Abstracting from Observation-equivalent Entities in Human Behavior Modeling

Unnamed Author
Unnamed Institution
Unnamed Country
Unnamed E-mail Address

Abstract

Recognizing human behavior from noisy and ambiguous sensor data is a prerequisite for many applications such as context-aware assistance. The sensor data, however, often do not allow to distinguish between multiple entities, e. g. a presence sensor does not allow to distinguish between two persons i. e. both are observation-equivalent. Conventional algorithms, however, consider each of these entities separately during the inference of human behavior, leading to a high computational burden in scenarios where a large number of entities have to be considered. Therefore, these algorithms can only be applied to very limited scenarios. We analyzed the challenges appearing in these scenarios and revealed that considering observation-equivalent entities separately is one reason for the huge computational effort. Thus, we propose to exploit observation-equivalence by representing entities as a group and inferring about these groups of entities. We sketch a mechanism that exploits observation-equivalencies which we call *lifted probabilistic inference*. To compare this approach with conventional inference approaches, we adapted an office scenario from the literature so that it contains observation-equivalent entities and simulated a corresponding dataset. This dataset can be used as a benchmark for the evaluation of different inference approaches with respect to observation-equivalence. We compare the number of states this approach, and a conventional inference algorithm is considering during inference on this benchmark dataset. On average, the conventional approach uses almost 200,000 states to cover the situations of the scenario during the inference whereas our lifted probabilistic inference approach uses less than 100 states. Thus, an observation-equivalent approach seems promising for a more efficient inference in scenarios with many observation-equivalent entities.

1 Introduction

Recognizing human behaviour from noisy and ambiguous sensor data is an important prerequisite for context-aware applications, such as assistive systems. The sensor data, also called observations, are time series of sensor measurements which are used to infer plans, activities, or intentions of humans acting in their environment. A plan consists of a detailed series of actions the human performs to achieve its desired intention. Within the observed environment, often there are not only humans involved, but also objects which we both subsume under the term entities. Thus, entities execute actions with possibly other entities involved to reach a specific goal. Often these entities can not be distinguished from each other given the observation data. To illustrate this, we want to consider the following two examples:

1. **Surveillance Scenario.** A scenario that is frequently used to evaluate inference mechanisms is that of a surveillance system (Bui 2003; Nguyen et al. 2003). A single person is acting in an environment with two rooms that are connected by a corridor. The rooms contain several points of interest, such as a computer station, a printer, and a library. The person is tracked by 5 cameras distributed in the environment and her position is returned as coordinates of a virtual grid. The goal of the person is to print a document or to use the library. This scenario, however, does not cover many real world situations, as only a single person acting in the environment is observed.
2. **Office Scenario.** An adaption of the previous scenario that introduces a varying number of persons is described in (Krüger et al. 2012; Yordanova 2014). Up to three persons act within an office environment according to individual goals. The environment contains a printer as well as a coffee machine which both may be out of the corresponding resources (water and ground coffee or paper, respectively). The printer can, additionally, be jammed. The goal of each person is to have a cup of coffee, print some documents, or both. Other than the first scenario, the observation data was recorded by sensor mats indicating the presence of at least one person at a location of interest.

These two scenarios model the real world in a very limited way, by reducing many parameters like the number of acting persons, to be computable. The reason for these strict limitations is that the computational effort of behaviour inference grows exponentially when the number of entities is increased. However, in some cases, these entities are not distinguishable from each other, given the observation data, because independently of the acting entity, the executed action sequence produces the exact same sequence of observations. Conventional algorithms do not consider these aspects during the recognition of the behavior from the sequence of observations. We propose to exploit these aspects by grouping entities together that potentially produce the exact same sequence of observations and performing the inference using this grouped entities, instead of considering every individual entity.

