stairs	□	■	■	room 214
A2	room 211	stairs	■	□	■	room 211
A3	room 214	room 214	□	■	■	room 214
A4	room 208	room 208	□	■	■	room 208
A5	room 205	room 205	□	□	■	room 205
A6	room 210	room 210	□	■	■	room 210
A7	room 211	room 211	□	■	□	room 211

Table 6.11.: Overview of the different agent configuration.

corridor was partitioned into five observable areas, being monitored by one presence sensor. Furthermore, the public accessible room 207[3] was instrumented with a presence sensor.

Seven participants were considered enough to investigate the effects of a varying number of persons on inference performance. The rationale here was that seven participants are enough to show a trend in the influence on the recognition performance. With respect to the investigation of the reusability aspect R5.3, two different numbers of participants would be enough to perform a proof by demonstration. Consequently, the decision about the number of participants was based on the considerations regarding RQ9

An identifier (A{1–7}) was assigned to each person. Beginning with seven involved persons, the number of persons was decreased by one for each trial run[4]. Each trial run (trial with a fixed number of persons T{1–7}) was repeated five times to increase variance of durations and sensor data. The overall number of trial runs was seven, resulting a an overall number of 35 (7×5) runs.

Each person was instructed to follow a predefined schedule similar to typical office routines but with shortened durations. The following different sub-tasks had to be accomplished: (1.) getting a printed document, (2.) getting a coffee, and (3.) having a meeting. Table 6.11 gives an overview of the person-specific configurations including the starting location, the particular office and the goal. For each trial run, an exception was made for the person with the highest identifier per trial run (e.g. in trial run T6 with six involved persons, the schedule of person six (A6) was adjusted.). The exception is based on the decision that all persons, except the one with the highest identifier are supposed to have a meeting together. The schedule and the derivation were created by a domain expert which was blind to the objective of the trial. The median length of the resulting observation sequences was 473 s (with $IQR = 467 - 478$) for all trials. The median plan length over all trials and agents was 36 ($IQR = 24 - 39$).

The complete experiment was recorded by three static cameras, two positioned in the rightmost and the leftmost corner of the corridor, one positioned in room 207. The resulting video material was used to create the annotation of the scenario using the annotation process described in Chapter 4. During the annotation process ten action classes and five entities have been identified. 27 ground actions have been found while ignoring the executing agent (e.g. walking-corridor1-corridor2). By also considering the executing person in the ground action (e.g. walking-A1-corridor1-corridor2), this number is increased to 112. An excerpt of the annotation sequence for A1 is given in Table B.10. The complete dataset has been made publicly available in [112].

[3]Room 207 was public accessible as it contains a printer and coffee machine that was shared by all persons.

[4]Two different terms are used to distinguish the different iterations of the trial run: 1. Trial run refers the part of the trial that was done in equal settings (e.g. Seven trial runs were done). 2. Repetition refers to the different iterations that were done with equal number of persons (e.g. Five repetition were done for each trial run).

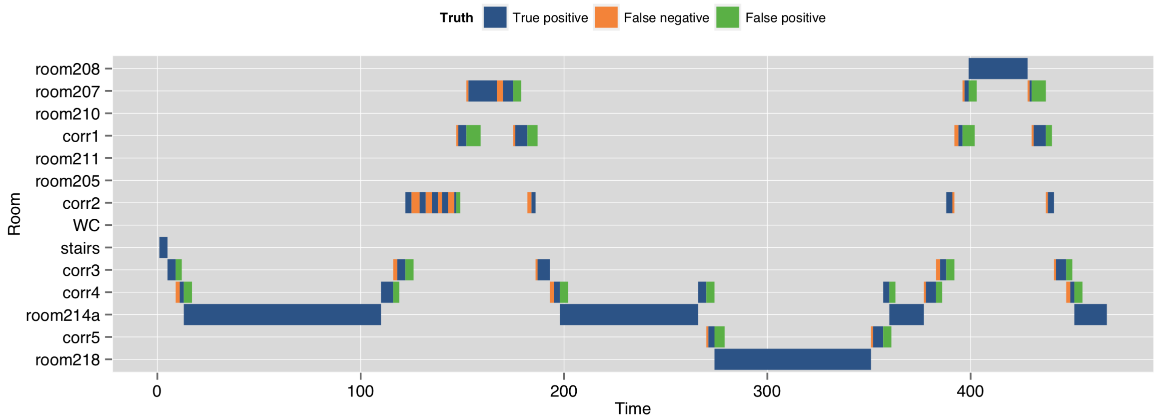

Figure 6.20.: Graphical illustration of the location sequence of the first repetition of person A1 **in trial** T1**.** For each room of the trial setting, it is highlighted whenever the person is in this room. Additionally, the corresponding sensor information is displayed.

Sensor data and preprocessing The PIR sensor nodes, which were developed for this trial, are based on a MSP430 LaunchPad. Each sensor node was equipped with a PIR sensor and an SD memory card to store sensor events. Before each trial run, each sensor node was synchronised with the experimenter's computer in order to ensure time synchronicity. After each trial run, the data was downloaded from the SD memory card. The PIR sensors are able to recognise two different situations. The first being that a person is inside the range of the sensor, the second that there is nobody within the range of the sensor. Each sensor node records changes to the situation.

After the trial, the six sensor data streams were synchronised. The sensor data was converted to an equidistant time base of 1 Hz. The result of the sensor data preprocessing was a data table consisting of six columns – one for each sensor (Y) – with one row for each second of the repetition. In the table, a "1" signals that at least one person is within the range of the sensor node, whereas a "0" indicates that no person is in the range. A graphical illustration of the sensor data for first repetition of the trial run T1 with one participant is provided in Figure 6.20. Three effects can be observed that potentially increase inference complexity: (1.) the sensor typically signals presence of a person too late, (2.) the sensor signals presence of a person too long, and (3.) the sensor's presence signal is interrupted during long sequences of presence (see corridor 2 at time 130).

Observation model The observation model $p(w \mid s)$ was constructed for each observable area, independently. Thereby, a counting mechanism, based on the parameter training of the NB classifier with m-estimate [160, p.179], was employed. For each observable area i a confusion matrix was constructed from the sensor observation of sensor ($Y_i \in \{0, 1\}$) and the annotation for room R ($R_i \in \{0, 1\}$). This training procedure was applied for each trial run. The observation model was created as combination of the observed areas (see equation 6.4).

$$p(w \mid s) = \prod_i p(Y_i{=}y_i \mid R_i{=}r_i) \tag{6.3}$$

$$p(Y_i{=}y_i \mid R_i{=}r_i) = \frac{1 + \#(R_i{=}r_i \land Y_i{=}y_i)}{2 + \#(R_i{=}r_i)} \tag{6.4}$$

6.3.3. Experimental Setup

Baseline classifier Purpose of the baseline classifiers was to estimate the location of each subject for each time-step. The NB classifier was selected as non-temporal baseline classifier. The reason is that according to Mitchell [160, p.177], the NB classifier is well suited for finite sets of observation items, as it is the case with binary sensors. A situation is here represented as a n-tuple, where n gives the number of all involved persons. For each person, the current location (one out of 14 possible rooms) is represented. The prior probabilities were generated by counting situations from the annotations. For the trial run with three involved persons, for instance, a situation would be that A1, A2, and A3 are at corridor1, corridor2, and room205, respectively. Consequently, situations that did not occur during the trial will not be considered during classification. Obviously, the NB classifier is overfitted with respect to the training data.

As for the other experiments, an HMM was selected as temporal baseline classifier. Similar as for the NB classifier, the state space of the HMM was created by considering situations from the annotated trial runs. The resulting number of states for the baseline HMM for one person was 11. For two and more persons the following number of different situations was found: 41, 81, 87, 166, 189, 351. Like the NB classifier, the HMMs are overfitted to the scenario. The prior state distribution was directly taken from the experimental description for each participant. For each trial run, the transition matrix was created by supervised learning of the annotated label sequence. Like for the previous experiments, the transition matrix was not subject to regularisation.

Causal behaviour model The causal behaviour model was created by employing the development process described in Section 5.3. Ten action schemata were created, each of them representing an action class, resulting in the set of ten action classes. Each action schema was parameterised with an executing agent and a location. This would lead to high number of ground actions (ten action schemata $\times$ 14 locations $\times$ number of persons). To reduce the number of actions, only actions that have been executed in the trial were considered. This reduced the number of actions and thereby the number of possible states.

Model characteristics The resulting number of ground actions was 25 for T1 with one involved protagonist. This number increased to 50 (62, 80, 103, 127, 150) for T2 (T{3–7}, resp.). To provide identification features within the causal model, for each agent the information about the location of the corresponding office was provided. Furthermore, for each agent a goal was specified. The goal was directly taken from the experiment description. Finally, an ordered list of working items was created for each agent. The shortest sequence of actions to be executed in order to reach the goal per agent in median was 30 ($IQR = 26.75 - 36.25$).

As the participants were modelled independently from each other, the state space is constructed by combining the possible states for each agent. The resulting state space sizes are provided in Table 6.12. Note that the number of states is, albeit being severely restricted, by far higher than for the baseline classifier.

Duration model For the action duration model, an empirical model was considered sufficient, as it was not aim of the experiment to assess the influence of the duration model. The action duration model was created based on the ground action. Thus, the duration of an action would serve as an additional identification features.

Reusability aspect R5.3 Beside investigations on the effect of the state space size on the recognition performance, Experiment X3 aims at investigating whether a causal behaviour model can be reused with a varying number of persons. For this purpose, a model was developed

Trial	A1	A2	A3	A4	A5	A6	A7	overall	baseline
T1	73	-	-	-	-	-	-	$7 * 10^1$	11
T2	73	73	-	-	-	-	-	$5 * 10^3$	41
T3	73	85	42	-	-	-	-	$3 * 10^5$	81
T4	73	85	48	36	-	-	-	$1 * 10^7$	87
T5	73	85	48	80	56	-	-	$1 * 10^9$	166
T6	73	85	54	80	54	42	-	$6 * 10^{10}$	189
T7	73	85	48	80	54	48	80	$5 * 10^{12}$	351

Table 6.12.: State space size for the different problems. The overall state space size is defined as the product of the state space size of each agent. The most right column gives the number of situation actually appeared during the trial.

that allows an adjustment of the number of agents. Similar as for the experiments X1 and X2, the definition of the actions and the state features was left unchanged while changing the number of persons. This was possible by introducing the executing agent as a parameter for each action (e.g. *walking-corr1-corr2* becomes *walking-A1-corr1-corr2* to reflect that subject AG1 is executing the action). The same technique was applied for state features that describe the current state of the respective protagonist.

To investigate the reusability aspect R5.3, a proof by demonstration was done. By applying the same causal behaviour model to different numbers of involved persons, the LTS, which is generated from the causal behaviour model and their parameters, was adjusted to the actual number of persons. The parameters include a specification of initial state properties such as the initial location and the office, and the final goal of the respective person.

Experimental procedure Due to the large number of possible states and the increased level of freedom, the recognition results were expected to be lower than for the temporal baseline classifier. In order to ensure that this decline in the recognition performance is not caused by CCBM in principle but by the large state space created by the behaviour model and the insufficient action selection, an alternative action selection mechanism was considered. For this reason action selection was based on the transition matrix of the trained HMM. As a consequence, the action duration model was not provided by the empirical duration model but rather by the self transitions of the HMM.

Note that the state in the causal behaviour model does not only consist of the location of the agents as in the HMM transition matrix. As described in Section 3.1, the state also incorporates the executed action and other contextual information, which, in consequence, resulted in a larger state space.

The following factors were considered for performance assessment:

1 Model: describes the kind of model that is used to describe the temporal relations: NB, HMM, and CCBM
2 Transition: describes the way how the transition probabilities were computed: f_δ and f_{HMM}
3 Trial: describes the trial run and thereby the number of participating protagonists: T{1–7}

For all CCBM inference runs the MF was applied. The reason here was that Experiment X2 showed that the MF is superior in contrast to the PF in large state spaces. The Weight for all runs was set to $\lambda_\delta = 1$. Table 6.13 provides an overview of all considered factors including their possible levels. The following combinations of factors were considered:
(NB, T{1–7}): Non-temporal baseline classification with the NB classifier.
(7 configurations)

Factor	Level	Comment
Target	f	filtering distribution $p(x_t \mid y_{1:t})$
Model	NB	Naïve Bayes classifier without temporal relations
	HMM	HMM transition matrix
	C	CCBM model
Mode	M	Marginal filter
Transition	f_δ	True goal distance, complete state space
	f_{HMM}	Transition probabilities of the trained HMM
Weight	Lλ	$\lambda_\delta = 1$
Duration	τ_d	discrete duration models based on empirical distribution function
Trial	1–7	Trial run with fixed number of participating protagonist

Table 6.13.: Factors and levels for the indoor localisation experiment.

(HMMf, T{1–7}): Baseline inference with temporal classifier.
(7 configurations)

(CMf, $\{f_\delta, f_{HMM}\}$, T{1–7}): Analysis of the influence of the selected factors on the recognition performance.
($2 \times 7 = 14$ configurations)

Each configuration was applied to the data of all of the five repetitions in each trial.

Experiment execution Objective of all inference runs was to estimate the location of all of the participating protagonist. Furthermore, the CCBM inference runs were set up to simultaneously estimate the activity for each agent. The evaluation was done in a similar way as for Experiment X1. From the situation-based estimate, an agent-based estimate was created by marginalising over all states with equal location for that agent. Then, for each time-step the most likely location was selected from the agent-based estimate. For each inference run the performance was assessed by using the measures discussed in Section 5.4. Like for the first two experiments, the accuracy was used as primary criterium.

6.3.4. Results

This section presents the results for Experiment X3. First the performance of the baseline classifiers is presented. Afterwards the CCBM-based recognition performance is reported. When reporting the results, we first focus on the estimation of the location of the protagonists. The AR results of the CCBM-based inference runs are reported afterwards.

Baseline classifier The NB achieved a mean recognition accuracy of .6 ($CI_{.95} = .58, .62$) for the trial run T1. For the HMMf the accuracy for T1 was .79 ($CI_{.95} = .78, .8$), which represent a large (Cohen's $d = -14.7$) and significant increase of 19.1 pp (paired t-test, t_4=38.4, p< .001). A similar increase of 20.1 pp (t_4=40.9, p< .001, $d = -20.4$) could also be observed for the second trial run (T2). Also for the remaining trial runs (T{3–7}) large and significant increases of the mean recognition accuracies could be observed (19.4, 18.6, 24.6, 25.1, 25.2). A detailed overview of the results is provided in Table 6.14. A plot of the results is given in Figure 6.21.

CCBM with goal based action selection With respect to (CMf,f_δ) for the trial run T1 a mean accuracy of .85 ($CI_{.95} = .84, .86$) was achieved. This is a significant mean increase of 6.01 pp (t_4=20, p< .001, $d = -8.04$) when compared to HMMf. For the second trial run (T2) a mean accuracy of .79 ($CI_{.95} = .76, .83$) was found. The difference to HMMf was negligible

Trial	NB	HMM	t	p	M	p_{SW}	d
T1	.6	.79	38.4	$< .001$	.19	.45	-14.7
T2	.59	.79	4.9	$< .001$	.2	.69	-2.4
T3	.57	.76	32.4	$< .001$	.19	.99	-16.1
T4	.63	.82	25.9	$< .001$	.19	.073	-14.3
T5	.6	.85	58.9	$< .001$	.25	.23	-25.8
T6	.62	.87	77	$< .001$	.25	.76	-58.7
T7	.6	.85	19.8	$< .001$	.25	.14	-12.9

Table 6.14.: Overview of the mean recognition accuracies of the baseline classifiers. Comparison using paired t-test (all $df = 4$). p_{SW} represents the p-value of the Shapiro-Wilk test for normality. d denotes the effect size according to Cohen's d

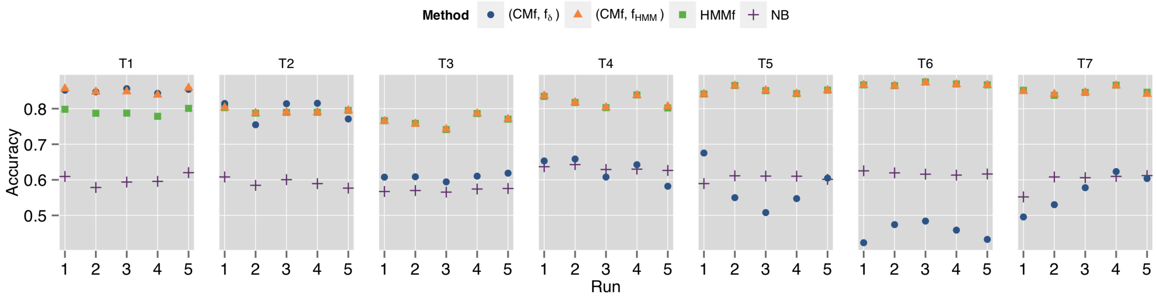

Figure 6.21.: Overview of the recognition accuracies for the location-based estimate of all classifiers. Note that for T{2–7} HMMf and (CMf,f_δ) achieved almost equal results and are therefore hardly distinguishable.

(Cohen's $d = -.041$). The mean difference of T2 to T1 was 5.66 pp. This decreasing trend could further be observed for the trial run T3 (T{4-6}) with differences of -15.7 pp (-19, -27.3, -41.4) in contrast to the HMMf. For T7 the mean decrease of the accuracy was -15.7 pp . All accuracy declines from HMMf to (CMf,f_δ) were large ($d > .8$) and significant with p-value $< .001$. Interestingly, while the mean accuracy for the HMMf showed an increasing trend, the opposite was the case for (CMf,f_δ).

CCBM with HMM transition matrix When considering (CMf,f_{HMM}) a mean accuracy of .85 ($CI_{95} = .84, .86$) could be observed for T1. This represents a significant mean increase to HMMf by 5.93 pp (t_4=93.6, p< .001, $d = -6.97$) but only a negligible ($d = -.13$) difference to (CMf,f_δ) (t_4=$-.35$, p.75). The mean accuracy for T2 (T{3-7}) was .79 ($CI_{95} = .84, .86$) (.76, .82, .85, .87, .85). For T2 no difference between HMMf, (CMf,f_δ), and (CMf,f_{HMM}) was found. With increasing the number of protagonists to 3 (T3), the observed difference of (CMf,f_δ) and (CMf,f_{HMM}) in the mean accuracy increases to 15.6 pp. Further increasing the number of participants resulted in differences of 19.1 pp, 27.3 pp, 41.3 pp, and 28.2 pp. Similar to HMMf an increasing trend in the mean accuracy could be observed. Table B.11 provides a detailed overview of the recognition accuracies of HMMf, (CMf,f_δ), and (CMf,f_{HMM}).

Alternative performance measures Considering alternative performance measures (see discussion in Section 5.4), for HMMf the mean value of Cohen's κ was .73 ($CI_{95} = .72, .74$). An increase of the number of involved persons (T2) (T{3–7} resulted in .75 ($CI_{95} = .74, .76$) (.71, .78, .83, .85, .83). For (CMf,f_δ,T1) a value of .81 ($CI_{95} = .8, .82$) was observed for κ. For T2 this value decreased to .75 ($CI_{95} = .71, .79$). Further decreases could be observed for an

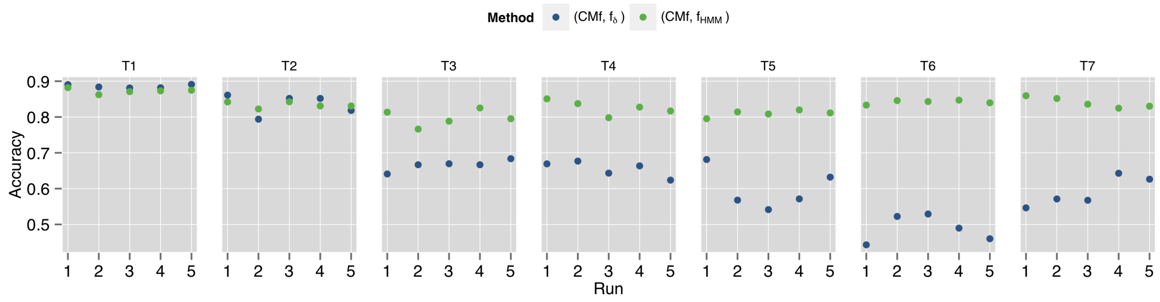

Figure 6.22.: Overview of the AR accuracies the CCBM-based classifier. The accuracy (CMf,f_δ) decreases with increasing the number of agents.

Predictor	β_0	β_1	$F_{(1,33)}$	r^2
NB	0.59	0.00	2.46	0.07
HMMf	0.76	0.01	58.77	0.64
(CMf,f_δ)	0.79	0.01	8.86	0.21
(CMf,f_{HMM})	0.86	-0.06	85.96	0.72

Table 6.15.: Linear models to predict the accuracy from the number of involved persons.

increased number of persons (T{3–7}): .54, .57, .52, .40, .51. For (CMf,f_{HMM}) a mean κ of .81 ($CI_{.95} = .8, .82$) for T1 was found. This value first decreases for T{2–4} to a minimum of .71 but then increases for T{5–7} to a maximum of .85.

CCBM action recognition With respect to recognising the activity, (CMf,f_δ) achieved a mean accuracy of .89 ($CI_{.95} = .88, .89$) for one protagonist (T1). For two persons (T2) a mean accuracy of .84 ($CI_{.95} = .8, .87$) was observed. The accuracy decreased to .67 (.66, .60, .49, .59) for T3 (T{4–7}). When using the restricted state space with trained transition probabilities (CMf,f_{HMM}) the recognition of the activity showed a mean accuracy of .87 ($CI_{.95} = .86, .88$) for T1. This represents a significant decrease of -1.31 pp (t_4=-5.15, p.007) in contrast to (CMf,f_δ). For T2 a mean accuracy of .83 ($CI_{.95} = .82, .84$) was observed for (CMf,f_δ), which did not differ significantly from (CMf,f_δ) (t_4=$-$.18, p.87). When considering three (T3) or more protagonists (T{4–7}), mean accuracies of .8 (.83, .81, .84, .84) were observed for (CMf,f_{HMM}). For all trial runs with more the three protagonist (T{3–7}) the accuracies increased significantly by at least 13.2 pp (p $<$ .001, $<$ .001, .002, $<$ .001, $<$ .001). A graphical overview of the action recognition results is fiven in Figure 6.22

CCBM context recognition With respect to the simultaneous recognition of activity and contextual information a significant correlation has been observed for both (CMf,f_δ) (Spearman's ρ=.97, S=205, p$<$ .001) and (CMf,f_{HMM}) (ρ=.45, S=3943, p.007).

To understand the effect of the number of involved persons on the accuracy, a linear regression model was fitted to predict the accuracy for each classifier. Table 6.15 provides an overview of these models. While none of the models were able to explain substantial amounts of the variance, they still allow to conclude a trend. Figure 6.23 provides a graphical representation of these models. For (CMf,f_δ) a significant negative influence of the number of the trial run was found ($F_{1,33} = 86$, p $<$ .001). When the number of persons increases by one, the accuracy decreases by 5.58 pp. Thus, as expected, increasing the state space size deteriorates

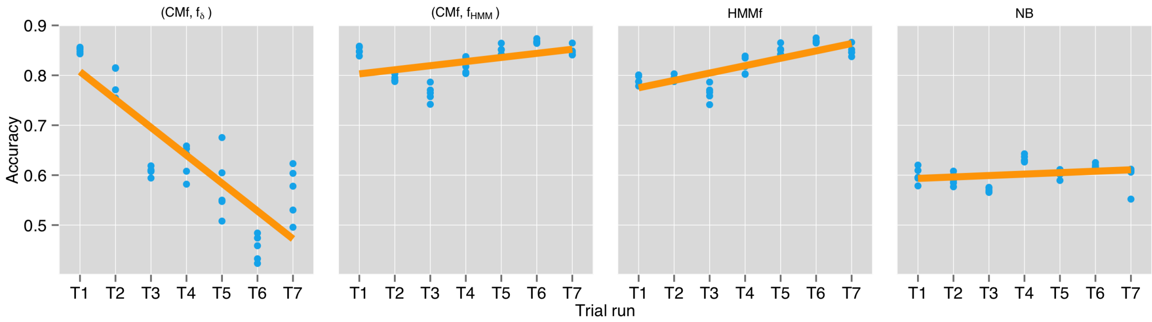

Figure 6.23.: Linear models fitted to predict accuracies of the different classifier from the number of involved persons.

the capabilities of tracking the correct hypotheses.

For HMMf, the opposite effect was suggested by the results. The fitted linear regression model, again, showed a significant effect of the trial number ($F_{1,33} = 58.8$, $p < .001$). An increase of the number of involved persons by one resulted in an increase of the accuracy by 1.48 pp. The reason for this is clearly that the HMMf is overfitted. An increased state space size results in a more sparse transition matrix. While for T1 the transition matrix of HMMf contains 29 of 121 possible transitions (ratio: .24), this ratio decreases to .03 for T3. This ratio further decreases to .006 (697 of 123201) for T7.

When using the transition matrix of HMMf and thereby strongly restricting the state size for (CMf,f_{HMM}), a similar significant increase in the accuracies could be observed ($F_{1,33} = 8.86$, p .005). Here, an increase in the number of persons by one resulted in an increase of the accuracy by .82 pp. Reason for this estimated lower increase is the difference in the intercept, which is higher for (CMf,f_{HMM}) (79.5) than for HMMf (76).

6.3.5. Discussion

Objective of Experiment X3 was to demonstrate the reusability of a causal behaviour model for different numbers of involved protagonists (research question RQ8). Furthermore, the influence of an increased number of persons on the recognition performance should be investigated (RQ9). The results of the experiment were reported in the previous section. This section further analyses the results and thus aims at answering the stated research questions.

Research question RQ8 Concerning the first question, a causal behaviour model has been developed, which was applied to different numbers of agents. By parameterising each action schema with the executing agent, a set of agent-specific ground actions have been generated. Each action describes the influence of the executing agent to the environment. Likewise, the set of agent-specific state features was created, describing those parts of the state space that depend on the agent. As a result, the state space grew exponentially with the number of involved persons (see Table 6.12).

As described, the persons in the trial acted independently and were therefore modelled as independent agents. While this would in general allow to separate the tracking problem for each agent, the selected setting prevents this, because it is not possible to assign sensor data to a specific agents. Thus, reconstruction of the behaviour sequences had to be done simultaneously for all involved persons. As a result, the state spaces could be traversed independently, allowing the goal distances to be computed for agent-specific state spaces. Traversing the compound state space would have been hardly possible due to high memory requirements.

The simultaneous recognition of the activities and the location of the agents demonstrated the reusability and therefore allows to answer the research question RQ8. It is indeed possible to reuse the causal behaviour model for a varying number of protagonists. Agent-specific information, such as the initial location or the goal can be used as parameters for the behaviour model. In fact, these agent-specific information have been used as identification features here.

Action selection for multiple agents In general, with the use of the exact goal distance f_δ as action selection heuristics the problem of exhaustively traversing large state spaces arises. This is infeasible for very large state spaces because of actual memory restrictions. Here, the problem could be circumvented by exploiting the fact that all agents are independent from each other. This is not the case in general. Additional research in the domain of action selection, especially in the case of multiple agents, is necessary to find sufficiently accurate approximations of the goal distance. A review of techniques from the domain of multi-agent planning, for instance, is suggested. On the other hand, as the recognition results in this experiment show a decline with increasing number of agents, the goal distance may not be sufficient to describe action selection in complex multi-agent scenarios. Thus, general research on multi-agent action selection is suggested.

The results of Experiment X3 suggest that CCBM is able to handle latent infinite state spaces. The recognition performance thereby strongly depends on the quality of the action selection heuristics. This corresponds to the results of Experiment X2, where the restricted goal distance consistently outperformed other action selection mechanisms. Here, the application of HMM transition matrix based action selection resulted in similar results as for the baseline classifiers.

Research question RQ9 With respect to research question RQ9, a strong decline of the recognition accuracy was observed when the number of agents, and thus the state space, grows. This, at first glance, seems to contradict the results of Experiment X2. A state space with millions of states was found to achieve higher recognition rates than the baseline HMM. However, the state features in the causal model of the second experiment were correlated – the value of a state feature depends on the value of another state features (e.g. The plate and the spoon are in use when the person is eating.). In contrast, the state space of the model of Experiment X3 constitutes independent parts for each agent. Adding an agent's state features to an existing state space, does neither introduce restrictions to the existing state space nor for the new part. In fact, a separate treatment of each agent would increase agent based recognition performance. However, as the sensor data does not allow any assignment, a separate treatment is impossible.

Experiment X3 concentrated on the simultaneous identification and location recognition from anonymous sensor data. By omitting the problem of identification, the problem can be addressed with lifted inference techniques [119] such as the hidden permutation model [38]. In this setting, these techniques can exploit the fact that the persons can not be distinguished by sensor data and focus on counting the persons per room rather that maintaining the identities. Further research should also target an automatic generation of such lifted models from the causal behaviour model.

Summary To summarise, the results show that (1.) research question RQ8 can be answered positive and (2.) a large effect of the number involved persons on recognition performance (RQ9) exist. CCBM allows the development of causal behaviour models that can be reused across different numbers of involved persons. Furthermore, CCBM is able to handle latent infinite state spaces as generated from an increased number of persons.

7
Discussion and Conclusion

"Any sufficiently advanced technology is indistinguishable from magic." – Arthur Clarke

SYNOPSIS: *Aim of this final chapter is to summarise the results this thesis. For this reason, first a summary of the individual chapters is provided. The results and their implications are then discussed with respect to the question* ***IQ****. Finally, the limitations of the experiments are discussed and further research directions are outlined.*

CHAPTER SOURCES: *Parts of this Chapter have been previously published in the following publication(s):*

- *Computational State Space Models for Activity and Intention Recognition. A Feasibility Study [133]*
- *Plan Synthesis for Probabilistic Activity Recognition [131]*
- *Towards Creating Assistive Software by Employing Human Behavior Models [129]*
- *Where are My Colleagues and Why? Tracking Multiple Persons in Indoor Environments[132]*
- *Evaluating the Robustness of Activity Recognition using Computational Causal Behavior Models[128]*

The previous chapters presented three experiments to investigate the capabilities of CCBM. This final chapter summarises the results with respect to the question **IQ**. For this purpose, Section 7.1 provides a summary of this thesis. Section 7.2 highlights in how far the requirements, raised initially, are met by CCBM. Finally, Section 7.4 provides further research topics.

7.1. Summary

The following section summarises this thesis. For this purpose the main points of each chapter are provided.

Motivation and requirements Main objective of this thesis is to investigate the initial question **IQ**. For this purpose, the requirements for inference systems for assistive systems are collected from the literature and two motivational examples: (1.) Plan – the ability to provide an estimate the current state, the action sequence and the final goal of the user, (2.) Online – the ability to infer the current situation in a complexity that is linear in length of the observation sequence processed so far, (3.) Uncertainty – the ability to cope with noisy and ambiguous sensor data, (4.) Latent infinity – the ability to handle very large, possibly infinite, state spaces, and (5.) Reusability – the ability to reuse parts of the domain model in different situations. It has been discussed that an inference system that provides the assistive system with information about the current situation and potential future development has to satisfy these requirements.

Computational state space models In order to assess the state of the art with respect to the initially ascertained requirements an in depth analysis of the related work on the research domains of AR and PR was conducted. This evaluation revealed that neither AR nor PR alone are able to provide a solution to this question. Thus, a combination of both research fields was strived to enable high-level reasoning on the base of low-level sensors. In addition to the review, a classification scheme that aids at summarising the related work was developed. Based on this classification scheme, a meta analysis [70] was performed to assess the problem size targeted by other researchers. For this purpose, several different factors (e.g. state space size or plan length) have been identified to serve as surrogates for measuring the problem size.

Section 2.1.4 introduced the concept of computational action languages to describe the connection between actions and states. Computational action languages describe behaviour as algorithms. Furthermore, the application of computational action languages allows to create reusable causal models of human behaviour. To this end, different researchers [17, 189, 95, 202] employed model-based descriptions in order to reconstruct action sequences of human protagonists. Computational action languages are utilised as they allow to replace training data by prior knowledge about causal dependencies. Computational State Space Models (see Section 2.2.1) make use of such symbolic descriptions of generative causal connections to describe actions that compute new states from old states. The application of generative causal structures is “more accessible to the mind” [181, p.21] rather than mere associations. However, when using such computational descriptions it is easy to arrive at very large – even infinite – state spaces. While such rich state spaces are desirable with respect to the descriptive power, they come at the price of an increased inference complexity. Due to more variation, a larger state space might achieve weaker performance as a smaller, more biased, state space (see discussion in Section 5.1). The fact that researchers, so far, applied CSSMs only in scenarios with easy-to-interpret sensor data (see discussion in Section 2.2.10), might be a direct consequence.

Computational causal behaviour models In Chapter 3, the results of the state of the art review were used to create a novel approach to integrate AR and PR – CCBM. CCBM captures the concept of CSSMs and thereby the flexibility and generalisability of a computational action language to create reusable causal models of human behaviour. It thus provides a plan synthesis based approach to PR. By utilising the framework of Bayesian filtering, CCBM combines the ability of high-level reasoning with low-level sensor data. Consequently, CCBM provides an integrated approach to activity, context and plan recognition. At the same time, causal behaviour models can be reused in the context of CCBM. Based on an explicit description of the environment by means of state features, the CCBM modelling language makes use of preconditions and effects to define human actions. A latent infinite LTS is created that describes all causally valid action sequences from the initial state to a goal. Thus, the LTS contains all possible action sequences a human protagonist might execute in order to achieve a goal.

To cope with noisy and ambiguous sensor data when reconstructing action sequences within such LTS, CCBM utilises methods of Bayesian filtering. A DBN has been introduced to capture the probabilistic semantics of the LTS and express the connections between different state variables (e.g. environment state and observation). To handle different types of observation data, for instance, the DBN contains two different observable variables Z and W, allowing the use of state based observations (e.g. dense sensing) or action observation (e.g. wearable sensors). Finally, a novel approximative filtering algorithm, especially tailored for categorical state spaces with sparse transition matrices, was introduced – the Marginal Filter.

The sample-based representation of probabilities in the PF is replaced by a weight-based representation. At the same time the marginal filter maintains a set of unique states, increasing the number of states to be tracked in the belief state. By additionally exploiting the fact that the transition matrix, generated from the causal model, is very sparse, the marginal filter allows the prediction step to be executed exactly.

With respect to *efficiency*, the framework of approximative Bayesian filtering allows inference to be executed in $\mathcal{O}(N\,T)$, where N is the number particles and T the length of the observation sequence. When it comes to the specific approximation methods, no difference between the PF and MF could be observed in small state spaces (see Section 6.1). However, the application of the MF in larger state spaces resulted in a significant increase of the recognition rates in comparison with the PF. As discussed earlier, the reason for this difference is that the particle filter was originally designed for continuous state spaces that defines a metric on states. This allows PFs to represent states by the density of samples in the vicinity of a point in the state space. Categorical state spaces, however, do not permit this approximation, as different points in the state space cannot be condensed to one state. The superiority of the MF over the PF is especially interesting in the light of the state of the art. Almost all researchers (except for Shi et al. [212]) applied (variants of) the particle filter method when it comes to approximate Bayesian inference in categorical state spaces.

Causally correct annotation In Chapter 4, it is discussed that methods exploiting causal dependencies of human behaviour rely on causally correct annotation. A novel annotation process is introduced that allows to produce causally correct annotation. Besides the opportunity to validate the correctness of the annotation sequence with respect to causal constraints, the annotation provides contextual information such as information about the environment. In fact, the annotation is based on an aLTS generated from a model-based description such as the CCBM modelling language.

Methods Chapter 5 gives an overview of the experimental procedures that were used to investigate the capabilities of CCBM. To this end, first, a list of research questions is introduced, each of them reflecting different aspects of the initial question **IQ**. The advantages of using empirical data rather than simulated are discusses briefly. Finally, Chapter 5 discusses the experimental procedure and the evaluation methods shared by all experiments. It is highlighted that the accuracy suffers from several drawbacks, albeit being the dominant performance measure in the domain of AR. For this purpose, other performance measures such as Cohen's κ, JSD, and DTW were selected for evaluation.

Experiments In Chapter 6, three experiments were conducted to investigate the capabilities of the proposed approach in comparison to standard baseline classifiers. Each experiment thereby addresses a subset of the research questions that were stated earlier. The evaluation of the experiments indicated that a suitably parameterised CCBM allows to achieve recognition rates at the same level as baseline classifiers that were learned from training data. Moreover, CCBM allowed to recognise the user's goal and additional contextual information at the same time. However, statements about the recognition performance of CCBM in absolute numbers require additional investigation.

Experiment X1 Experiment X1 – a meeting – represents a baseline experiment, constituting a problem size similar to that of the related work. The rationale here was that CCBM being able to achieve similar performance as baseline classifiers justifies further investigation with problems of larger size. Furthermore, the experiment focussed on demonstrating the reusability capabilities of CCBM with respect to the application domain. For this reason, a causal model that was created for one specific application domain was reused for another application scenario of the same domain. The evaluation of the first experiment showed that while the performance with respect to estimating the current action was similar to that of the baseline classifiers, CCBM outperforms the baseline classifier when it comes to the recognition of the correct goal. In conclusion, the results of Experiment X1 proved CCBM to be able to achieve similar recognition rates as the baseline classifier with problems of the same size as the related work. At the same time, the computational action language allowed the causal model to be reused within the same application domain.

Experiment X2 Experiment X2 – a kitchen task – was conducted to investigate the capabilities of CCBM on a problem of larger size. In fact, the kitchen task exceeds the complexity of previous research. Additionally, this experiment aimed at determining the effect of the different parameters (e.g. filtering method or observation model) to the recognition performance. Furthermore, it was investigated whether the same causal model can be used to reconstruct action sequences based on the different types of sensor observation. For this purpose, different observation models, exploiting either the W or the Z component of the DBN, were used. The evaluation of the second experiment showed a superiority of CCBM with respect to reconstructing the action sequence in comparison to the baseline classifiers. At the same time CCBM allowed to estimate contextual information, provided by the underlying LTS, with high accuracy. Maintaining rich state spaces was found to have no negative influence on the recognition performance, but rather expands the variety of estimation tasks accessible with such a model. Additionally, it has been shown that all parameters, except for the duration model, significantly influence the recognition performance, substantiating the initial choice to introduce the respective modelling factors (see Section 3.1). The small and non-significant effect of the duration model, in contrast, justifies the choice of empirical duration models as replacement for parametric ones. The filtering mode (i.e. PF or MF) and the observation model were identified as factors with the largest effect. The use of non-i.i.d. sensor data, for instance, resulted in a massive drop in the recognition accuracy. With respect to reusability, a proof by demonstration was obtained that CCBM allows to exchange the type of observation model without further changes to the causal model of human behaviour.

Experiment X3 Experiments X1 and X2 indicated a superiority of CCBM in comparison to the baseline classifiers with respect to the recognition performance. Experiment X3 – indoor localisation – was conducted to investigate the influence of increasing the state space by adding additional degrees of freedom to the model. Moreover, the multi-agent reusability (R5.3) was investigated. By parameterising the causal model with the number of involved persons, the human behaviour model could be reused with a varying number of persons. The evaluation of the third experiment showed that the causal model can indeed be reused with a varying number of persons. A negative effect to the recognition performance was observed. At first glance, this seems to be in contrast to the experiences from the second experiment, where CCBM allows superior recognition rates although employing a rich state space. A more detailed look at the state features reveals differences. The state features in the kitchen task are highly correlated with the action sequence of the human protagonist. In the multi-agent model of Experiment X3, each agent is modelled as completely independent. In consequence, adding an additional person to the model introduces state features that are uncorrelated with the state features of the other agents. Even the observation model, which permits the identification of persons from sensor data is unable to counteract this effect. The negative influence of the number of persons to the recognition performance could not be observed for the baseline classifiers which determined the state space from the training data. By using using the transition matrix of the baseline classifier, it was shown that the negative influence is not present for CCBM in general, but rather suggest further investigations of action selection mechanisms in multi-agent settings.

7.2. Discussion

CCBM combines the flexibility of computational action languages with the probabilistic semantics of DBNs. To this end, it allows to create reusable causal models of human behaviour and to reconstruct actions sequences of human protagonists from sensor data. By employing a model-based description by means of preconditions and effects (see Section 2.1.4) the CCBM modelling language uses an explicit description of the environment state. This allows to reason about activities and contextual information at the same time. By taking goal-directed action selection (see Section 3.1.4) into account, it allows to estimate the user's goal and thus the potential sequence of future actions. In consequence, CCBM represents an integrated approach to activity, context and plan recognition. CCBM therefore satisfies the requirement for PR (R1).