Our contribution in this paper is threefold:

1. We analyze the challenges in the scenarios resulting from observation-equivalence.
2. We sketch a mechanism exploiting observation-equivalence which we call *lifted probabilistic inference*
3. We simulate a dataset, based on an adapted office scenario, that incoporates observation-equivalent entities. Using this dataset, we make a first comparision of the *lifted probabilistic inference* and a conventional inference approach.

The results of this comparison are very promising and suggest that exploiting observation-equivalence has the potential for a more efficient inference than conventional approaches.

After summarizing current inference methods in Section 2, we analyze the challenges in the scenarios in Section 3. The lifted probabilistic inference is sketched in Section 4. Section 5 summarizes publicly avaiblable dataset according to the revealed challenges and derives our adapted benchmark dataset. Furthermore, the lifted probabilistic inference is compared with the conventional appraoch. We finish this paper with our conlusion and future work (Section 6).

2 Inference

In this section, we briefly introduce the core concepts of activity recognition and establish the Bayesian Filtering framework that builds the basis for the inference mechanisms used.

The goal of activity recognition is, given a sequence of observations $y_{1:t}$, to estimate the sequence of actions $a_{1:t}$ that have been performed, resulting in these observations. In order to do so, a description of human behavior is needed that models the (human) actions including the change of the situation after executing them. Precondition/effect-languages such as STRIPS (Fikes and Nilsson 1971) or PDDL (Mcdermott et al. 1998) enable such a symbolic description of dynamic systems. These languages describe the state of the world by a set of predicates that may be changed by the application of an action. A predicate is a named binary variable indicating if the fact from the name is true or not, e. g. `Alice_is_at_Room_A`. As an example if the human behavior model is used to describe the movements of 6 persons in 10 rooms, 60 predicates are generated describing every combination of persons and rooms. Because every possible combination will be generated, we say that these states are *grounded*. The mechanisms in (Baker, Saxe, and Tenenbaum 2009; Ramírez and Geffner 2011; Hiatt, Harrison, and Trafton 2011) are based on grounded states.

However, because in precondition/effect-languages, the actions are applied deterministic and discrete, these languages cannot be be used directly when there is uncertainty such as from the sensor data. A state of the art mechanism that is used to solve this is the framework of Bayesian Filtering. In Bayesian Filtering, we estimate the sequence of states $x_{1:t}$, that includes both the model state and action. Assuming that each observation depends only on the current state of the world, which can be expressed by the distribution $p(y_t \mid x_t)$, and that the current state of the world only depends on the previous state of the world $p(x_t \mid x_{t-1})$, the estimation of $x_{1:t}$ can be performed recursively: at step t, at first the next state x_{t+1} is predicted using

$$p(x_{1:t+1} \mid y_{1:t}) = p(x_{t+1} \mid x_t) p(x_{1:t} \mid y_{1:t}) \tag{1}$$

Second, the prediction is updated by taking the next observation y_{t+1} into account.

$$p(x_{1:t+1} \mid y_{1:t+1}) = \frac{p(y_{t+1} \mid x_{t+1}) p(x_{1:t+1} \mid y_{1:t})}{p(y_{t+1} \mid y_{1:t})} \tag{2}$$

(Krüger 2016; Krüger et al. 2014) give an overview over different approaches for activity recognition using Bayesian Filtering, as well as the resulting state space sizes. Most of the approaches use simple scenarios where the size of the state space remains relatively small. However, more complex scenarios can result in very large state spaces, e. g. 146 million states for the approach described in (Nyolt et al. 2015). In these scenarios, exact inference algorithms are infeasible. Therefore, approximate inference algorithms (e. g. particle filters) are used.

3 Challenges

Although the two scenarios described in Section 1 are modeling the real world in a very limited way, they reveal some interesting and challenging characteristics: 1. Observation-equivalence, 2. Similar Entities, 3. Entity Properties, 4. Interactions of Entities, and 5. Change of the Entity's Properties. All those are discussed below, by giving specific examples and describing them in detail. Furthermore, we formulate challenges appearing from this characteristics.

Observation-equivalence. *The sensor mats in the office scenario observe the movement of 3 persons. These persons have the potential to produce the exact same sequence of observations, independent of the particular person that is moving. For example, the movement of Person 1 from Room A to C compared to the movement of Person 3 from Room A to C whereas the other persons remain in Room A cannot be distinguished, given the sensor data. Thus, the observations are equivalent in this part of the observation sequence. However, this observation-equivalence must not be valid all the time. If we extend the scenario by additional identifying sensory inputs recorded by an access control card reader, at the time a particular person uses her access control card, this person can be distinguished from the others. Thus, we do know that e. g. Alice used her access control card, but we still cannot distinguish the other persons.*

Many sensors do not allow to identify a particular entity from a group of entities, because every member of the group produces the exact same sequence of sensor readings while executing the same sequence of actions. Thus, we say this sequence of actions is observation-equivalent with respect to the group of entities, or the group of entities is observation equivalent with respect to a sequence of actions. Please note, observation-equivalence must not be valid for the whole sequence of observations but only for a sub-sequence.

Challenge 1. Due to non-identifying sensors, different entities involved in recognition tasks may produce identical sensor data. An inference mechanism should be aware of observation-equivalence and should be able to group the relevant entities for a section of the observation sequence. Furthermore, the inference mechanism should exploit these abstraction to groups during the inference.

Similar Entities. *The particular sheet of paper which is used for refilling the printer or the actual printer that is used for printing a document is not always necessary to know. It is only important to know that there is paper or that a printer was used.*

Recognizing a particular entity out of a group of similar entities is not always needed. In these cases, it can be abstracted from the particular entities, even if the sensor data allows an identification. At the same time we might need to count the number of entities available, e. g. to order new paper.

Challenge 2. The inference mechanism should have the ability to group entities and exploit this abstract representation during the computation, independent of the observation data. At the same time we may need to track the cardinality of these groups.

Entity Properties. *The 3 persons are similar entities that can be modeled by two properties:* location *and* holding. *E. g. one person is at Room A, the other two are at Room B, but none of them holds something in their hands. The other way round, we can say that the three persons do not hold something in their hands and their location is Room B for two of them and Room A for the third person.*

Entities are usually non-atomic, i. e. they possess an internal structure, which can for example be implemented as a map from property names to values. Similar entities usually share a common structure with similar values for some of the properties. The exact values of the properties are not necessarily known or observable precisely, but only estimates can be made based on some underlying probability distribution. In addition, the distribution may not be independent within one entity or within a group of entities. E. g. the location of two persons is not independent, because they cannot be at exactly the same position.

Challenge 3. An inference engine should be able to represent and manipulate properties of entities. It should be able to reason about dependencies between properties of a single entity and between properties of a group of entities.

Interactions of Entities. *Ground coffee and water are transformed into freshly brewed coffee and coffee dregs after the corresponding button at the coffee machine was pushed by a person. The water can be refilled from a water faucet and coffee can be drunk.*

Different entities may be necessary to execute a given action, and as an effect of that action the entities might be transformed, be destroyed, or new entities may be created. A transformation may also update a single property only, e.g., a change of the location.

Challenge 4. Entities interact with each other, which may result in the creation, transformation or destruction of entities. The inference mechanism should be able to represent and reason about these interactions.

Change of the Entity's Properties. *E. g. the access control card reader adds a new property to the entity standing in front of it, namely the name of the person. It may also be necessary to remove irrelevant properties from an entity to allow grouping as described above.*

In addition to the transformation of entities due to actions as described above, the internal structure of some entity may also be updated due to a new sensor reading. Properties may also become irrelevant at some point in time. I. e., previously unknown or irrelevant properties may become known or known properties may be removed.

Challenge 5. The inference mechanism must enable the application of modifications to the entities' properties.