Inference algorithms With respect to the inference, it could be shown that the MF achieves a better recognition performance when it comes to large state spaces. The experiments showed that the MF achieves similar recognition performance as the PF in small state spaces, but outperforms the PF in large state spaces. This is a result of the improved belief state representation. The standard method for approximative inference – the PF – represents state probabilities by sets of particles. The MF, an inference method particularly tailored for categorical state spaces, maintains one probability per state and thus makes better usage of the resources. This allows the MF to maintain more states and thus allows a more accurate belief state representation.

CCBM satisfies the requirements With respect to the requirements raised initially, it has been shown that all requirements are satisfied. R1 is addressed by combining the domains of AR and PR in a way that CCBM allows to infer the user's plan from sensor data. This has been shown in all experiments. The requirement R2 has been addressed by the selection of the inference framework. As discussed in Section 3.2, the framework of Bayesian filtering allows inference to done with linear time complexity. Inference for one time-step being independent from the length of the observation sequence processed allows for online behaviour reconstruction. Assistive systems based on POMDPs such as discussed in Hoey et al. [99] rely on the ability to track the user's state efficiently. Consequently, one prerequisite for efficient assistance is efficient state tracking. A second benefit of the Bayesian filtering framework is the ability to cope with noisy and ambiguous sensor data. Thus, the requirement R3 is also addressed by this choice. The maintenance of latent infinite state spaces, as requested by requirement R4, is addressed by use of the computational action languages that allows to express actions as computable functions. The application of generative filtering methods allow the state space, which is described by the computational action language, to be explored incrementally – only the parts of the state space that are considered during inference are explored. Finally, the requirement for reusability R5 is addressed by employing a model-based description that can be configured with application specific parameters. Three different aspects have been identified and each experiment did a proof by demonstration to show the satisfaction of one aspect. The following aspects of reusability were established: R5.1 A causal behaviour model that was created for a specific application scenario – a three person meeting with a four-point agenda – has been reused in similar, yet different scenario, of the same application domain. This aspect was demonstrated in Experiment X1. R5.2 A switch between the different types of observation models required no change to the causal behaviour model. This aspect was demonstrated in Experiment X2. R5.3 A causal model, establishing a set of actions for a set of agents, has been reused with a different number of agents. This aspect was demonstrated in Experiment X3.

To conclude, by combining a computational action language to represent a causal model of human behaviour with probabilistic inference techniques that allow to cope for noisy and ambiguous sensor data, it is possible to reconstruct human action sequences efficiently.

7.3. Limitations

Beside the contributions of this work, the presented studies have the following limitations:

- The studies were based on a small set of convenience samples. While this does not invalidate the studies' results, statements about the "real" effect can hardly be made. Therefore, all studies concentrated on a comparison of recognition results with baseline classifiers rather than on absolute values.
- The evaluation of all studies are not based on cross validation. The reasons are of practical nature, as it would require to manually create causal behaviour models for the selected "training" subset. Additionally, this would conceptually require a set of equally skilled domain experts. To not penalise the baseline classifiers, each of them was also trained without cross validation. However, it would be of interest in how far a causal model that was created by considering a subset of "training" sequences is able to explain the remaining sequences. This would allow to assess the number of changes that are introduced to the causal model by one individual action sequence.
- The observation models were not subject of optimisation. For all experiments, observation models were selected based on prior analysis. As observation models were not the primary focus of the studies, no further effort was spent in optimisation. The observation model of the kitchen study, for instance, used a simple scrambling mechanism to eliminate the effect of non-i.i.d. samples. This is not possible in application settings, as it requires to preprocess the complete observation sequence based on a annotation labels.

To establish a more comprehensive statement about the capabilities of CCBM and specifically the recognition performance a study of larger size is necessary. An analysis of the recognition performance of CCBM with larger datasets (e.g. the CMUMMAC database [60]) is suggested.

7.4. Outlook and Future Research

The results of this thesis show that CCBM allows the efficient reconstruction of causal structures of human behaviour. However, as described in Section 7.3, the studies were subject to several limitations which prevent to draw general conclusions. This includes observations about the growth (with respect to action schemata and state features) of the causal model with including additional trial runs. The conclusion of generalisable results with respect to reusability and recognition performance requires additional studies do be conducted. Additionally, different areas for further research were sketched within this thesis. This section provides an overview of both, potential future extensions to CCBM and studies to assess the capabilities of CCBM in a more generalisable extent.

Additional studies Aim of this thesis was to provide a performance comparison of CCBM with standard baseline classifiers (e.g. HMM or NB). For this purpose, it was considered sufficient to concentrate on a small number of protagonists. Additionally, investigations were based on simple observation models that were not subject of further optimisation. To provide statements on a larger scale, additional studies are suggested. The CMUMMAC database [60], for instance, provides a large amount of trial runs. 55 participants were instructed to make one (or multiple) meal(s), while being observed by various sensors such as accelerometers, video cameras, or RFIDs. In all, about 150 trial runs were recorded, each of them targeting one of five recipes. Additionally, several trials with abnormal behaviour such as the kitchen being on fire were recorded. However, a missing comprehensive annotation, prevents the dataset from being used in the literature (see discussion in Chapter 4). This dataset is also suited to be annotated by means of the proposed annotation process. First attempts to annotate the “Brownie” subset of the dataset showed very high interrater reliabilities (Cohen’s κ of about .8 and higher). This signals that the proposed annotation process is able to produce reliable annotation.

Machine learning approach to causal behaviour models The development of causal models of human behaviour requires domain experts to capture causal dependencies. Some application domains, however, provide such causal knowledge in structured descriptions. Examples of such domains are the operation of machines for manufacturing [1], operation based on instruction manuals, or cooking based on recipes [261]. Two different approaches exist to put such structured knowledge into causal behaviour models:

1 Manually extracting the causal dependencies from the description and transferring the knowledge to the causal model. First attempts to this approach have been made by Bader et al. [16].

2 The second approach is to automatically analyse the descriptions in order to infer the causal dependencies. Yordanova [257] proposed an approach to automatically generate causal models of human behaviour by using such technique.

Similarly, applying machine learning approaches to generate planning models is part of active research [264, 162, 254]. The development of causal behaviour would benefit from applying such techniques, as they would allow to automatically create a starting point for model development.

Observation models With respect to observation models, several improvements are thinkable. While for the studies in this thesis, non-i.i.d. observation data was handled by a simple (but effective) scrambling mechanism, the online application requires more sophisticated methods. Here, the use of sub-models to describe the process of single actions is suggested. Such a sub-model could describe how the sensor data changes during the execution of an action. The concept of Rao-Blackwellisation [66] provides a powerful mechanism to include motion models such as a Kalman filters into the statistical model. Such model could, for instance, be used to describe the progress of taking objects from a cupboard, where first the arm moves away from the body to the cupboard, then the hand grabs the object and finally the arm moves back to the body.

Additionally, research on the unsupervised generation of observations models is recommended, as it would reduce the need for annotated sensor data. First attempts have be sketched by Hein [90]. Finally, the application of an evaluation based on cross-validation seems advisable, to understand the effect of single subjects to the overall recognition performance. This, in combination with a model development based on machine learning allows to draw conclusion about the generalisability of the proposed approach.

Action selection heuristics The goal-based action selection was demonstrated to achieve good recognition rates. However, the analysis of the different goal-directed action selection heuristics in Section 6.2.4.4 revealed substantial differences with respect to the action sequences actually executed. The full goal distance, which was generated from knowledge about the causal model, achieved significantly inferior results than the restricted goal distance that was generated by analysing the actual action sequences. A machine learning approach could be utilised to predict the error of the full goal-distance (or any other goal-based heuristic) to the restricted. Satzger and Kramer [204], for instance, proposed to train a neural network for error prediction of goal-directed heuristics. In this way, the number of action sequences to be analysed could be reduced. Additionally, the effect of goal distance based heuristics, such as the landmark heuristic [193], to the recognition performance should be investigated. These heuristics are used in the planning community for planning without full state space exploration and show good results with respect to planning time [193]. Finally, as discussed in Section 3.1.4 further research on situation driven [210] action selection features is suggested. A comparison of the shortest possible plans with the plans actually executed by the humans suggests that goal orientation is not the only factor that influences the selection of the next action. Inference as well as prediction would benefit from knowledge about additional features that influence action selection.

The studies in this thesis focussed on goal-directed behaviour. CCBM is able to reconstruct any causally correct action sequence, as each action, with preconditions met by the current state, are considered. By employing different action selection features, for instance, one for goal-directed behaviour and one for erratical behaviour, CCBM would allow to distinguish between these two types of behaviours. Similar to the goal recognition, for each different behaviour the action selection mechanism would influence the probability of the action sequence in different way. As a result, this would allow inference to discriminate between goal-directed (with known goal) and erratical behaviour. Furthermore, it might be of interest to distinguish between goal-directed behaviour where the goal is unknown from erroneous behaviour. First ideas to recognise situations of confusion for people suffering from dementia by use CCBM have been illustrated by Henkel et al. [94].

Improvement of the annotation process Finally, with respect to the annotation process described in Chapter 4, finding the "correct" start and end time of actions has been observed to be the main reason for low interrater reliability. This effect is even increased in comparison to "standard" activity annotation due to the fine grained annotation. Since activities such as cooking are split up into their sub tasks (e.g. interleaved sequences of filling and stirring), the number of transitions increases. One approach to increase the reliability would be to move away from annotating the points of transition between two actions, but rather annotate only the mid region of each action. The transitions would then be automatically annotated as soft passages from one action to another. Additionally, this better reflects the transition in the sensor observations (see discussion in Section 3.1.5).

Bibliography

[1] M. Aehnelt and S. Bader. Information assistance for smart assembly stations. In *Proceedings of the 7th International Conference on Agents and Artificial Intelligence (ICAART'15)*, pages 143–150, Lisbon, Portugal, 10–12 Jan. 2015. DOI:10.5220/0005216501430150. (Cited on page 125.)

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1–16:43, Apr. 2011. DOI:10.1145/1922649.1922653. (Cited on pages 13, 20, 23, and 92.)

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, Dec. 1974. DOI:10.1109/TAC.1974.1100705. (Cited on page 155.)

[4] D. Albrecht, I. Zukerman, and A. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User Modeling and User-Adapted Interaction*, 8(1-2):5–47, Mar. 1998. DOI:10.1023/A:1008238218679. (Cited on page 56.)

[5] A. Ali, R. C. King, and G.-Z. Yang. Semi-supervised segmentation for activity recognition with multiple eigenspaces. In *5th International Summer School and Symposium on Medical Devices and Biosensors (ISSS-MDBS'08)*, pages 314–317, Hong Kong, China, 1–3 June 2008. IEEE. DOI:10.1109/ISSMDBS.2008.4575082. (Cited on page 152.)

[6] American Occupational Therapy Association. Occupational therapy practice framework: Domain and process (third edition). *The American Journal of Occupational Therapy*, 68: S1–S48, Mar. 2014. DOI:10.5014/ajot.2014.682006. (Cited on page 90.)

[7] O. Amft, H. Junker, and G. Troster. Detection of eating and drinking arm gestures using inertial body-worn sensors. In *Proceedings of Ninth IEEE International Symposium on Wearable Computers*, pages 160–163, Osaka, Japan, 18–25 Oct. 2005. DOI:10.1109/ISWC.2005.17. (Cited on pages 12 and 152.)

[8] J. R. Anderson. *The Architecture of Cognition.* Harvard University Press, Cambridge, MA, 1983. ISBN 978-0805822335. (Cited on pages 29 and 43.)

[9] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, Oct. 2004. DOI:10.1037/0033-295X.111.4.1036. (Cited on page 32.)

[10] P. Andrews, I. Zaihrayeu, and J. Pane. A classification of semantic annotation systems. *Semantic Web*, 3(3):223–248, Aug. 2012. DOI:10.3233/SW-2011-0056. (Cited on page 56.)

[11] M. G. Armentano and A. Amandi. Plan recognition for interface agents. *Artificial Intelligence Review*, 28(2):131–162, Aug. 2007. DOI:10.1007/s10462-009-9095-8. (Cited on pages 15, 16, and 151.)

[12] M. G. Armentano and A. Amandi. Goal recognition with variable-order markov models. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1635–1640, Pasadena, CA, USA, 14–17 July 2009. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1661445.1661708. (Cited on pages 42 and 155.)

[13] M. S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50 (2):174–188, Feb. 2002. DOI:10.1109/78.978374. (Cited on page 46.)

[14] A. P. Association. *Publication Manual of the American Psychological Association*. American Psychological Association, Washington, DC, sixth edition, 2009. ISBN 1-433-80561-8. (Cited on page 74.)

[15] D. Avrahami-Zilberbrand and G. A. Kaminka. Keyhole adversarial plan recognition for recognition of suspicious and anomalous behavior. In G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui, editors, *Plan, activity, and intent recognition*, pages 87–122. Elsevier, Amsterdam, The Netherlands, 10 Apr. 2014. ISBN 9780123985323. (Cited on pages 15 and 151.)

[16] S. Bader, F. Krüger, and T. Kirste. Computational causal behaviour models for assisted manufacturing. In *Proceedings of 2nd International Workshop on Sensor-based Activity Recognition and Interaction (IWOAR'15)*, Rostock, Germany, 25–26 June 2015. ACM. DOI:10.1145/2790044.2790058. (Cited on page 125.)

[17] C. L. Baker, R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, Dec. 2009. DOI:10.1016/j.cognition.2009.07.005. (Cited on pages 12, 20, 22, 23, 29, 30, 31, 33, 34, 39, 42, 43, 99, 120, 151, and 155.)

[18] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg, 21–23 Apr. 2004. DOI:10.1007/978-3-540-24646-6_1. (Cited on pages 12, 13, 56, and 152.)

[19] M. Bauer. Acquisition of user preferences for plan recognition. In *Proceedings of the Fifth International Conference on User Modeling (UM'96)*, pages 105–112, Kailua-Kona, Hawaii, 2–5 Jan. 1996. URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.1027. (Cited on page 56.)

[20] C. Baum and D. F. Edwards. Cognitive performance in senile dementia of the alzheimer's type: The kitchen task assessment. *The American Journal of Occupational Therapy*, 47 (5):431–436, May 1993. DOI:doi:10.5014/ajot.47.5.431. (Cited on page 90.)

[21] R. Baxter, D. Lane, and Y. Petillot. Recognising agent behaviour during variable length activities. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, pages 803–808, Amsterdam, The Netherlands, 16–20 Aug. 2010. IOS Press. URL http://dl.acm.org/citation.cfm?id=1860967.1861124. (Cited on page 10.)

[22] A. Ben-David. Comparison of classification accuracy using cohen's weighted kappa. *Expert Systems with Applications*, 34(2):825–832, Feb. 2008. DOI:10.1016/j.eswa.2006.10.022. (Cited on page 73.)

[23] A. Berger, S. D. Pietra, and V. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):40–71, Mar. 1996. URL http://dl.acm.org/citation.cfm?id=234289. (Cited on page 43.)

[24] N. Blaylock and J. Allen. Statistical goal parameter recognition. In *Proceedings of the 14th international Conference on Automated Planning and Scheduling*, pages 297–305, Whistler, British Columbia, Canada, 3–7 June 2004. AAAI Press. URL https://www.aaai.org/Papers/ICAPS/2004/ICAPS04-036.pdf. (Cited on page 56.)

[25] N. Blaylock and J. Allen. Recognizing instantiated goals using statistical models. In *Proceedings of IJCAI Workshop on Modeling Others from Observations (MOO'05)*, pages 79–86, Edinburgh, Scotland, 8 Apr. 2005. URL https://www.cs.rochester.edu/research/cisd/pubs/2005/blaylock-allen-moo2005.pdf. (Cited on page 15.)

[26] N. Blaylock and J. Allen. Hierarchical goal recognition. In G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui, editors, *Plan, activity, and intent recognition*, pages 3–32. Elsevier, Amsterdam, The Netherlands, 10 Apr. 2014. ISBN 9780123985323. (Cited on pages 16, 42, 55, 56, 66, 74, 151, and 155.)

[27] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461, Aug. 1973. DOI:10.1016/S0022-0000(73)80033-9. (Cited on page 50.)

[28] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, and A. Mihailidis. A decision-theoretic approach to task assistance for persons with dementia. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, IJCAI'05, pages 1293–1299, Edinburgh, Scotland, 30 July–5 Aug. 2005. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1642293.1642499. (Cited on page 23.)

[29] B. Bonet and H. Geffner. mgpt: A probabilistic planner based on heuristic search. *Journal Artificial Intelligence Research (JAIR)*, 24:933–944, Dec. 2005. DOI:10.1613/jair.1688. (Cited on page 54.)

[30] E. Z. Borzeshi, O. P. Concha, R. Y. Da Xu, and M. Piccardi. Joint action segmentation and classification by an extended hidden markov model. *IEEE Signal Processing Letters*, 20(12):1207–1210, Oct. 2013. DOI:10.1109/LSP.2013.2284196. (Cited on page 152.)

[31] B. Bouchard, A. Bouzouane, and S. Giroux. A keyhole plan recognition model for Alzheimer's patients: first results. *Applied Artificial Intelligence*, 21(7):623–658, Aug. 2007. DOI:10.1080/08839510701492579. (Cited on pages 15, 23, and 151.)

[32] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in bayesian networks. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence, (UAI'96)*, pages 115–123, Portland, OR, 1996. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=2074298. (Cited on pages 26 and 37.)

[33] O. Brdiczka, P. Reignier, and J. Crowley. Detecting individual activities from video in a smart home. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4692 of *Lecture Notes in Computer Science*, pages 363–370. Springer Berlin Heidelberg, Vietri su Mare, Italy, 12–14 Sept. 2007. DOI:10.1007/978-3-540-74819-9_45. (Cited on pages 13 and 152.)

[34] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. The balanced accuracy and its posterior distribution. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR'10)*, pages 3121–3124, Istanbul, Turkey, 23–26 Aug. 2010. IEEE. DOI:10.1109/ICPR.2010.764. (Cited on page 72.)

[35] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall. Recognizing daily activities with rfid-based sensors. In *Proceedings of the 11th International Conference on Ubiquitous Computing (Ubicomp'09)*, pages 51–60, Orlando, FL, USA, 30 Sep.–3 Oct. 2009. ACM. DOI:10.1145/1620545.1620553. (Cited on page 152.)

[36] H. H. Bui. A general model for online probabilistic plan recognition. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1309–1315, Acapulco, Mexico, 9–15 Aug. 2003. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1630659.1630846. (Cited on page 42.)

[37] H. H. Bui, S. Venkatesh, and G. A. W. West. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research (JAIR)*, 17:451–499, Dec. 2002. DOI:10.1613/jair.839. (Cited on pages 16, 20, 23, 24, 25, 26, 27, 28, 29, 31, 33, 34, 40, 46, 66, 76, and 106.)

[38] H. H. Bui, D. Phung, S. Venkatesh, and H. Phan. The hidden permutation model and location-based activity recognition. In *Proceedings of the 23rd national Conference on Artificial Intelligence (AAAI'08)*, pages 1345–1350, Chicago, Illinois, 13–17 July 2008. AAAI Press. URL http://dl.acm.org/citation.cfm?id=1620270.1620283. (Cited on pages 23 and 118.)

[39] A. Bulling, J. Ward, H. Gellersen, and G. Tröster. Eye movement analysis for activity recognition using electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):741–753, Apr. 2011. DOI:10.1109/TPAMI.2010.86. (Cited on page 152.)

[40] A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33:1–33:33, Jan. 2014. DOI:10.1145/2499621. (Cited on pages 12, 13, 17, 20, 55, 56, 92, 93, and 152.)

[41] C. Burghardt, M. Giersich, and T. Kirste. Synthesizing probabilistic models for team activities using partial order planning. In *KI'2007 Workshop: Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (AIM-CU)*, Osnabrück, Germany, 2007. (Cited on pages 20, 23, 28, and 34.)

[42] C. Burghardt, M. Wurdel, S. Bader, G. Ruscher, and T. Kirste. Synthesising generative probabilistic models for high-level activity recognition. In L. Chen, C. Nugent, J. Biswas, and J. Hoey, editors, *Activity Recognition in Pervasive Intelligent Environment*, volume 4 of *Atlantis Ambient and Pervasive Intelligence*, pages 209–236. Atlantis Press, Paris, France, 5 May 2011. ISBN 978-90-78677-42-0. DOI:10.2991/978-94-91216-05-3_10. (Cited on pages 28, 29, 34, and 80.)

[43] B. Caramiaux, N. Montecchio, A. Tanaka, and F. Bevilacqua. Adaptive gesture recognition with variation estimation for interactive systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(4):18:1–18:34, Dec. 2014. DOI:10.1145/2643204. (Cited on pages 12, 46, and 155.)

[44] S. Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11:31–48, Mar. 2001. DOI:10.1023/A:1011118925938. (Cited on pages 15 and 16.)

[45] P. Casale, O. Pujol, and P. Radeva. Human activity recognition from accelerometer data using a wearable device. In J. Vitrià, J. M. Sanches, and M. Hernández, editors, *Pattern Recognition and Image Analysis*, volume 6669 of *Lecture Notes in Computer Science*, pages 289–296. Springer Berlin Heidelberg, Las Palmas de Gran Canaria, Spain, 8–10 June 2011. DOI:10.1007/978-3-642-21257-4_36. (Cited on pages 13 and 152.)

[46] L. Chen, J. Hoey, C. Nugent, D. Cook, and Z. Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(6):790–808, May 2012. DOI:10.1109/TSMCC.2012.2198883. (Cited on pages 2, 13, 14, 16, 20, 23, 91, and 107.)

[47] L. Chen, C. D. Nugent, and H. Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 24(6): 961–974, June 2012. DOI:10.1109/TKDE.2011.51. (Cited on pages 23 and 152.)

[48] J. Cheng, O. Amft, and P. Lukowicz. Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition. In P. Floréen, A. Krüger, and M. Spasojevic, editors, *Pervasive Computing*, volume 6030 of *Lecture Notes in Computer Science*, pages 319–336. Springer Berlin Heidelberg, Helsinki, Finland, 17–20 May 2010. DOI:10.1007/978-3-642-12654-3_19. (Cited on page 152.)

[49] H.-S. Chung, J.-M. Kim, Y.-C. Byun, and S.-Y. Byun. Retrieving and exploring ontology-based human motion sequences. In O. Gervasi, M. Gavrilova, V. Kumar, A. Laganà, H. Lee, Y. Mun, D. Taniar, and C. Tan, editors, *Computational Science and Its Applications – ICCSA 2005*, volume 3482 of *Lecture Notes in Computer Science*, pages 788–797. Springer Berlin Heidelberg, 9–12 May 2005. DOI:10.1007/11424857_86. (Cited on page 57.)

[50] B. Clarkson, K. Mase, and A. Pentland. Recognizing user's context from wearable sensors: Baseline system. Technical Report Vismod TR #519, Vision and Modeling, Media Lab. MIT, 4 Mar. 2000. URL http://hd.media.mit.edu/tech-reports/TR-519.pdf. (Cited on page 13.)

[51] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, Apr. 1960. DOI:10.1177/001316446002000104. (Cited on page 73.)

[52] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences.* Lawrence Erlbaum Associates, second edition, 12 Aug. 1988. ISBN 978-0805802832. (Cited on page 74.)

[53] P. R. Cohen, C. R. Perrault, and J. F. Allen. Beyond question answering. pages 245–274. Lawrence Erlbaum Associates Inc, first edition, 1 Aug. 1981. ISBN 978-0898592665. (Cited on pages 15 and 151.)

[54] D. Cook, J. C. Augusto, and V. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, Aug. 2009. DOI:doi:10.1016/j.pmcj.2009.04.001. (Cited on page 56.)

[55] D. J. Cook and S. K. Das. How smart are our environments? An updated look at the state of the art. *Pervasive and Mobile Computing*, 3(2):53–73, Mar. 2007. DOI:10.1016/j.pmcj.2006.12.001. (Cited on page 1.)

[56] P. Dai, H. Di, L. Dong, L. Tao, and G. Xu. Group interaction analysis in dynamic context. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(1):275–282, Feb. 2008. DOI:10.1109/TSMCB.2008.2009559. (Cited on pages 23 and 76.)

[57] D. Dash, M. Voortman, and M. de Jongh. Sequences of mechanisms for causal reasoning in artificial intelligence. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, pages 839–845, Beijing, China, 3–9 Aug. 2013. URL http://dl.acm.org/citation.cfm?id=2540250. (Cited on page 23.)

[58] N. Davies, E. Mynatt, and I. Siio. *UbiComp 2004: 6th International Conference Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 7–10 Sept. 2004. DOI:10.1007/b99948. (Cited on page 152.)

[59] F. de la Torre, J. Hodgins, J. Montano, S. Valcarcel, R. Forcada, and J. Macey. Guide to the carnegie mellon university multimodal activity (CMU-MMAC) database. Technical Report CMU-RI-TR-08-22, Robotics Institute, Carnegie Mellon University, Apr. 2008. URL http://repository.cmu.edu/cgi/viewcontent.cgi?article=1134&context=robotics. (Cited on pages 57, 60, and 92.)

[60] F. de la Torre, J. K. Hodgins, J. Montano, and S. Valcarcel. Detailed human data acquisition of kitchen activities: the CMU-Multimodal Activity Database (CMU-MMAC). In *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, in conjuction with CHI 2009*, pages 1–5, 4 Apr. 2009. URL http://www.cs.cmu.edu/~ftorre/web_page/ca.cs.cmu.edu/projects/CMU-MMAC.html. (Cited on pages 56, 92, 93, 124, and 125.)

[61] R. Denney. A comparison of the model-based & algebraic styles of specification as a basis for test specification. *SIGSOFT Software Engineering Notes*, 21(5):60–64, Sept. 1996. ISSN 0163-5948. DOI:10.1145/235969.235988. (Cited on pages 18 and 19.)

[62] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook. Simple and complex activity recognition through smart phones. In *8th International Conference on Intelligent Environments (IE'12)*, pages 214–221, Guanajuato, Mexico, 26–29 June 2012. IEEE. DOI:10.1109/IE.2012.39. (Cited on page 152.)

[63] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, Dec. 1959. DOI:10.1007/BF01386390. (Cited on page 43.)

[64] M. Donnelly, T. Magherini, C. Nugent, F. Cruciani, and C. Paggetti. Annotating sensor data to identify activities of daily living. In B. Abdulrazak, S. Giroux, B. Bouchard, H. Pigot, and M. Mokhtari, editors, *Toward Useful Services for Elderly and People with Disabilities*, volume 6719 of *Lecture Notes in Computer Science*, pages 41–48. Springer Berlin Heidelberg, Montreal, Canada, 20–22 June 2011. DOI:10.1007/978-3-642-21535-3_6. (Cited on page 56.)

[65] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA'05)*, pages 64–69, 15–17 Sept. 2005. DOI:10.1109/ISPA.2005.195385. (Cited on page 46.)

[66] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the 16th conference on Uncertainty in Artificial Intelligence (UAI'00)*, pages 176–183, Stanford, California, 20 June–3 July 2000. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=720075. (Cited on page 125.)

[67] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, July 2000. DOI:10.1023/A:1008935410038. (Cited on page 46.)

[68] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Berlin Heidelberg, 21 June 2001. ISBN 978-0387951461. (Cited on page 46.)

[69] T. Duong, D. Phung, H. Bui, and S. Venkatesh. Efficient duration and hierarchical modeling for human activity recognition. *Artificial Intelligence*, 173(7–8):830–856, May 2009. DOI:doi:10.1016/j.artint.2008.12.005. (Cited on pages 23 and 152.)

[70] P. Ellis. *The Essential Guide to Effect Sizes: Statistical Power, Meta-Analysis, and the Interpretation of Research Results.* Cambridge University Press, 1 July 2010. ISBN 978-0521142465. (Cited on pages 74 and 120.)

[71] M. Ermes, J. Pärkkä, and L. Cluitmans. Advancing from offline to online activity recognition with wearable sensors. In *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'08).*, pages 4451–4454. IEEE, 20–25 Aug. 2008. DOI:10.1109/IEMBS.2008.4650199. (Cited on page 152.)

[72] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1):20–26, Jan. 2008. DOI:10.1109/TITB.2007.899496. (Cited on page 12.)

[73] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. In *Proceedings of the second International Joint Conference on Artificial Intelligence (IJCAI'71)*, pages 608–620, London, England, 1–3 Sept. 1971. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1622939. (Cited on page 19.)

[74] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, July 2003. ISSN 1536-1268. DOI:10.1109/MPRV.2003.1228524. (Cited on pages 107 and 108.)

[75] A. Garland and N. Lesh. Plan evaluation with incomplete action descriptions. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'02)*, pages 461–467, Edmonton, Alberta, Canada, 28 July–1 Aug. 2002. American Association for Artificial Intelligence. URL https://www.aaai.org/Papers/AAAI/2002/AAAI02-070.pdf. (Cited on page 14.)

[76] C. W. Geib and R. P. Goldman. Plan recognition in intrusion detection systems. In *Proceedings of the DARPA Information Survivability Conference & Exposition II, (DISCEX'01)*, volume 1, pages 46–55, Anaheim, CA, USA, 12–14 June 2001. IEEE. DOI:10.1109/DISCEX.2001.932191. (Cited on pages 15 and 151.)

[77] C. W. Geib and R. P. Goldman. Partial observability and probabilistic plan/goal recognition. In *Proceedings of IJCAI Workshop on Modeling Others from Observations (MOO'05)*, Edinburgh, Scotland, 8 Apr. 2005. URL http://people.ict.usc.edu/~pynadath/MOO-2005/4.pdf. (Cited on pages 12 and 151.)

[78] C. W. Geib and R. P. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, July 2009. DOI:10.1016/j.artint.2009.01.003. (Cited on page 15.)

[79] M. Giersich. *Concept of a Robust & Training-free Probabilistic System for Real-time Intention Analysis in Teams.* PhD thesis, University of Rostock, Rostock, Germany, Oct. 2009. URL http://rosdok.uni-rostock.de/resolve/id/rosdok_disshab_0000000355. (Cited on pages 68 and 78.)

[80] M. Giersich and T. Kirste. Effects of agendas on model-based intention inference of cooperative teams. In *Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing, (CollaborateCom'07)*, pages 456–463, New York, NY, USA, 12-15 Nov. 2007. IEEE. DOI:10.1109/COLCOM.2007.4553875. (Cited on page 76.)

[81] G. Gigerenzer. *Simply Rational Decision Making in the Real World*. Oxford University Press, 31 Mar. 2015. ISBN 978-0199390076. (Cited on page 42.)

[82] T. Giorgino. Computing and visualizing dynamic time warping alignments in r: the dtw package. *Journal of statistical Software*, 31(7):1–24, 14 Aug. 2009. DOI:10.18637/jss.v031.i07. (Cited on page 73.)

[83] J. A. Goguen and G. Malcom. *Software Engineering with OBJ: Algebraic Specifications in Action*, volume 2 of *Advances in Formal Methods*. Springer Berlin Heidelberg, 2000. ISBN 978-1475765410. (Cited on page 18.)

[84] S. Gong, C. Loy, and T. Xiang. Security and surveillance. In T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal, editors, *Visual Analysis of Humans*, pages 455–472. Springer London, 8 Oct. 2011. DOI:10.1007/978-0-85729-997-0_23. (Cited on pages 1 and 12.)

[85] J. Guttag and J. Horning. The algebraic specification of abstract data types. *Acta Informatica*, 10(1):27–52, Mar. 1978. DOI:10.1007/BF00260922. (Cited on page 18.)

[86] J. Hamm, B. Stone, M. Belkin, and S. Dennis. Automatic annotation of daily activity from smartphone-based multisensory streams. In D. Uhler, K. Mehta, and J. Wong, editors, *Mobile Computing, Applications, and Services*, volume 110 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 328–342. Springer Berlin Heidelberg, Seattle, WA, USA, 11–12 Oct. 2013. DOI:10.1007/978-3-642-36632-1_19. (Cited on page 56.)

[87] T. A. Han and L. M. Pereira. State-of-the-art of intention recognition and its use in decision making. *AI Communications*, 26(2):237–246, Oct. 2013. DOI:10.3233/AIC-130559. (Cited on pages 16, 23, and 151.)

[88] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, second edition, Feb. 2009. ISBN 978-0387848587. (Cited on page 64.)

[89] Z. He and L. Jin. Activity recognition from acceleration data based on discrete cosine transform and svm. In *IEEE International Conference on Systems, Man and Cybernetics (SMC'09)*, pages 5041–5044, San Antonio, TX, USA, 11–14 Oct. 2009. IEEE. DOI:10.1109/ICSMC.2009.5346042. (Cited on page 152.)

[90] A. Hein. Combining unsupervised learning and generative models for activity recognition with wearable sensors. In *Doctoral Consortium Proceedings of the Fifteenth International Symposium on Wearable Computers (ISWC'11)*, San Francisco, CA, USA, 12–15 June 2011. (Cited on page 126.)

[91] A. Hein and T. Kirste. A hybrid approach for recognizing adls and care activities using inertial sensors and rfid. In C. Stephanidis, editor, *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, volume 5615 of *Lecture Notes in Computer Science*, pages 178–188. Springer Berlin Heidelberg, San Diego, CA, USA, 19–24 July 2009. DOI:10.1007/978-3-642-02710-9_21. (Cited on page 155.)

[92] A. Hein, F. Krüger, K. Yordanova, and T. Kirste. Towards causally correct annotation for activity recognition. In M. A. Gerald Bieber and B. Urban, editors, *WOAR 2014: Workshop on Sensor-based Activity Recognition*. Fraunhofer-Verlag, Rostock, Germany, Mar. 2014. ISBN 978-3-8396-0792-3. (Cited on page 55.)

[93] M. Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, July 2006. DOI:10.1613/jair.1705. (Cited on page 43.)

[94] R. Henkel, P. Koldrack, F. Krüger, S. Teipel, and T. Kirste. Supporting situation awareness of dementia patients in outdoor environments. In *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare*, pages 245–248, Istanbul, Turkey, 20–23 May 2015. DOI:10.4108/icst.pervasivehealth.2015.259288. (Cited on page 126.)

[95] L. Hiatt, A. Harrison, and G. Trafton. Accommodating human variability in human-robot teams through theory of mind. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 2077–2071, Barcelona, Spain, 16–22 July 2011. DOI:10.5591/978-1-57735-516-8/IJCAI11-345. (Cited on pages 1, 20, 23, 32, 33, 34, 56, and 120.)

[96] R. M. Hierons, K. Bogdanov, J. P. Bowen, R. Cleaveland, J. Derrick, J. Dick, M. Gheorghe, M. Harman, K. Kapoor, P. Krause, G. Lüttgen, A. J. H. Simons, S. Vilkomir, M. R. Woodward, and H. Zedan. Using formal specifications to support testing. *ACM Computing Surveys (CSUR)*, 41(2):9:1–9:76, Feb. 2009. DOI:10.1145/1459352.1459354. (Cited on pages 17 and 18.)

[97] J. Hightower and G. Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In N. Davies, E. D. Mynatt, and I. Siio, editors, *UbiComp 2004: Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*, pages 88–106. Springer Berlin Heidelberg, Nottingham, England, 7–10 Sept. 2004. DOI:10.1007/978-3-540-30119-6_6. (Cited on pages 13, 17, and 46.)

[98] J. Hoey, P. Poupart, A. v. Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519, May 2010. DOI:10.1016/j.cviu.2009.06.008. (Cited on pages 1, 5, 13, and 23.)

[99] J. Hoey, T. Plötz, D. Jackson, A. Monk, C. Pham, and P. Olivier. Rapid specification and automated generation of prompting systems to assist people with dementia. *Pervasive and Mobile Computing*, 7(3):299–318, June 2011. DOI:10.1016/j.pmcj.2010.11.007. (Cited on pages 1, 23, and 123.)

[100] J. Hoffmann and B. Nebel. The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, May 2001. DOI:doi:10.1613/jair.855. (Cited on page 43.)

[101] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 256–265, Madison, Wisconsin, 24–26 July 1998. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=2074094.2074124. (Cited on pages 16 and 151.)

[102] R. Howey, D. Long, and M. Fox. Val: automatic plan validation, continuous effects and mixed initiative planning using pddl. In *16th IEEE International Conference on Tools with Artificial Intelligence, (ICTAI'04)*, pages 294–301, Boca Raton, FL, USA, 15–17 Nov. 2004. DOI:10.1109/ICTAI.2004.120. (Cited on page 60.)

[103] T. Huỳnh and B. Schiele. Towards less supervision in activity recognition from wearable sensors. In *Proceedings of the 10th IEEE International Symposium on Wearable Computers (ISWC'06)*, pages 3–10, Montreux, Switzerland, 11-14 Oct. 2006. IEEE. DOI:10.1109/ISWC.2006.286336. (Cited on page 152.)

[104] T. Huỳnh, U. Blanke, and B. Schiele. Scalable recognition of daily activities with wearable sensors. In *Location- and context-awareness*, volume 4718 of *Lecture Notes in Computer Science*, pages 50–67. Springer Berlin Heidelberg, Oberpfaffenhofen, Germany, 20–21 Sept. 2007. DOI:10.1007/978-3-540-75160-1_4. (Cited on page 152.)

[105] J. Indulska and K. Henricksen. Context awareness. In A. Helal, M. Mokhtari, and B. Abdulrazak, editors, *The Engineering Handbook of Smart Technology for Aging, Disability, and Independence*, Computer Engineering Series. John Wiley & Sons, Inc., Oct. 2008. DOI:10.1002/9780470379424.ch31. (Cited on pages 4 and 6.)

[106] P. Jachowicz and R. Goebel. Describing plan recognition as nonmonotonic reasoning and belief revision. volume 1342 of *Lecture Notes in Computer Science*, pages 236–245. Springer Berlin Heidelberg, Perth, Australia, 30 Nov.–1 Dec. 1997. DOI:10.1007/3-540-63797-4_76. (Cited on page 14.)

[107] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork. Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, (EMBS'08)*, pages 5250–5253. IEEE, 20–25 Aug. 2008. DOI:10.1109/IEMBS.2008.4650398. (Cited on page 152.)

[108] M. T. Johnson. Capacity and complexity of hmm duration modeling techniques. *IEEE Signal Processing Letters*, 12(5):407–410, May 2005. DOI:10.1109/LSP.2005.845598. (Cited on pages 40 and 154.)

[109] C. B. Jones. *Systematic Software Development Using VDM.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, second edition, Apr. 1990. ISBN 978-0138807337. (Cited on page 18.)

[110] M. Jones, J. Chiang, A. Stephens, and P. J. The use of physical props in motion capture studies. *SAE International Journal of Passenger Cars – Mechanical Systems*, 1(1):1163–1171, 17 June 2009. DOI:10.4271/2008-01-1928. (Cited on page 91.)

[111] L. Kalra, X. Zhao, A. J. Soto, and E. Milios. Detection of daily living activities using a two-stage markov model. *Journal of Ambient Intelligence and Smart Environments*, 5 (3):273–285, 2013. DOI:10.3233/AIS-130208. (Cited on page 23.)

[112] M. Kasparick and F. Krüger. Probabilistic action selection – tracking multiple persons in indoor environments. University Library, University of Rostock, 14 Aug. 2015. (Cited on page 110.)

[113] S. Katz. Assessing self-maintenance: Activities of daily living, mobility, and instrumental activities of daily living. *Journal of the American Geriatrics Society*, 31(12):721–727, Dec. 1983. DOI:10.1111/j.1532-5415.1983.tb03391.x. (Cited on page 76.)

[114] H. A. Kautz. *A Formal Theory of Plan Recognition.* PhD thesis, Dept. of Computer Science, University of Rochester, 1987. (Cited on page 10.)

[115] H. A. Kautz and J. F. Allen. Generalized plan recognition. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI'86)*, pages 32–37, Philadelphia, PA, USA, 11–15 Aug. 1986. Morgan Kaufmann. URL http://www.aaai.org/Papers/AAAI/1986/AAAI86-006.pdf. (Cited on pages 11, 15, 16, and 151.)

[116] P. Kiefer. *Mobile intention recognition.* Springer Berlin Heidelberg, 26 Jan. 2012. ISBN 978-1489988096. DOI:10.1007/978-1-4614-1854-2. (Cited on page 15.)

[117] P. Kiefer and K. Stein. A framework for mobile intention recognition in spatially structured environments. In *Proceedings of the second Workshop on Behaviour Monitoring and Interpretation (BMI)*, pages 28–41, Kaiserslautern, Germany, 2008. URL http://ceur-ws.org/Vol-396/paper03kiefer.pdf. (Cited on pages 15 and 23.)

[118] B. Kim, C. M. Chacha, and J. A. Shah. Inferring team task plans from human meetings: A generative modeling approach with logic-based prior. *Journal of Artificial Intelligence Research*, 52:361–398, Mar. 2015. DOI:10.1613/jair.4496. (Cited on page 76.)