4 Lifted Probabilistic Inference

Current inference approaches are not aware of observation-equivalence and furthermore use grounded states that maintain information about all particular entities separately, e. g. a state maintains binary values for all combination of locations and persons in the office scenario (Section 1). Considering the office scenario, it shows that the persons have the potential to produce the exact same sequence of observations which we propose to exploit so that actions like *get-coffee* must not be computed for every particular person but for one representative only. In order to exploit these aspects, we propose to abstract from particular entities encoded in the grounded states to represent groups of entities by what we call a *lifted state*.

Without a detailed analysis (due to the space restrictions) we want to give a notion of how such a lifted state may be represented. With respect to the office scenario a (slightly simplified) distibution of lifted states that enables inference wrt. observation equivalence can be formalised as follows:

$$\begin{aligned}
0.75 \times [\![& 3\langle \mathsf{type}{:}\mathit{person}, \mathsf{location}{:}\mathcal{L}, \mathsf{holds}{:}\mathit{Nothing}\rangle, \\
& 1\langle \mathsf{type}{:}\mathit{cm}, \mathsf{location}{:}A, \mathsf{hasCoffee}{:}\mathit{False}, \mathsf{hasWater}{:}\mathit{False}\rangle, \\
& 1\langle \mathsf{type}{:}\mathit{printer}, \mathsf{location}{:}B, \mathsf{hasPaper}{:}\mathit{False}\rangle]\!] \\
& \{\mathcal{L} \mapsto \mathit{Urn}(A, B, D)\} \\
0.25 \times [\![& 2\langle \mathsf{type}{:}\mathit{person}, \mathsf{location}{:}\mathcal{L}, \mathsf{holds}{:}\mathit{Nothing}\rangle, \\
& 1\langle \mathsf{type}{:}\mathit{person}, \mathsf{location}{:}\mathcal{L}, \mathsf{holds}{:}\mathit{Coffee}\rangle, \\
& 1\langle \mathsf{type}{:}\mathit{cm}, \mathsf{location}{:}A, \mathsf{hasCoffee}{:}\mathit{False}, \mathsf{hasWater}{:}\mathit{False}\rangle, \\
& 1\langle \mathsf{type}{:}\mathit{printer}, \mathsf{location}{:}B, \mathsf{hasPaper}{:}\mathit{False}\rangle]\!] \\
& \{\mathcal{L} \mapsto \mathit{Urn}(A, B, D)\}
\end{aligned}$$

The distribution contains two different hypotheses with probability 0.75 and 0.25, respectively. The first describes a situation where the 3 persons are at the locations A, B, and D and nobody holds something in their hands. The coffee machine is located at A with neither coffee nor water inserted. The printer is located at B with no paper inserted. The first lifted state, thus, represents all grounded states with the three persons being at different locations. I. e. it represents 6 different grounded states. The second lifted state describes the same situation except that one person holds coffee in her hands.

As can be seen these lifted states group similar entities together and reduce them to the characteristics of interest, e. g. the 3 persons whose locations are needed and whether they hold something (Challenge 1, Challenge 2 and Challenge 3). Challenge 4 and Challenge 5 are encoded within the semantics of the model's actions.

Instead of grounding the states during the inference task, our mechanism uses the lifted state representation and performs all computations directly within the lifted domain as long as possible. Possible here means that there are cases when lifted states need to be grounded to some extent, e. g. if an identifying observation is made.

5 Experiment and Results

To evaluate the ability of different inference approaches in dealing with the challenges presented in Section 3, we propose using a benchmark in the form of a dataset. The scenario for this dataset is adapted from the office scenario described in Section 1, and is aimed specifically at modeling multiple observation-equivalent entities. The main challenge of the adapted scenario is the large state space, resulting from the large number of entities and applicable actions in every step. We used the simulated dataset in order to compare our lifted probabilistic inference mechanism with a conventional mechanism that is not aware of observation-equivalencies.

5.1 Existing Datasets

Before presenting our own dataset, we describe existing human behavior datasets and simulation approaches for such datasets. We also evaluate how well these approaches are able to represent the challenges described in Section 3.