[119] A. Kimmig, L. Mihalkova, and L. Getoor. Lifted graphical models: A survey. *Machine Learning*, 99(1):1–45, Apr. 2015. DOI:10.1007/s10994-014-5443-2. (Cited on page 118.)

[120] T. Kirste. A reference model for situation-aware assistance. In *Proceedings of the Mensch und Computer*, Bad Honneff, Germany, 5–8 Mar. 2001. (Cited on page 1.)

[121] T. Kirste. Detecting high-level team intentions (three person meeting dataset). University Library, University of Rostock, 14 Aug. 2015. (Cited on pages 67, 78, and 83.)

[122] T. Kirste and F. Krüger. CCBM – A tool for activity recognition using Computational Causal Behavior Models. ISSN 0944-5900, Universität Rostock, 2012. (Cited on page 35.)

[123] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Semantic annotation, indexing, and retrieval. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 484–499. Springer Berlin Heidelberg, Sanibel Island, FL, USA, 20–23 Oct. 2003. DOI:10.1007/978-3-540-39718-2_31. (Cited on page 56.)

[124] M. Klaas, N. de Freitas, and A. Doucet. Toward practical N^2 monte carlo: the marginal particle filter. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI'05)*, pages 308–315, Edinburgh, Scotland, 26–29 July 2005. AUAI Press. URL http://uai.sis.pitt.edu/papers/05/p308-klaas.pdf. (Cited on pages 46, 47, and 48.)

[125] P. Krauthausen and U. Hanebeck. Intention recognition for partial-order plans using dynamic bayesian networks. In *12th International Conference on Information Fusion (FUSION'09)*, pages 444–451, Seattle, WA, USA, 6–9 July 2009. URL http://ieeexplore.ieee.org/xpl/login.jsp?arnumber=5203783. (Cited on page 18.)

[126] C. W. Krueger. Software reuse. *ACM Computing Surveys (CSUR)*, 24(2):131–183, June 1992. DOI:10.1145/130844.130856. (Cited on page 18.)

[127] F. Krüger and T. Kirste. Synthesizing sequential bayesian filters for plan and activity recognition from extended precondition-effect rules. In *Adjunct proceedings of the 2nd*

international joint conference on Ambient Intelligence (AMI'11), pages 1–2, Amsterdam, Netherlands, Nov. 2011. Springer Berlin Heidelberg. (Cited on page 35.)

[128] F. Krüger, A. Steiniger, S. Bader, and T. Kirste. Evaluating the robustness of activity recognition using computational causal behavior models. In *Proceedings of the International Workshop on Situation, Activity and Goal Awareness held at Ubicomp'12*, pages 1066–1074, Pittsburgh, PA, USA, 5–8 Sept. 2012. ACM. DOI:10.1145/2370216.2370443. (Cited on pages 75 and 119.)

[129] F. Krüger, K. Yordanova, C. Burghardt, and T. Kirste. Towards creating assistive software by employing human behavior models. *Journal of Ambient Intelligence and Smart Environments*, 4(3):209–226, May 2012. DOI:10.3233/AIS-2012-0148. (Cited on pages 1, 9, 35, 75, and 119.)

[130] F. Krüger, K. Yordanova, and T. Kirste. Tool Support for Probabilistic Intention Recognition using Plan Synthesis. In *Proceedings of the 3rd international joint conference on Ambient Intelligence*, pages 439–444, Pisa, Italy, Nov. 2012. DOI:10.1007/978-3-642-34898-3_40. (Cited on page 35.)

[131] F. Krüger, K. Yordanova, A. Hein, and T. Kirste. Plan synthesis for probabilistic activity recognition. In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence (ICAART'13)*, pages 283–288, Barcelona, Spain, 15–18 Feb. 2013. DOI:10.5220/0004256002830288. (Cited on pages 9, 63, 75, 80, and 119.)

[132] F. Krüger, M. Kasparick, T. Mundt, and T. Kirste. Where are my colleagues and why? Tracking multiple persons in indoor environments. In *10th International Conference on Intelligent Environments (IE'14)*, Shanghai, China, 2–4 July 2014. DOI:10.1109/IE.2014.35. (Cited on pages 9, 63, 75, and 119.)

[133] F. Krüger, M. Nyolt, K. Yordanova, A. Hein, and T. Kirste. Computational state space models for activity and intention recognition. a feasibility study. *PLOS ONE*, 9(11): e109381, Nov. 2014. DOI:10.1371/journal.pone.0109381. (Cited on pages 9, 35, 55, 63, 75, and 119.)

[134] F. Krüger, A. Hein, K. Yordanova, and T. Kirste. Recognising the actions during cooking task (cooking task dataset). University Library, University of Rostock, 2015. (Cited on page 93.)

[135] S. E. Krüger, M. Schafföner, M. Katz, E. Andelic, and A. Wendemuth. Speech recognition with support vector machines in a hybrid system. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech'05)*, pages 993–996, Lisboa, Portugal, 4–8 Sept. 2005. Causal Productions Pty Ltd. URL http://www.isca-speech.org/archive/interspeech_2005/i05_0993.html. (Cited on page 155.)

[136] K. Kunze, M. Barry, E. Heinz, P. Lukowicz, D. Majoe, and J. Gutknecht. Towards recognizing tai chi – an initial experiment using wearable sensors. In *3rd International Forum on Applied Wearable Computing (IFAWC'06)*, pages 1–6, Bremen, Germany, 15–16 Mar. 2006. VDE. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5758288. (Cited on pages 13 and 152.)

[137] C. Ladha, N. Y. Hammerla, P. Olivier, and T. Plötz. Climbax: Skill assessment for climbing enthusiasts. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp'13)*, pages 235–244, Zurich, Switzerland, 8–12 Sept. 2013. ACM. DOI:10.1145/2493432.2493492. (Cited on page 13.)

[138] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, Mar. 1977. DOI:10.2307/2529310. (Cited on pages 99 and 106.)

[139] O. Lara and M. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys Tutorials*, 15(3):1192–1209, July 2013. DOI:10.1109/SURV.2012.110112.00192. (Cited on pages 6, 13, 20, 92, and 152.)

[140] R. J. Lawrence. The lognormal distribution of the duration of strikes. *Journal of the Royal Statistical Society. Series A (General)*, 147(3):464–483, 1984. DOI:10.2307/2981578. (Cited on page 40.)

[141] S.-W. Lee and K. Mase. Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, 1(3):24–32, July 2002. DOI:10.1109/MPRV.2002.1037719. (Cited on page 12.)

[142] A. Legay, B. Delahaye, and S. Bensalem. Statistical model checking: An overview. In *Runtime Verification*, volume 6418 of *Lecture Notes in Computer Science*, pages 122–135. Springer Berlin Heidelberg, St. Julians, Malta, 1–4 Nov. 2010. DOI:10.1007/978-3-642-16612-9_11. (Cited on page 70.)

[143] N. Lesh and O. Etzioni. A sound and fast goal recognizer. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1704–1710, Montreal, Quebec, Canada, 20–25 Aug. 1995. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1643031.1643119. (Cited on pages 15, 16, and 151.)

[144] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, volume 5, pages 766–772, Edinburgh, Scotland, 30 Jun.–5 Aug. 2005. Morgan Kaufmann Publishers Inc. URL http://www.ijcai.org/Proceedings/05/Papers/0888.pdf. (Cited on pages 152 and 155.)

[145] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Cybernetics and Control Theory*, 10:707–710, 1966. (Cited on page 73.)

[146] S. J. Levine and B. C. Williams. Concurrent plan recognition and execution for human-robot teams. In *Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS'14)*, pages 490–498, Portsmouth, NH, USA, 21–26 July 2014. URL http://www.aaai.org/ocs/index.php/ICAPS/ICAPS14/paper/view/7945/8053. (Cited on pages 15, 16, and 151.)

[147] L. Liao, D. Fox, and H. A. Kautz. Learning and inferring transportation routines. In D. L. McGuinness and G. Ferguson, editors, *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'04)*, pages 348–353, San Jose, CA, USA, 25–29 July 2004. URL http://www.aaai.org/Papers/AAAI/2004/AAAI04-056.pdf. (Cited on page 27.)

[148] L. Liao, D. Fox, and H. Kautz. Location-based activity recognition using relational markov networks. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 773–778, Edinburgh, Scotland, 30 Jun.–5 Aug. 2005. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1642293.1642417. (Cited on page 152.)

[149] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *AI*, 171(5-6):311–331, Apr. 2007. DOI:10.1016/j.artint.2007.01.006. (Cited on pages 16, 20, 23, 26, 27, 29, 34, 42, 46, 56, 64, 66, 76, 106, and 155.)

[150] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, Jan. 1991. DOI:10.1109/18.61115. (Cited on page 74.)

[151] V. Lisý, R. Píbil, J. Stiborek, B. Bosanský, and M. Pechoucek. Game-theoretic approach to adversarial plan recognition. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 546–551, Montpellier, France, 27–31 Aug. 2012. IOS Press. DOI:10.3233/978-1-61499-098-7-546. (Cited on pages 15 and 151.)

[152] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *Ubicomp 2007: Ubiquitous Computing*, volume 4717 of *Lecture Notes in Computer Science*, pages 483–500. Springer Berlin Heidelberg, Innsbruck, Austria, 16–19 Sept. 2007. DOI:10.1007/978-3-540-74853-3_28. (Cited on page 152.)

[153] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services (MobySys'09)*, pages 165–178, Kraków, Poland, 22–25 June 2009. ACM. DOI:10.1145/1555816.1555834. (Cited on page 152.)

[154] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pages 4 pp.–116, 3–5 Apr. 2006. DOI:10.1109/BSN.2006.6. (Cited on page 152.)

[155] I. Maurtua, P. T. Kirisci, T. Stiefmeier, M. L. Sbodio, and H. Witt. A wearable computing prototype for supporting training activities in automotive production. In *Proceedings of the 4th International Forum on Applied Wearable Computing (IFAWC'07)*, pages 1–12, Tel Aviv, Israel, 12–13 Mar. 2007. VDE. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5760480. (Cited on page 13.)

[156] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL - the planning domain definition language. Technical Report TR-98-003, Yale Center for Computational Vision and Control, Oct. 1998. (Cited on page 52.)

[157] M. Mernik, J. Heering, and A. M. Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, Dec. 2005. DOI:10.1145/1118890.1118892. (Cited on page 18.)

[158] B. Milner. Inclusion of temporal information into features for speech recognition. In *Proceedings of the Fourth International Conference on Spoken Language (ICSLP'96)*, volume 1, pages 256–259. IEEE, 3–6 Oct. 1996. DOI:10.1109/ICSLP.1996.607093. (Cited on page 45.)

[159] D. Minnen, T. Starner, I. Essa, and C. Isbell. Discovering characteristic actions from on-body sensor data. In *Proceedings of the 10th IEEE international symposium on Wearable computers (ISWC'06)*, pages 11–18, Montreux, Switzerland, 11–14 Oct. 2006. IEEE. URL http://www.cc.gatech.edu/~isbell/papers/isbell-discovery-iswc-2006.pdf. (Cited on page 152.)

[160] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1 Oct. 1997. ISBN 978-0071154673. (Cited on pages 111 and 112.)

[161] P. D. Mosses. *CASL Reference Manual: The Complete Documentation of the Common Algebraic Specification Languages*, volume 2960 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Apr. 2004. DOI:10.1007/b96103. (Cited on page 18.)

[162] K. Mourão, L. S. Zettlemoyer, R. Petrick, and M. Steedman. Learning strips operators from noisy and incomplete observations. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI'12)*, Catalina Island, CA, USA, 15–17 Aug. 2012. URL http://auai.org/uai2012/papers/122.pdf. (Cited on page 125.)

[163] S. M. Müller and A. Hein. Multi–target data association in binary sensor networks. In *Proceedings of the Fifth International Conference on Ambient Computing, Applications, Services and Technologies (AMBIENT'15)*, pages 79 – 84, Nice, France, 19–24 July 2015. URL http://www.thinkmind.org/download.php?articleid=ambient_2015_3_50_40044. (Cited on page 108.)

[164] T. Mundt, F. Krüger, and T. Wollenberg. Who refuses to wash hands? privacy issues in modern house installation networks. In *Proceedings of the Seventh International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA'12)*, pages 271–277, Victoria, BC, Canada, 12–14 Nov. 2012. DOI:10.1109/BWCCA.2012.51. (Cited on pages 108 and 109.)

[165] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012. ISBN 978-0262018029. (Cited on pages 36, 38, and 44.)

[166] P. Natarajan and R. Nevatia. Hierarchical multi-channel hidden semi markov graphical models for activity recognition. *Computer Vision and Image Understanding*, 117(10): 1329–1344, Oct. 2013. DOI:10.1016/j.cviu.2012.08.011. (Cited on page 152.)

[167] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, Vancouver, BC, Canada, 3–8 Dec. 2001. URL http://www.robotics.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf. (Cited on page 10.)

[168] N. Nguyen, H. Bui, S. Venkatsh, and G. West. Recognizing and monitoring high-level behaviors in complex spatial environments. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II–620–5, Madison, Wisconsin, 18–20 June 2003. DOI:10.1109/CVPR.2003.1211524. (Cited on pages 23, 24, and 25.)

[169] N. Nguyen, S. Venkatesh, and H. Bui. Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association. In *Proceedings of the 17th British Machine Vision Conference (BMVC'06)*, pages 1239–1248, Edinburgh, Scotland, 4–7 Sept. 2006. British Machine Vision Association. URL http://dro.deakin.edu.au/view/DU:30044805. (Cited on pages 23, 46, and 47.)

[170] F. Nielson, H. R. Nielson, and C. Hankin. *Principles of Program Analysis*. Springer Berlin Heidelberg, second edition, 10 Dec. 2004. ISBN 978-3540654100. (Cited on page 92.)

[171] M. Nyolt and T. Kirste. On resampling for bayesian filters in discrete state spaces. In *Proceedings of the 27th International Conference on Tools with Artificial Intelligence (ICTAI'15)*, Vietri sul Mare, Italy, 9–11 Nov. 2015. IEEE Computer Society. DOI:10.1109/ICTAI.2015.83. (Cited on page 50.)

[172] M. Nyolt, F. Krüger, K. Yordanova, A. Hein, and T. Kirste. Marginal filtering in large state spaces. *International Journal of Approximate Reasoning*, 61:16–32, June 2015. DOI:10.1016/j.ijar.2015.04.003. (Cited on pages 35 and 49.)

[173] M. Nyolt, K. Yordanova, and T. Kirste. Checking models for activity recognition. In *Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART'15)*, pages 497–502, Lisbon, Portugal, 10–12 Jan. 2015. DOI:10.5220/0005275204970502. (Cited on page 52.)

[174] J. Oh, F. Meneguzzi, and K. Sycara. Probabilistic plan recognition for proactive assistant agents. In G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui, editors, *Plan, activity, and intent recognition*. Elsevier, Amsterdam, The Netherlands, 10 Apr. 2014. ISBN 9780123985323. (Cited on page 23.)

[175] S. Olejnik and J. Algina. Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods*, 8(4):434–447, Dec. 2003. DOI:10.1037/1082-989X.8.4.434. (Cited on pages 74 and 101.)

[176] M. Ostendorf, V. Digalakis, and O. A. Kimball. From hmms to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(5):360–378, 1995. DOI:10.1109/89.536930. (Cited on page 45.)

[177] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct. 2010. DOI:10.1109/TKDE.2009.191. (Cited on page 14.)

[178] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In A. Dey, A. Schmidt, and J. McCarthy, editors, *UbiComp 2003: Ubiquitous Computing*, volume 2864 of *Lecture Notes in Computer Science*, pages 73–89. Springer Berlin Heidelberg, Seattle, WA, USA, 12–15 Oct. 2003. DOI:10.1007/978-3-540-39653-6_6. (Cited on page 27.)

[179] D. J. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic, K. Olson, S. Wang, D. Fox, and H. Kautz. Opportunity knocks: A system to provide cognitive assistance with transportation services. In N. Davies, E. Mynatt, and I. Siio, editors, *UbiComp 2004: Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*, pages 433–450. Springer Berlin Heidelberg, Nottingham, England, 7–10 Sept. 2004. ISBN 978-3-540-22955-1. DOI:10.1007/978-3-540-30119-6_26. (Cited on pages 1, 27, and 42.)

[180] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 44–51, Osaka, Japan, 18–21 Oct. 2005. IEEE. DOI:10.1109/ISWC.2005.22. (Cited on page 92.)

[181] J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, second edition, 14 Sept. 2009. ISBN 978-0521895606. (Cited on pages 11, 120, and 155.)

[182] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton. Activity identification using body-mounted sensors—a review of classification techniques. *Physiological Measurement*, 30(4):R1–33, Apr. 2009. DOI:10.1088/0967-3334/30/4/R01. (Cited on page 154.)

[183] T. J. Prescott, J. J. Bryson, and A. K. Seth. Theme issue 'modelling natural action selection'. *Philosphical Transactions of the Royal Society B – Biological Sciences*, 362(1485):

1521–1721, Sept. 2007. URL http://rstb.royalsocietypublishing.org/content/362/1485. (Cited on page 43.)

[184] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL https://www.R-project.org/. Last visited: 6/21/2016. (Cited on pages 69 and 74.)

[185] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb. 1989. DOI:10.1109/5.18626. (Cited on page 154.)

[186] S. Raghavan, P. Singla, and R. J. Mooney. Plan recognition using statistical models. In G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui, editors, *Plan, activity, and intent recognition*, pages 57–86. Elsevier, Amsterdam, The Netherlands, 10 Apr. 2014. ISBN 9780123985323. (Cited on pages 15 and 151.)

[187] M. Ramírez and H. Geffner. Plan recognition as planning. In *Proceedings of the 21th International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1778–1783, Pasadena, CA, USA, 14–17 July 2009. URL http://ijcai.org/papers09/Papers/IJCAI09-296.pdf. (Cited on pages 15 and 30.)

[188] M. Ramírez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the 24th National Conference of Artificial Intelligence (AAAI'10)*, pages 1211–1217, Atlanta, Georgia, USA, 11–15 July 2010. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/download/1821/2160. (Cited on pages 16, 22, 23, 30, 64, 90, 99, 151, and 155.)

[189] M. Ramírez and H. Geffner. Goal recognition over pomdps: inferring the intention of a pomdp agent. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 2009–2014, Barcelona, Spain, 16–22 July 2011. AAAI Press. URL http://ijcai.org/papers11/Papers/IJCAI11-335.pdf. (Cited on pages 12, 15, 16, 20, 23, 30, 31, 33, 34, 39, 42, 43, 52, 56, 120, and 151.)

[190] C. Randell and H. Muller. Context awareness by analysing accelerometer data. In *Proceedings of the Fourth International Symposium on Wearable Computers (ISWC'00)*, pages 175–176, Atlanta, GA, USA, 16–17 Oct. 2000. DOI:10.1109/ISWC.2000.888488. (Cited on page 152.)

[191] N. Ravi, N. Dandekar, P. Mysore, and M. 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, Pittsburgh, Pennsylvania, 9–13 July 2005. AAAI Press. URL http://www.aaai.org/Papers/AAAI/2005/IAAI05-013.pdf. (Cited on pages 13 and 152.)

[192] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics (TACL)*, 1:25–36, Mar. 2013. URL http://www.aclweb.org/anthology/Q13-1003. (Cited on page 56.)

[193] S. Richter and M. Westphal. The lama planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research (JAIR)*, 39:127–177, Sept. 2010. DOI:10.1613/jair.2972. (Cited on pages 52 and 126.)

[194] N. D. Rodríguez, M. P. Cuéllar, J. Lilius, and M. D. Calvo-Flores. A survey on ontologies for human behavior recognition. *ACM Computing Surveys*, 46(4):43:1–43:33, Apr. 2014. DOI:10.1145/2523819. (Cited on page 12.)

[195] D. Roetenberg, H. Luinge, and P. Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. Technical report, 3 Apr. 2013. URL https://www.xsens.com/images/stories/PDF/MVN_white_paper.pdf. (Cited on page 92.)

[196] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*, pages 25–36, Providence, RI, USA, 16–21 June 2012. DOI:10.1109/CVPR.2012.6247801. (Cited on page 90.)

[197] F. Rossi, editor. *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, Menlo Park, California, 3–9 Aug. 2013. AAAI Press. (Cited on page 23.)

[198] P. Roy, B. Bouchard, A. Bouzouane, and S. Giroux. A hybrid plan recognition model for alzheimer's patients: interleaved-erroneous dilemma. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07)*, pages 131–137, Fremont, CA, USA, 2–5 Nov. 2007. DOI:10.1109/IAT.2007.17. (Cited on page 12.)

[199] P. C. Roy, S. Giroux, B. Bouchard, A. Bouzouane, C. Phua, A. Tolstikov, and J. Biswas. A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimer's patients. In L. Chen, C. D. Nugent, J. Biswas, and J. Hoey, editors, *Activity Recognition in Pervasive Intelligent Environments*, volume 4 of *Atlantis Ambient and Pervasive Intelligence*, pages 33–58. Atlantis Press, May 2011. DOI:10.2991/978-94-91216-05-3_2. (Cited on pages 16, 23, and 25.)

[200] S. Russell and P. Norvig. *Artificial Intelligence – A Modern Approach*. Prentice Hall, third edition, 16 Feb. 2010. ISBN 978-0136042594. (Cited on pages 3, 42, and 43.)

[201] S. Saad, D. De Beul, S. Mahmoudi, and P. Manneback. An ontology for video human movement representation based on benesh notation. In *Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS'12)*, pages 77–82, Tangier, Morocco, 10–12 May 2012. IEEE. DOI:10.1109/ICMCS.2012.6320129. (Cited on page 57.)

[202] A. Sadilek and H. Kautz. Location-based reasoning about complex multi-agent behavior. *Journal of Artificial Intelligence Research (JAIR)*, 43(1):87–133, Jan. 2012. DOI:10.1613/jair.3421. (Cited on pages 20, 23, 24, 25, 33, 34, 66, 76, and 120.)

[203] F. Sadri. Logic-based approaches to intention recognition. In N.-Y. Chong and F. Mastrogiovanni, editors, *Handbook of Research on Ambient Intelligence: Trends and Perspectives*, pages 346–375. IGI Global, May 2011. DOI:10.4018/978-1-61692-857-5.ch018. (Cited on pages 16 and 151.)

[204] B. Satzger and O. Kramer. Goal distance estimation for automated planning using neural networks and support vector machines. *Natural Computing*, 12(1):87–100, Mar. 2013. DOI:10.1007/s11047-012-9332-y. (Cited on pages 43 and 126.)

[205] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, Mar. 1978. URL http://www.jstor.org/stable/2958889. (Cited on page 155.)

[206] C. Schwering, D. Beck, S. Schiffer, and G. Lakemeyer. Plan recognition by program execution in continuous temporal domains. In *KI 2012: Advances in Artificial Intelligence*, volume 7526 of *Lecture Notes in Computer Science*, pages 156–167. Springer Berlin Heidelberg, Saarbrücken, Germany, 24–27 Sept. 2012. DOI:10.1007/978-3-642-33347-7_14. (Cited on pages 15 and 151.)

[207] P. Seibel. *Practical Common Lisp.* Apress, 1 Apr. 2005. ISBN 978-1590592397. (Cited on page 51.)

[208] R. Selten. What is bounded rationality? In G. Gigerenzer and R. Selten, editors, *Bounded Rationality: The Adaptive Toolbox.* MIT Press, Mar. 2001. ISBN 978-0262072144. (Cited on page 3.)

[209] A. Serna, H. Pigot, and V. Rialle. Modeling the progression of alzheimer's disease for cognitive assistance in smart homes. *User Modeling and User-Adapted Interaction*, 17: 415–438, Sept. 2007. DOI:10.1007/s11257-007-9032-y. (Cited on page 90.)

[210] A. K. Seth, T. J. Prescott, and J. J. Bryson. *Modelling natural action selection.* Cambridge University Press, 29 Sept. 2012. ISBN 978-1107000490. (Cited on page 126.)

[211] S. Shaphiro and M. Wilk. An analysis of variance test for normality. *Biometrika*, 52(3): 591–611, June 1965. DOI:10.1093/biomet/52.3-4.591. (Cited on page 74.)

[212] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation networks for recognition of partially ordered sequential action. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, pages 862–869, Washington, D.C., USA, 27 June–2 July 2004. DOI:10.1109/CVPR.2004.1315255. (Cited on pages 18, 23, 25, 46, 48, and 121.)

[213] C. L. Sidner. Plan parsing for intended response recognition in discourse. *Computational intelligence*, 1(1):1–10, Apr. 1985. DOI:10.1111/j.1467-8640.1985.tb00054.x. (Cited on page 15.)

[214] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.* John Wiley & Sons, July 2006. ISBN 978-0-471-70858-2. (Cited on page 45.)

[215] I. Sommerville. *Software Engineering.* Pearson/Prentice Hall, ninth edition, Mar. 2010. ISBN 978-3868940992. (Cited on page 18.)

[216] D. Spinellis. Notable design patterns for domain-specific languages. *Journal of Systems and Software*, 56:91–99, Feb. 2010. DOI:10.1016/S0164-1212(00)00089-3. (Cited on pages 51 and 53.)

[217] J. M. Spivey. *The Z notation.* Oriel College, Oxford, OX1 4EW, England, second edition, 1998. (Cited on page 18.)

[218] E. Spriggs, F. de La Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 17–24, Miami, Florida, USA, 20–25 June 2009. DOI:10.1109/CVPRW.2009.5204354. (Cited on page 56.)

[219] J. Stadermann and G. Rigoll. A hybrid svm/hmm acoustic modeling approach to automatic speech recognition. In *Proceedings of the Eighth International Conference on Spoken Language Processing (ICSLP'04)*, volume 10, pages 661–664, Jeju Island, Korea, 4–8 Oct.

2004. URL http://www.isca-speech.org/archive/interspeech_2004/i04_0661.html. (Cited on page 155.)

[220] T. Starner, J. Weaver, and A. Pentland. A wearable computer-based american sign language recogniser. *Personal Technologies*, 1(4):241–250, 1997. DOI:10.1007/BF01682027. (Cited on page 152.)

[221] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 729–738, Zurich, Switzerland, 8–12 Sept. 2013. ACM. DOI:10.1145/2493432.2493482. (Cited on pages 13 and 152.)

[222] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, (2):42–50, Apr. 2008. DOI:10.1109/MPRV.2008.40. (Cited on pages 13 and 152.)

[223] M. Stikic and K. Van Laerhoven. Recording housekeeping activities with situated tags and wrist-worn sensors: Experiment setup and issues encountered. In *Proceedings of the first International Workshop on Wireless Sensor Networks for Health Care (WSNHC'07)*, Braunschweig, Germany, 6 June 2007. URL http://publica.fraunhofer.de/documents/N-66514.html. (Cited on page 56.)

[224] M. Stikic, D. Larlus, S. Ebert, and B. Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2521–2537, Feb. 2011. DOI:10.1109/TPAMI.2011.36. (Cited on page 152.)

[225] B. Stroustrup. *The C++Programming Language*. Addison Wesley, fourth edition, 28 May 2013. ISBN 978-0321563842. (Cited on pages 51 and 53.)

[226] D. P. Strum, M.D., J. H. May, and L. G. Vargas. Modeling the uncertainty of surgical procedure times: Comparison of log-normal and normal models. *Anesthesiology*, 92(4):1160–1167, Apr. 2000. URL http://anesthesiology.pubs.asahq.org/article.aspx?articleid=1945938. (Cited on page 40.)

[227] M. Suda. Property directed reachability for automated planning. *Journal of Artificial Intelligence Research*, 50:265–319, June 2014. DOI:doi:10.1613/jair.4231. (Cited on page 52.)

[228] G. Sukthankar. *Activity Recognition for Agent Teams*. PhD thesis, Robotics Institute School of Computer Science Carnegie Mellon University, July 2007. URL http://www.cs.cmu.edu/~gitars/gsukthankar-thesis.pdf. (Cited on page 12.)

[229] G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui, editors. *Plan, activity, and intent recognition: Theory and practice*. Elsevier, Amsterdam, The Netherlands, 10 Apr. 2014. ISBN 9780123985323. (Cited on pages 20, 23, and 151.)

[230] G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui. Introduction. In G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui, editors, *Plan, activity, and intent recognition*, pages xix–xxxv. Elsevier, Amsterdam, The Netherlands, 10 Apr. 2014. ISBN 9780123985323. (Cited on pages 10 and 16.)

[231] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Linz/Vienna, Austria, 21–23 Apr. 2004. DOI:10.1007/978-3-540-24646-6_10. (Cited on page 152.)

[232] D. Tecuci and B. W. Porter. Memory based goal schema recognition. In *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference (FLAIRS'09)*, Sanibel Island, FL, USA, 19–21 May 2009. URL https://www.aaai.org/ocs/index.php/FLAIRS/2009/paper/viewFile/114/254. (Cited on page 151.)

[233] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, Mar. 2011. DOI:10.1126/science.1192788. (Cited on page 42.)

[234] M. Torchiano. *effsize: Efficient Effect Size Computation*, 2016. URL https://CRAN.R-project.org/package=effsize. R package version 0.6.2. (Cited on page 74.)

[235] G. Trafton, L. Hiatt, A. Harrison, F. Tamborello, S. Khemlani, and A. Schultz. Act-r/e: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, 2(1):30–55, 2013. URL http://www.hri-journal.org/index.php/HRI/article/view/121. (Cited on page 32.)

[236] Ubisense. Real-time location systems (RTLS) and geospatial consulting. URL http://www.ubisense.net. http://www.ubisense.net, Retrieved: 24th January 2012. (Cited on page 79.)

[237] T. L. M. van Kasteren. *Activity Recognition for Health Monitoring Elderly using Temporal Probabilistic Models*. PhD thesis, Universiteit van Amsterdam, 27 Apr. 2011. URL http://hdl.handle.net/11245/1.333749. (Cited on pages 2 and 12.)

[238] T. L. M. van Kasteren and B. J. A. Kröse. A sensing and annotation system for recording datasets in multiple homes. In *Proceedings of the 27 Annual Conference on Human Factors and Computing Systems workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research.*, pages 4763–4766, Boston, USA, Apr. 2009. (Cited on pages 6 and 56.)

[239] T. L. M. van Kasteren, A. Noulas, G. Englebienne, and B. J. A. Kröse. Accurate activity recognition in a home setting. In *Proceedings of the tenth International Conference on Ubiquitous Computing (Ubicomp'08)*, pages 1–9, Seoul, Korea, 21–24 Sept. 2008. ACM, ACM. DOI:10.1145/1409635.1409637. (Cited on pages 56 and 152.)

[240] A. Vargha and H. D. Delaney. A critique and improvement of the "cl" common language effect size statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics*, 25(2):101–132, 2000. URL http://www.jstor.org/stable/1165329. (Cited on page 74.)

[241] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer Berlin Heidelberg, New York, fourth edition, 2002. ISBN 978-0-387-95457-8. (Cited on page 69.)

[242] L. Vinh, S. Lee, H. Le, H. Ngo, H. Kim, M. Han, and Y.-K. Lee. Semi-markov conditional random fields for accelerometer-based activity recognition. *Applied Intelligence*, 35(2): 226–241, 2011. DOI:10.1007/s10489-010-0216-5. (Cited on page 152.)

[243] T. Walsh, editor. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, Menlo Park, California, 2011. AAAI Press. (Cited on page 23.)

[244] L. Wang, T. Gu, X. Tao, H. Chen, and J. Lu. Recognizing multi-user activities using wearable sensors in a smart home. *Pervasive and Mobile Computing*, 7(3):287–298, June 2011. DOI:10.1016/j.pmcj.2010.11.008. (Cited on page 6.)

[245] J. A. Ward, P. Lukowicz, and H. W. Gellersen. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology*, 2(1):6:1–6:23, Jan. 2011. DOI:10.1145/1889681.1889687. (Cited on page 73.)

[246] M. Weiser. The computer for the 21st century. *SIGMOBILE Mobile Computing and Communications Review*, 3(3):3–11, July 1999. ISSN 1559-1662. DOI:10.1145/329124.329126. (Cited on page 1.)

[247] T. Westeyn, K. Vadas, X. Bian, T. Starner, and G. D. Abowd. Recognizing mimicked autistic self-stimulatory behaviors using hmms. In *Proceedings of the ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 164–167, Osaka, Japan, 18–21 Oct. 2005. IEEE. DOI:10.1109/ISWC.2005.45. (Cited on page 152.)

[248] D. H. Wilson and C. Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 62–79. Springer Berlin Heidelberg, Munich, Germany, 8–13 May 2005. DOI:10.1007/11428572_5. (Cited on pages 6, 17, 107, and 108.)

[249] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. ELAN: a professional framework for multimodality research. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, pages 1556–1559, Genoa, Italy, 24–26 May 2006. URL http://lrec.elra.info/proceedings/lrec2006/pdf/153_pdf.pdf. (Cited on page 59.)

[250] J. Wobbrock, A. Wilson, and Y. Li. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07)*, pages 159–168, Newport, Rhode Island, USA, 7–10 Oct. 2007. ACM Press. DOI:10.1145/1294211.1294238. (Cited on page 12.)

[251] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV'07)*, pages 1–8, Rio de Janero, Brazil, 14–21 Oct. 2007. IEEE. DOI:10.1109/ICCV.2007.4408865. (Cited on pages 23 and 92.)

[252] D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th national conference on Artificial Intelligence (AAAI'05)*, pages 21–27, Pittsburgh, Pennsylvania, 9–13 July 2005. AAAI Press. URL https://www.aaai.org/Papers/AAAI/2005/AAAI05-004.pdf. (Cited on page 152.)

[253] A. Y. Yang, R. Jafari, S. S. Sastry, and R. Bajcsy. Distributed recognition of human actions using wearable motion sensor networks. *Journal of Ambient Intelligence and Smart Environments*, 1(2):103–115, Apr. 2009. DOI:10.3233/AIS-2009-0016. (Cited on page 152.)

[254] Q. Yang, K. Wu, and Y. Jiang. Learning actions models from plan examples with incomplete knowledge. In *Proceeding of the 15th International Conference on Automated Planning and Scheduling (ICAPS'05)*, pages 241–250, Monterey, CA, USA, 5–10 July 2005. URL https://www.aaai.org/Papers/ICAPS/2005/ICAPS05-025.pdf. (Cited on page 125.)

[255] K. Yordanova. Toward a unified human behaviour modelling approach. Technical Report CS-02-11, Institut für Informatik, Universität Rostock, Rostock, Germany, May 2011. (Cited on pages 51 and 52.)

[256] K. Yordanova. *Methods for Engineering Symbolic Human Behaviour Models for Activity Recognition*. PhD thesis, Institute of Computer Science, Rostock, Germany, 2 Apr. 2014. (Cited on pages 39 and 81.)

[257] K. Yordanova. From textual instructions to sensor-based recognition of user behaviour. In *Companion Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI'16)*, pages 67–73, Sonoma, CA, USA, 7–10 Mar. 2016. DOI:10.1145/2876456.2879488. (Cited on page 125.)

[258] K. Yordanova and T. Kirste. A process for systematic development of symbolic models for activity recognition. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5 (4), Dec. 2015. DOI:10.1145/2806893. (Cited on pages 54 and 67.)

[259] S.-Z. Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, Feb. 2010. DOI:10.1016/j.artint.2009.11.011. (Cited on page 154.)

[260] Z. Yu and Y. Nakamura. Smart meeting systems: A survey of state-of-the-art and open issues. *ACM Computing Surveys*, 42(2):8:1–8:20, Mar. 2010. DOI:10.1145/1667062.1667065. (Cited on page 1.)

[261] Z. Zhang, P. Webster, V. Uren, A. Varga, and F. Ciravegna. Automatically extracting procedural knowledge from instructional texts using natural language processing. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 520–527, Istanbul, Turkey, 21–27 May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/244_Paper.pdf. (Cited on page 125.)

[262] C. Zhu and W. Sheng. Human daily activity recognition in robot-assisted living using multi-sensor fusion. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2009. ICRA'09.*, pages 2154–2159, Kobe, Japan, 12–17 May 2009. IEEE. DOI:10.1109/ROBOT.2009.5152756. (Cited on page 152.)

[263] H. H. Zhuo and S. Kambhampati. Action-model acquisition from noisy plan traces. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, pages 2444–2450, Beijing, China, 3–9 Aug. 2013. AAAI. URL http://www.ijcai.org/Proceedings/13/Papers/359.pdf. (Cited on page 66.)

[264] H. H. Zhuo and Q. Yang. Action-model acquisition for planning via transfer learning. *Artificial Intelligence*, 212:80–103, July 2014. DOI:10.1016/j.artint.2014.03.004. (Cited on page 125.)

[265] V. Ziparo, L. Iocchi, M. Leonetti, and D. Nardi. A probabilistic action duration model for plan selection and monitoring. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, pages 4716–4721, Taipei, Taiwan, Oct. 2010. DOI:10.1109/IROS.2010.5650106. (Cited on page 40.)

A
Details

A.1. Additional Information about Related Work

	Reference	*F.keyhole*	*F.probability*	*F.plan.synth*	*F.direct*	*M.accuracy*	*M.convergence*
1	Ramírez and Geffner [189][87]	■	■	■	▲	■	□
2	Ramírez and Geffner [188][87]	■	■	■	▲	□	□
3	Baker et al. [17][188]	■	■	■	▲	□	□
4	Bouchard et al. [31][-]	■	□	■	▲	□	□
5	Lesh and Etzioni [143][11]	■	□	□	⧫	□	■
6	Kautz and Allen [115][11]	■	□	□	★	□	□
7	Horvitz et al. [101][11]	■	■	□	⧫	□	□
8	Geib and Goldman [77][11]	■	■	□	⧫	□	□
9	Lisỳ et al. [151][15]	□	■	■	⧫	□	□
10	Raghavan et al. [186][229]	■	■	■	⧫	■	■
11	Blaylock and Allen [26][229]	■	■	□	⧫	□	■
12	Tecuci and Porter [232][-]	■	□	□	⧫	□	■
13	Cohen et al. [53][203]	□	□		⧫	□	□
14	Geib and Goldman [76][203]	■	■	■	⧫	□	□
15	Levine and Williams [146][-]	■	□	□	▼	□	□
16	Schwering et al. [206][-]	■	■	■	⧫	□	□
17	Avrahami-Zilberbrand and Kaminka [15][229]	■	□	□	▲	□	□

Table A.1.: Overview of plan recognition aproaches. The superscript reference in the first column gives the original source. "■"=Feature included, "□"=Feature not included, "⧫"=interaction, "▲"=simulated, "★"=specified manually, "▼"=accurate sensors

	Reference	N.sensors	N.classes	Classifier	Scenario	M.accuracy
1	Bao and Intille [18][58]	IMU	20	DT, NB	ADL	■
2	Ravi et al. [191][40]	IMU	8	DT, SVM, NB	ADL	■
3	Duong et al. [69][-]	V	6	HSMM	ADL	■
4	Yang et al. [253][-]	IMU	13	Classifier	ADL	■
5	Casale et al. [45][-]	IMU	5	RF	ADL	■
6	Dernbach et al. [62][-]	IMU	15	ANN	ADL	■
7	Borzeshi et al. [30][-]	V	14	HMM	Kitchen	■
8	Natarajan and Nevatia [166][-]	V	9	HMM	Gestures	■
9	Stein and McKenna [221][-]	IMU, V	10	NB, RF	ADL	■
10	Logan et al. [152][40]	PlaceLab	43	NB, DT	ADL	■
11	Tapia et al. [231][47]	Binary	35	NB	ADL	■
12	Stiefmeier et al. [222][40]	IMU, UWB	46	SM	Industry	■
13	Kunze et al. [136][40]	IMU	3	KNN	Sports	□
14	Minnen et al. [159][40]	IMU	6	Motif	Sports	■
15	Starner et al. [220][40]	V	40	HMM	Gestures	■
16	Liao et al. [148][40]	GPS	6	CRF	ADL	■
17	van Kasteren et al. [239][40]	RS	8	CRF, HMM	ADL	■
18	Huỳnh et al. [104][40]	IMU	16	SVM, K-Means, KNN	ADL	■
19	Bulling et al. [39][40]	EOG	5	SVM	Office	■
20	Lester et al. [144][40]	IMU, BP, A	10	HMM, NB, DT	ADL	■
21	Amft et al. [7][40]	IMU	4	HMM	Gestures	■
22	Lu et al. [153][40]	A	3	HMM, DT	ADL	■
23	Westeyn et al. [247][40]	IMU	7	HMM	Medical	■
24	Buettner et al. [35][40]	RFID	14	HMM	ADL	□
25	Jatoba et al. [107][139]	IMU	6	DT, KNN, NB	ADL	■
26	Maurer et al. [154][139]	IMU, A, T	6	DT	Location	■
27	Ermes et al. [71][139]	IMU	5	DT	Location	■
28	Randell and Muller [190][139]	IMU	6	ANN	Location	■
29	He and Jin [89][139]	IMU	4	SVM	Location	■
30	Vinh et al. [242][139]	IMU	4	CRF	ADL	□
31	Stikic et al. [224][139]	IMU	20	MG, SVM	ADL	■
32	Ali et al. [5][139]	IMU	5	MES	Surgery	■
33	Huỳnh and Schiele [103][139]	IMU	8	MES, SVM	Location	■
34	Zhu and Sheng [262][139]	IMU	4	HMM, ANN	Location	■
35	Cheng et al. [48][139]	Capacitive	11	LDA	Location	■
36	Brdiczka et al. [33][47]	V	5	SVM	Location	■
37	Wyatt et al. [252][47]	RFID	26	KLD	ADL	■

Table A.2.: Overview of AR approaches. The superscript reference in the first column gives the original source. "■"=Feature included, "□"=Feature not included, sensor codes: IMU=Inertial Measurement Unit, MD=Motion Detector, PM=Pressure Mat, BB=Break Beam, RS=Reed Switch, BP=Barometric Pressure, A=Audio, V=Video, IR=Infrared, US=Ultrasound, LR=Laser Range Finder, RFID=Radio Frequency identification, T=Temperature,
classifier codes: KNN=k-nearest neighbour, LDA=linear discriminant analysis, KLD=Kullback Leibler Divergence, CRF=Conditional Random Field, ANN=artificial Neural Network, RF=Random Forest, SM=String Matching, MG=Multigraph, MES=Multiple Eigenspaces

A.2. Example

Example A.1: Example of a CCBM domain and problem file.

The CCBM domain specifies a maze, where person are moving in. The maze has 5×5 cells and a person can move from one cell to each adjacent cell. The CCBM problem definition specifies one person "nora", who is initially located in the upper left cell ($x = 1, y = 1$). Five potential goals are provided, one for each cell in the bottom row ($y = 1$).

```
(define (domain maze)
    (:types person)

    (:functions
        (x-position-of ?p - person) - (number 1 5)
        (y-position-of ?p - person) - (number 1 5))

    (:action step
        :parameters (?p - person ?x1 ?y1 ?x2 ?y2 - (number 1 5))
        :agent ?p
        :precondition (and
            (= (x-position-of ?p) ?x1)
            (= (y-position-of ?p) ?y1)
            (or
                (and (= ?x1 ?x2) (= ?y1 ?y2)) ; stay
                (and (= ?x1 ?x2) (= ?y1 (+ ?y2 1))) ; walk down
                (and (= ?x1 ?x2) (= ?y1 (- ?y2 1))) ; walk up
                (and (= ?y1 ?y2) (= ?x1 (+ ?x2 1))) ; walk left
                (and (= ?y1 ?y2) (= ?x1 (- ?x2 1))) ; walk right
            ))
        :effect (and
            (assign (x-position-of ?p) ?x2)
            (assign (y-position-of ?p) ?y2)
        )))
```

```
(define (problem maze) (:domain maze)
    (:objects nora - person)

    (:init
        (= (y-position-of nora) 1)
        (= (x-position-of nora) 1))

    (:goals
        (= g1 (and (= (y-position-of nora) 5) (= (x-position-of nora) 1)))
        (= g2 (and (= (y-position-of nora) 5) (= (x-position-of nora) 2)))
        (= g3 (and (= (y-position-of nora) 5) (= (x-position-of nora) 3)))
        (= g4 (and (= (y-position-of nora) 5) (= (x-position-of nora) 4)))
        (= g5 (and (= (y-position-of nora) 5) (= (x-position-of nora) 5)))
    ))
```

A.3. Construction of Hidden Markov Models

Beside the application of approximative inference, CCBM allows the construction of HMMs from the specified LTS. HMMs [185] are a special case of DBNs. They allow inference to be done exactly, but have requirements for the state space in return. The exact representation of the belief state requires the state space to be finite and small enough to be computed by matrix multiplication. In HMMs, the system model is represented by a so-called transition matrix with both dimensions of the size of the state space. In addition, HMMs do not allow the use of custom state duration functions. State durations in HMMs have to be represented by the geometric distribution, the discrete specialisation of the exponential distribution, and included in the diagonal of the transition matrix. Several extensions exist, allowing state durations beyond the geometric distribution. Expanded State HMM [108] for example introduce combinations of geometric distribution by expanding single states to topologies of states. Hidden Semi Markov models [259], on the other hand, break the Markov property by introducing custom state durations. However, all extensions increase the complexity of the state space and therefore increase inference complexity, which in return prevents exact solutions.

In order to handle the state space, it is reduced by trimming the (A, D, G, S, U) tuple to (A, G, S) triples by removing all timing related nodes from the DBN. This is possible since the geometrical distribution is memoryless: the probability of remaining in a state is constant. In HMMs, states often represent actions rather than system states [182]. In contrast, CCBM constructs the state space of HMMs from combinations of system states and grounded actions. The state transition probabilities are computed by using a reduced version of the action selection function γ, introduced in Section 3.1.4. In it, all λ parameters, except for λ_δ are set to 0, resulting in goal-directed transition probabilities. Consequently, the causality and contextual information, inherent to the CCBM approach are maintained, in contrast to a simple translation of actions to HMM states. Another limitation is the restriction to models with a single execution slot. The reason for this decision is also to restrict the state space complexity as it typically grows exponential in the number of agents. Due to the simplification of the state space, the inference algorithms for estimating the state sequence reduce to matrix multiplication. The belief state B is represented by a vector of length of the size of the state space.

Since counting the transitions from actions based on the causal model as described is only possible if the state space is completely expanded, this technique is viable for small state spaces only. In fact, the use of HMM-based inference would not satisfy the requirement for latent infinity. This technique can therefore only be used with problems of very limited complexity.

A.4. Notes on Intention Recognition

Intention recognition – recognising that a given goal G is the objective of an actor – is based on the assumption that different goals will give rise to different action sequences. As described in Section 3.1.4, the selection of an action is determined by the action selection model, which considers the goal as one influencing feature. While this allows to improve action selection for a known goal, it also enables reasoning about different goals. In general, two contrary approaches to goal recognition exist: inter-model and intra-model goal recognition.

Inter-model goal recognition Firstly, the inter-model approach, which uses a set of models, each parameterised with a different goal. During inference, each model is applied to the observation sequence and used to infer the state sequence. The likelihood $p(y_{1:t})$ of the model – an indicator, how well the model fits the data – is then used to select the most likely model.

Ramírez and Geffner [188] apply the inter-model design for their planner-based goal recognition approach. Armentano and Amandi [12] learn different models for different goals, apply the observation sequence, and eventually use a likelihood-based comparison to detect the correct goal.

Intra-model goal recognition The second approach uses an intra-model design and includes different goals within one model. At inference time, each goal influences action selection in different directions. After application of the correction step, which weights different hypotheses according to the sensor model, this results in different goal probabilities. Thus, a posterior goal density is constructed, which allows reasoning about different goals. Different approaches to intra-model goal recognition exist [26, 43, 17]. They either sample the goal at the beginning from a prior goal density and consider it fixed afterwards [43], or allow the goal to be changed by a goal transition function. The latter is used by Baker et al. [17] and Liao et al. [149].

Comparison of both approaches While both approaches have been shown to achieve reasonable results, they differ from both, the statistical and the computational point of view. The inter-model goal recognition approach relates the likelihood, computed from different models, and determines the goal distribution. This is modelled by defining action selection conditional to G.

From the computational point of view, the inter-model approach allows to distribute the calculation among various processes, one for each goal, without any additional effort. The statistical point of view, in contrast, introduces two issues. The first being that the likelihood of an model usually depends on the complexity (the degrees of freedom) of the model. Common approaches for likelihood-based model selection are the Bayesian Information Criterion [205] and the Akaike Information Criterion [3]. Both approaches penalise the degree of freedom of the model, which requires the full state space to be expanded.

The second, more serious issue is that it requires the sensor model to provide the observation probabilities $p(y \mid x)$. While this is typically easy, as this *"(. . .) often can be drawn directly from our experiential knowledge (. . .)"*,([181, p.5]), it becomes painful when using sensor observation models based on discriminate classifiers such as SVMs or DTs. These classifiers provide, if at all, a conditional action class density $p(c \mid y)$. While Bayes' rule allows to flip this relation, providing the prior observation density $p(y)$ still remains uncomputable. This is no problem, as long as the resulting likelihoods are not compared to each other in order to construct model densities. Thus, the prior observation density $p(y)$ can be ignored as constant factor.

The literature provides different ways of combining generative and discriminative classifiers [219, 91, 135]. However, currently only few work exist that addresses the model selection problem based on combinations of generative and discriminative classifiers. Lester et al. [144] for example, fitted sigmoids to the output of different classifiers, allowing to convert them into probability distributions, which are then fed into the temporal classifier.

The intra-model approach, on the other hand, encodes different goals into one model. This results in a linear increase of the belief state and a quadratic increase of the transition matrix. While exact methods can hardly handle this growth, approximate techniques have to increase the number of samples. This increases the computational requirements. Combining different goals into one model results in a sound statistical model (as introduced in Section 3.1). The action class density, computed by applying discriminate classifiers, can be interpreted as likelihood by multiplying the prior class density and then directly fed to the inference process as observation probabilities. Both, the intra- and the inter-model goal recognition approaches are provided within the CCBM toolbox. Intra-model goal recognition is applied in Experiment X1.

B Additional Information about Experiments

B.1. Experiment X1

Time	A.Seen	A.X	A.Y	B.Seen	B.X	B.Y	C.Seen	C.X	C.Y
0	0	0	0	0	0	0	1	150.473	-143.161
0.216202	0	0	0	0	0	0	1	157.175	-169.946
0.216202	0	0	0	0	0	0	0	157.175	-169.946
0.648499	0	0	0	0	0	0	1	188.268	-199.87
1.4052	1	193.107	-82.845	0	0	0	0	188.268	-199.87
1.6214	1	230.089	-36.153	0	0	0	0	188.268	-199.87
1.9457	0	230.089	-36.153	0	0	0	1	187.265	-196.157
2.5402	1	250.928	77.934	0	0	0	0	187.265	-196.157
2.5942	0	250.928	77.934	0	0	0	1	174.379	-81.656
2.7563	1	253.329	121.539	0	0	0	0	174.379	-81.656
2.8104	0	253.329	121.539	0	0	0	1	190.982	-30.724
2.9725	1	266.518	133.157	0	0	0	0	190.982	-30.724
3.0266	0	266.518	133.157	0	0	0	1	212.95	-3.945
3.1347	0	266.518	133.157	1	143.319	-170.781	0	212.95	-3.945
3.1887	1	292.253	147.207	0	143.319	-170.781	0	212.95	-3.945
3.2428	0	292.253	147.207	0	143.319	-170.781	1	205.842	2.184

Table B.1.: Extract of the preprocessed position data from the first meeting dataset $\mathcal{D}_1$. For each person, three columns contain x and y coordinates relative to one corner of the room and a flag that signals whether the position was updated or carried from the last time slice.

	Agent	Action	Distribution	ll	Parameter1	Parameter2
1	A	enter	gamma	-25.66	11.91	3.84
2	A	moveDoorSeat	gamma	-34.41	71.24	0.76
3	A	sit	normal	-180.84	1423.58	453.03
4	A	moveSeatStage	weibull	-32.75	11.58	95.02
5	A	present	normal	-91.00	686.10	22.90
6	A	moveStageSeat	cauchy	-89.24	97.14	8.15
7	A	moveSeatDoor	normal	-74.96	80.55	10.27
8	A	exit	normal	2.09	0.95	0.22
9	A	discuss	weibull	-100.85	9.54	324.31
10	A	moveDoorStage	gamma	-46.98	3.96	0.11
11	B	enter	lognormal	-25.11	1.09	0.29
12	B	moveDoorSeat	weibull	-82.12	1.83	64.26
13	B	sit	lognormal	-222.62	6.69	0.40
14	B	moveSeatStage	lognormal	-70.67	4.55	0.16
15	B	present	weibull	-115.19	16.69	1042.75
16	B	moveStageSeat	weibull	-86.40	5.56	92.88
17	B	moveSeatDoor	normal	-75.36	76.50	10.48
18	B	exit	weibull	5.05	24.66	2.99
19	B	discuss	weibull	-100.94	9.57	326.83
20	B	moveDoorStage	cauchy	-9.35	99.44	0.86
21	C	enter	lognormal	-28.27	1.04	0.35
22	C	moveDoorSeat	weibull	-65.90	1.70	54.66
23	C	sit	weibull	-190.20	3.50	1637.19
24	C	moveSeatStage	lognormal	-52.56	4.51	0.11
25	C	present	weibull	-119.32	7.22	615.74
26	C	moveStageSeat	cauchy	-88.56	87.73	6.06
27	C	moveSeatDoor	normal	-75.17	73.10	10.38
28	C	exit	weibull	-3.25	27.27	4.98
29	C	discuss	weibull	-100.48	9.91	328.29
30	C	moveDoorStage	normal	-23.17	95.17	11.51

Table B.2.: Selected probability density functions for each action including their parameters. Parameter meanings: gamma(α, β), normal(μ, σ), weibull(k, λ), cauchy(x,γ), lognormal($\log(\mu)$, $\log(\sigma)$)

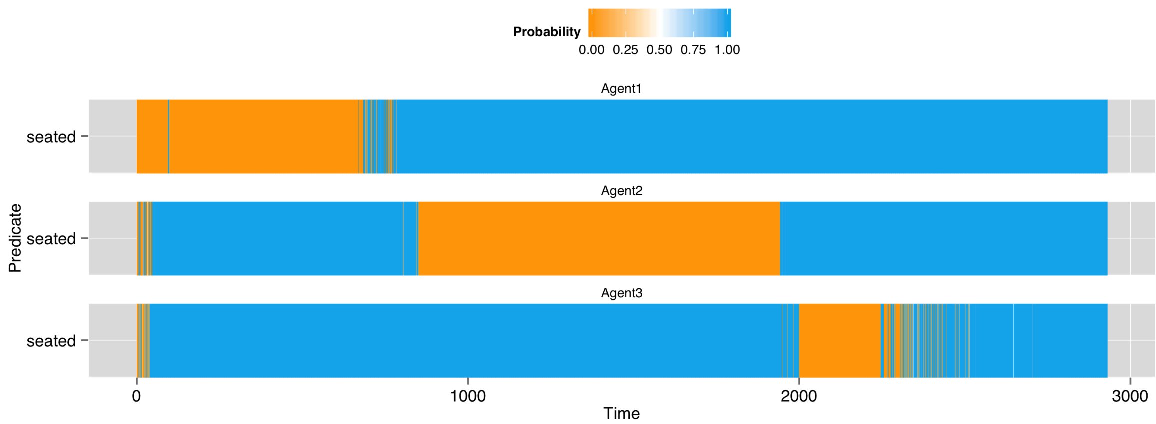

Figure B.1.: The probability of each agent for being seated at each time-step. The probability is encoded by the color. Blue represents probabilities above .5 and orange probabilities below .5. White represents the probability .5. The different presentation phases can be seen, as for these phases two agents have a probability of > .5 for seated and one agent < .5.

B.2. Experiment X2

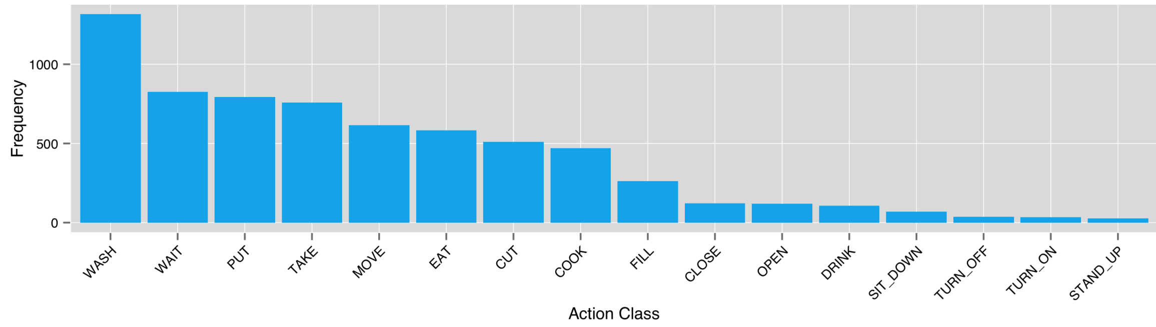

Figure B.2.: Frequencies of actions in the dataset. The action WASH (1316) is the most frequent action whereas STAND_UP (26) is least frequent.

Duration	Distance	Mode	t	p_t	M_t	V	p_V	M_V	p_{SW}	d	A
τ_c	f_δ	CMf	3.14	.02	0.037	27	.031	0.036	.043	$-.7$	.27
τ_d	f_δ	CMf	7.87	$< .001$	0.067	28	.016	0.070	.14	-1.17	.16
τ_c	$f_{\bar{\delta}}$	CMf	5.78	.001	0.066	28	.016	0.069	.12	-1.08	.2
τ_d	$f_{\bar{\delta}}$	CMf	9.37	$< .001$	0.098	28	.016	0.109	.081	-1.43	.061
τ_c	f_h	CMf	3.61	.011	0.043	27	.031	0.042	.43	$-.77$	.27
τ_d	f_h	CMf	7.67	$< .001$	0.064	28	.016	0.064	.79	-1.09	.18
τ_c	f_δ	CMs	6.75	$< .001$	0.063	28	.016	0.068	.078	-1.06	.2
τ_d	f_δ	CMs	6.70	$< .001$	0.083	28	.016	0.087	.12	-1.28	.1
τ_c	$f_{\bar{\delta}}$	CMs	6.71	$< .001$	0.082	28	.016	0.087	.16	-1.26	.1
τ_d	$f_{\bar{\delta}}$	CMs	5.88	.001	0.102	28	.016	0.108	.27	-1.4	.061
τ_c	f_h	CMs	7.14	$< .001$	0.060	28	.016	0.064	.12	-1.03	.2
τ_d	f_h	CMs	6.27	$< .001$	0.080	28	.016	0.092	.044	-1.22	.14

Table B.3.: Performance comparison of different CCBM configurations with appropriate HMM configuration. All CCBM configurations are based on (CM, O21s, L1). Paired t-test and Wilcoxon signed rank tests were used for comparison. For all t-tests, $df = 6$. p_{SW} gives the p-value for the Shapiro-Wilk normality test.

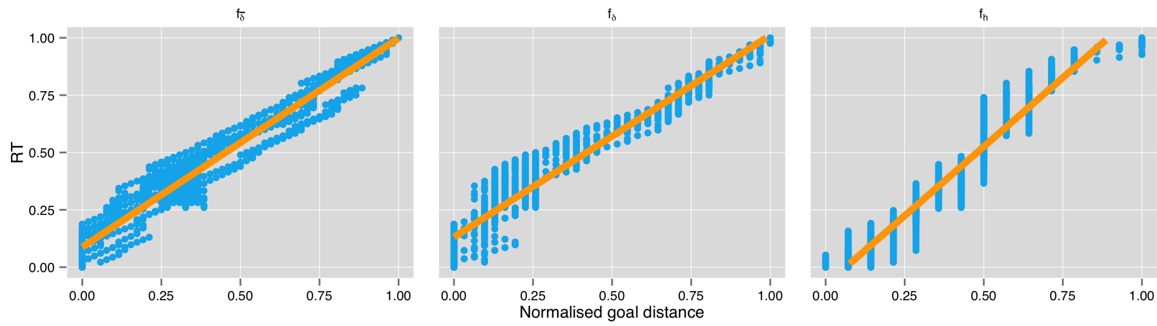

Figure B.3.: Linear Models fitted to predict the normalised remaining time to goal (RT) from the normalised goal distances for different goal distance approximations.

	counter	cupboard	cutting_board	hands	plate	stove	pot	sink	table	void	# locations
bottle	•			•							2
cutting_board	•			•							2
food	•		•	•	•		•			•	6
glass	•	•		•				•	•		5
knife	•			•				•	•		4
plate	•	•		•				•	•		5
pot	•			•		•		•			4
self	•							•	•		3
stir_spoon	•			•			•	•			4
sponge				•				•			2
spoon	•			•	•			•	•		5

Table B.4.: Value domains of location slot by domain object for the kitchen experiment. By considering the locations of the domain objects, the number of potential combinations is 1,152,000.

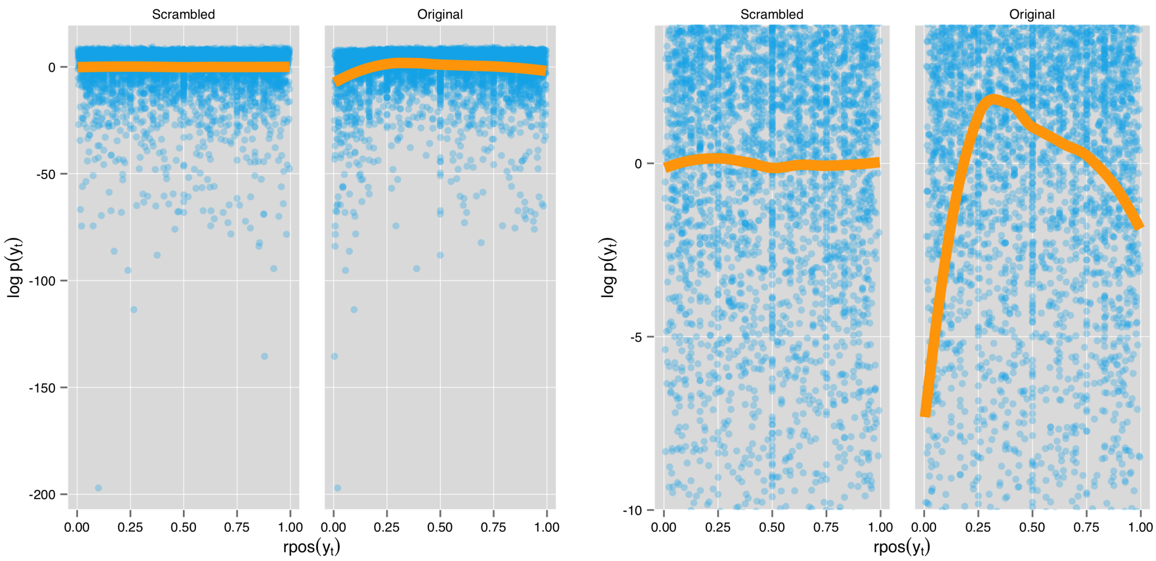

Figure B.4.: Effect of scrambling on (expected) log probability of observations vs. normalised relative run position. The orange lines are computed by locally weighted regression using the `loess` function in R. Right plot: detail of left plot.
$rpos(y_t)$ gives the relative position of the observation with respect to the action class. For each observation sequence of the same action class c the relative position is computed by normalising the index to the interval (0,1). The expected (log) probability is computed by use of the action observation model described in Section 6.2.2.1. The local regression curves represents an approximation of the expected values. A centering effect can be observed in the original data which cannot be observed in the scrambled data.

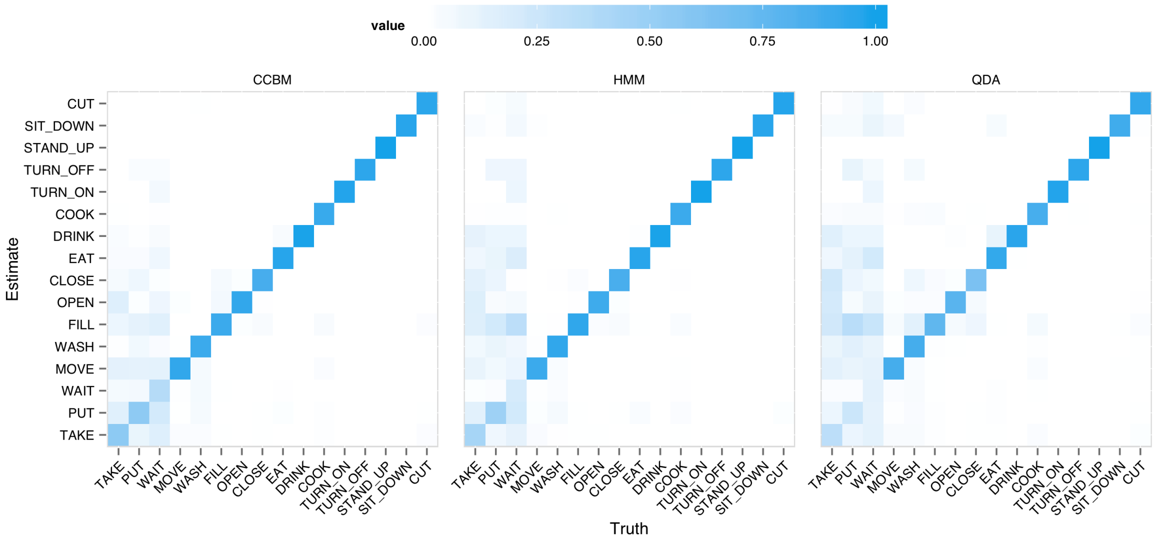

Figure B.5.: Confusion matrices for QDA, HMM, and CCBM. For each annotated activity, the probability of classification from sensor data is provided. Blue represents high and white low probabilities. The blue diagonal represents correct classification.

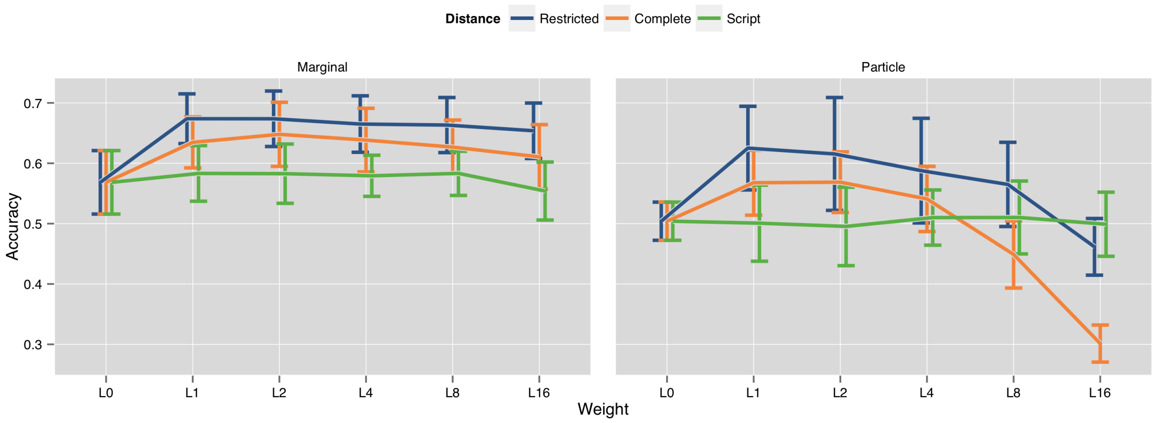

Figure B.6.: Interactions between Mode, Distance, **and** Weight. Error bars give 95% confidence interval due to *between subject* variance. Effect comparisons are based on within subject differences.

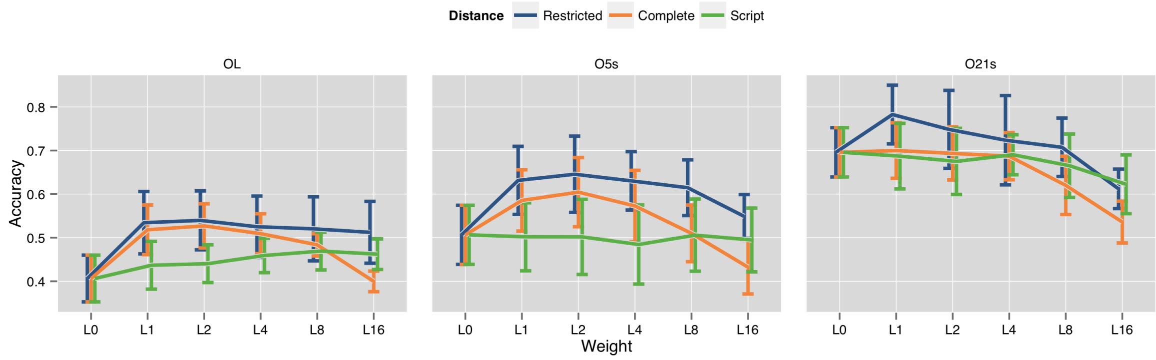

Figure B.7.: Interactions between Observations, Distance, and Weight. Error bars give 95% confidence interval due to *between subject* variance. Effect comparisons are based on within subject differences.

	available	*clean*	*cooked*	*filled*	*hungry*	*location*	*on*	*open*	*prepared*	*seated*	*thirsty*
bottle						•		•			
cupboard								•			
cutting_board						•					
food		•	•			•			•		
glass		•		•		•					
hands	•	•									
knife						•					
plate		•				•					
pot		•				•					
self					•	•				•	•
sponge						•					
spoon		•				•					
stir_spoon		•				•					
stove							•				

Table B.5.: The domain objects and their slots for the kitchen experiment. All slots have boolean value domains with two exceptions: $\mathsf{available}(\mathsf{hands})$ is an integer. The actions implement the constraint $0 \leq \mathsf{available}(\mathsf{hands}) \leq 2$. $\mathsf{location}(\mathit{object})$ is a symbolic value. Allowed values for the different objects are given in Table B.4 (these constraints are again implemented by the actions).

Predictor	β_0	β_1	$F_{(1,958)}$	r^2	$F_{(958,958)}$
$f_{\bar{\delta}}$ (Restricted)	0.09	0.91	17579	.95	1.00
f_{δ} (Complete)	0.13	0.88	13441	.93	0.78
f_h (Script)	-0.08	1.20	10029	.91	0.59

Table B.6.: Detailed properties of linear models in Figure B.3 Properties of linear models for predicting relative remaining time (RT) from different distance heuristics using $\mathsf{RT} = \beta_0 + \beta_1 \mathsf{Predictor}$. (For all F, $p < .001$)

Step	Time	Action	Step	Time	Action
1	0	wash hands	46	511	sit-down
2	35	wait	47	536	take spoon plate
3	43	move sink counter	48	541	eat
4	64	take food counter	49	616	put spoon plate
5	70	move counter sink	50	619	take glass table
6	77	wash food	51	624	drink
7	111	move sink counter	52	631	wait
8	117	take knife counter	53	638	drink
9	122	put food cutting-board	54	647	put glass table
10	126	cut food	55	653	take spoon plate
11	173	put knife counter	56	657	eat
12	177	wait	57	692	put spoon plate
13	184	take knife+cutting-board counter	58	699	take plate+glass table
14	191	fill food cutting-board pot	59	705	stand-up
15	204	put knife+cutting-board counter	60	710	move table sink
16	211	take pot counter	61	724	wait
17	219	put pot stove	62	729	put glass sink
18	228	wait	63	735	wait
19	230	turn-on stove	64	759	put plate sink
20	235	wait	65	767	take sponge sink
21	247	take wooden-spoon counter	66	773	take spoon plate
22	256	cook	67	777	wash spoon
23	334	put wooden-spoon pot	68	797	wait
24	336	turn-off stove	69	809	wash spoon
25	341	wait	70	816	put spoon sink
26	349	open cupboard	71	822	take plate sink
27	360	take plate+glass cupboard	72	825	wash plate
28	376	put plate+glass counter	73	874	put plate sink
29	389	take pot stove	74	881	take glass sink
30	395	fill food pot plate	75	884	wash glass
31	416	put pot stove	76	919	put glass sink
32	422	wait	77	926	put sponge sink
33	427	take bottle counter	78	929	move sink counter
34	433	open bottle	79	935	take pot stove
35	441	fill water bottle glass	80	939	move counter sink
36	453	close bottle	81	943	put pot sink
37	463	put bottle counter	82	946	take wooden-spoon pot
38	467	wait	83	948	take sponge sink
39	469	take plate counter	84	950	wash wooden-spoon
40	474	wait	85	964	put wooden-spoon sink
41	481	take spoon counter	86	970	take pot sink
42	485	put spoon plate	87	973	wash pot
43	488	take glass counter	88	1020	put pot sink
44	491	move counter table	89	1027	put sponge sink
45	506	put plate+glass table	90	1028	(done)

Table B.7.: Action sequence of subject S1 (aLTS annotations).

Class	n	Model	LL	Parameter 1	Parameter 2
Base	143	lognormal	−410.61	1.9447	0.6112
COOK	9	weibull	−44.10	1.3574	56.3776
CUT	7	gamma	−32.36	8.0067	0.1101
DRINK	11	weibull	−21.55	6.6757	10.4570
EAT	13	lognormal	−59.49	3.5710	0.6610
FILL	21	weibull	−62.79	2.8518	14.0273
MOVE	54	lognormal	−162.58	2.3149	0.4852
PUT	162	lognormal	−355.67	1.4638	0.5030
STAND_UP	7	weibull	−10.01	4.2093	4.0999
TAKE	165	lognormal	−358.01	1.3774	0.5344
WASH	49	gamma	−207.42	1.5952	0.0594

Table B.8.: Duration models selected for action classes. Parameter meanings: gamma(α, β), weibull(k, λ), lognormal(log(μ), log(σ))

Effect	F value		p value		η^2_G	
Mode	$F_{(1,6)}$	= 168.93	< .001	*	.34	*
Observations	$F_{(2,12)}$	= 34.51	< .001	*	.61	*
Distance	$F_{(2,12)}$	= 23.23	< .001	*	.14	*
Weight	$F_{(5,30)}$	= 39.25	< .001	*	.18	*
Duration	$F_{(1,6)}$	= 1.76	.23		.004	
Mode:Observations	$F_{(2,12)}$	= 79.17	< .001	*	.24	*
Mode:Distance	$F_{(2,12)}$	= 13.68	< .001	*	.033	
Observations:Distance	$F_{(4,24)}$	= 3.34	.026	*	.03	
Mode:Weight	$F_{(5,30)}$	= 19.65	< .001	*	.089	*
Observations:Weight	$F_{(10,60)}$	= 13.46	< .001	*	.079	*
Distance:Weight	$F_{(10,60)}$	= 20.39	< .001	*	.1	*
Mode:Duration	$F_{(1,6)}$	= 1.63	.25		.001	
Observations:Duration	$F_{(2,12)}$	= 10.48	.002	*	.025	
Distance:Duration	$F_{(2,12)}$	= 0.65	.54		.002	
Weight:Duration	$F_{(5,30)}$	= 4.67	.003	*	.015	
Mode:Observations:Distance	$F_{(4,24)}$	= 5.32	.003	*	.008	
Mode:Observations:Weight	$F_{(10,60)}$	= 2.62	.01	*	.013	
Mode:Distance:Weight	$F_{(10,60)}$	= 22.30	< .001	*	.077	*
Observations:Distance:Weight	$F_{(20,120)}$	= 2.49	.001	*	.018	
Mode:Observations:Duration	$F_{(2,12)}$	= 0.16	.85		.0003	
Mode:Distance:Duration	$F_{(2,12)}$	= 24.06	< .001	*	.01	
Observations:Distance:Duration	$F_{(4,24)}$	= 2.67	.057		.009	
Mode:Weight:Duration	$F_{(5,30)}$	= 6.69	< .001	*	.022	
Observations:Weight:Duration	$F_{(10,60)}$	= 1.49	.17		.008	
Distance:Weight:Duration	$F_{(10,60)}$	= 3.00	.004	*	.008	
Mode:Observations:Distance:Weight	$F_{(20,120)}$	= 4.76	< .001	*	.026	
Mode:Observations:Distance:Duration	$F_{(4,24)}$	= 0.90	.48		.002	
Mode:Observations:Weight:Duration	$F_{(10,60)}$	= 0.61	.8		.004	
Mode:Distance:Weight:Duration	$F_{(10,60)}$	= 4.19	< .001	*	.009	
Observations:Distance:Weight:Duration	$F_{(20,120)}$	= 2.34	.002	*	.013	
Mode:Observations:Distance:Weight:Duration	$F_{(20,120)}$	= 1.54	.081		.008	

Table B.9.: Significance of effects of CCBM configuration factors on Accuracy**, using 216** CMf / CPf **configurations. (2 modes, 3 observations, 3 distances, 6 weights, 2 durations.)**

B.3. Experiment X3

Time	Action	Agent	Location
4	climbing	agent1	stairs
8	walking	agent1	corr3
12	walking	agent1	corr4
14	walking	agent1	room214a
109	working	agent1	room214a
115	walking	agent1	corr4
121	walking	agent1	corr3
123	walking	agent1	corr2
142	posterWatching	agent1	corr2
146	walking	agent1	corr2
151	walking	agent1	corr1
155	walking	agent1	room207
172	makeCoffee	agent1	room207
174	walking	agent1	room207
181	walking	agent1	corr1
185	walking	agent1	corr2
192	walking	agent1	corr3
197	walking	agent1	corr4
199	walking	agent1	room214a
265	working	agent1	room214a
269	walking	agent1	corr4
273	walking	agent1	corr5
276	walking	agent1	room218
350	meeting	agent1	room218
356	walking	agent1	corr5
359	walking	agent1	corr4
360	walking	agent1	room214a
376	working	agent1	room214a
382	walking	agent1	corr4
387	walking	agent1	corr3
391	walking	agent1	corr2
395	walking	agent1	corr1
398	walking	agent1	room207
399	walking	agent1	room208
427	conversation	agent1	room208
429	walking	agent1	room207
436	walking	agent1	corr1
440	walking	agent1	corr2
446	walking	agent1	corr3
450	walking	agent1	corr4
452	walking	agent1	room214a
466	working	agent1	room214a

Table B.10.: Overview of the annotation sequence of the first iteration for one participating person.

Trial	Transition	Accuracy	t	p	M	p_{SW}	d
T1	f_δ	.85	-20	$< .001$	$-.06$	.58	-8.04
T2	f_δ	.79	$-.069$	.95	$-.0008$	.26	$-.041$
T3	f_δ	.61	3.1	$< .001$	.16	.97	11.8
T4	f_δ	.63	19.1	$< .001$	.19	.26	7.27
T5	f_δ	.58	8.81	$< .001$	.27	.41	5.89
T6	f_δ	.45	37.9	$< .001$	.41	.36	22.1
T7	f_δ	.57	13.1	$< .001$	.28	.57	7.47
T1	f_{HMM}	.85	-93.6	$< .001$	$-.059$	.58	-6.97
T2	f_{HMM}	.79	-	-	-	-	0
T3	f_{HMM}	.76	.99	.38	.0004	.97	.025
T4	f_{HMM}	.82	-1.18	.3	$-.001$	.26	$-.065$
T5	f_{HMM}	.85	1.65	.17	.0003	.41	.035
T6	f_{HMM}	.87	6.62	.003	.0009	.36	.24
T7	f_{HMM}	.85	.85	.44	.001	.57	.12

Table B.11.: Comparison of the CCBM to the HMM recognition accuracies for the different trial runs. Comparison using paired t-test (all df = 4). p_{SW} gives the results of the Shapiro-Wilk test of normality. The "-" signals that due to equal results executing tests was not meaningful.

Theses

1. Efficient sensor-based reconstruction of causal structures of human behaviour can be achieved by employing causal models of human behaviour in combination with probabilistic Bayesian inference with the Marginal Filter.

2. Computational Causal Behaviour Models allow inference based on causal behaviour models. CCBM simultaneously estimates activities, context information, and the goal from location data with models of similar complexity as related approaches while achieving recognition rates at the same level as baseline classifiers.

3. CCBM achieves successful state estimation of everyday activities with large state spaces (containing hundreds of millions of states).

4. All modelling factors have relevant influence on recognition performance in CCBM-based inference.

5. The action sequences of multiple cooperative agents can be reconstructed by employing CCBM with recognition rates at the same level as baseline classifiers.

6. CCBM achieves recognition rates for fine grained activity recognition from wearable sensors that are comparable to baseline classifiers.

7. The Marginal Filter achieves better recognition rates than the Particle Filter in large categorical state spaces with sparse transition matrices due to more efficient usage of the resources.

8. A causal behaviour model that was created for one specific application domain can be reused in a different scenario of the same application domain.

9. A causal behaviour model that was developed for inference from action observation can be reused for inference based on state observation and vice versa without changes.

10. A causal behaviour model can be reused for different numbers of persons.

Curriculum Vitae

Name:	Frank Krüger
Date of Birth:	19 December 1980
Place of Birth:	Rostock, Germany
Nationality:	German

Education, Academic, and Work Experience

11/2009 –	Researcher at the University of Rostock Rostock, Germany
05/2006 – 10/2009	Software developer at FORTecH Software GmbH Rostock, Germany
10/2000 – 05/2006	Diploma in Computer Science at the University of Rostock, Sanitz, Germany Degree: Diploma in Computer Science
08/1992 – 06/2000	High school education at the Gymnasium Sanitz, Sanitz, Germany Degree: Abitur

Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst habe. Ich habe keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt. Die aus fremden Quellen wörtlich oder inhaltlich entnommenen Stellen sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Rostock, December 12, 2016 Frank Krüger