Here, we only consider datasets that consist of both observation sequences and the corresponding action sequences, so that activity recognition algorithms can be evaluated.

A number of human behavior datasets exist: Data from wearable and ubiquitous sensors, as well as simulated data. Wearable sensor data, e. g. (De la Torre et al. 2008; Krüger et al. 2015), are only recorded for one person at a time, and therefore multiple entities are not taken into account. Recordings of persons in a smart environment using binary ubiquitous sensors (e.g. presence sensors, light switch sensors, door sensors, drawer sensors) (Tapia, Intille, and Larson 2004; Singla, Cook, and Schmitter-Edgecombe 2010; van Kasteren, Englebienne, and Kröse 2011) could in principle contain observation-equivalent entities (e. g. multiple agents as well as multiple objects involved). However, the datasets that are publicly available are all designed to avoid this.

Furthermore, real-world datasets are not scalable: To test the performance of an activity recognition system, a benchmark should be able to address different problem sizes by parameterizing the domain.

In contrast, a number of simulation tools for human behavior exist. In (Synnott, Nugent, and Jeffers 2015), an overview of simulation approaches for smart environments is given. For example, in (Monekosso and Remagnino 2009), each sensor in a smart environment is described by a probability distribution and data is created by sampling from the joint distribution. However, this approach is only applicable for a coarse-grained analysis of the resulting data, as the data do not represent the underlying activities. Another system for simulating smart environment data is described in (Mendez-Vazquez, Helal, and Cook 2009). This approach selects actions based on a markov model (i.e. an action is selected based on a limited number of previous actions, but not on the complete state of the world). However, this approach can only simulate a single agent and does not include the appearance of multiple entities.

5.2 Office Scenario Dataset

None of the existing datasets models all of the challenges presented in Section 3. Therefore, we generated a new dataset by simulating action sequences from the following adaption of the office scenario which was presented in Section 1. The setup is depicted in Figure 1:

- In an office building, there are five rooms and a hallway. All rooms are connected via the hallway.
- There are n agents, 2 coffee machines and 10 coffee capsules (the number of agents can vary for the different simulation runs, see below).
- At the beginning, all agents are in room A. The location of the coffee machines and the capsules varies for the different simulation runs.
- Agents can walk from a room to the hallway or from the hallway to a room.
- When an agent is at the location of the capsules, she can pick up a capsule.
- When an agent is at the location of a coffee machine, he can insert the capsule (if he holds one), or make a coffee (if the coffee machine contains a capsule).
- Agents can hold at most one item (capsule or coffee) at a time.
- The goal is reached when all agents are holding a coffee.
- All rooms contain PIR sensors that detect if at least on agent is present. Therefore, each observation is a 6-tuple of binary variables with 1 indicating that at least one person is present and 0 indicating that there was no person recognized. The observations are accurate, i.e. no sensor errors are modeled.

This scenario is designed so that it models the challenges discussed in Section 3. The agents and the capsules are entities that are suitable for grouping (Challenge 1 and Challenge 2). Capsules are destroyed when they are picked up by the agent (Challenge 4). Furthermore, the agent's property *holding* gets updated indicating that the agent holds a capsule (Challenge 5). The other actions *moving*, *getting coffee*, and *refilling a capsule* also involve interactions as described in Challenge 4 and Challenge 5.

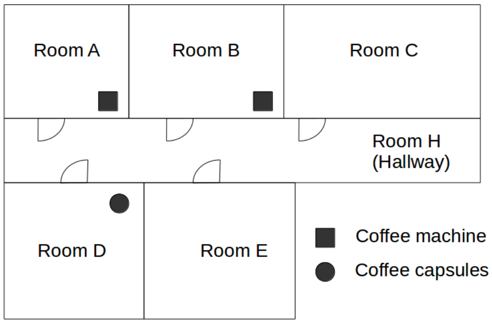

Figure 1: Environment for the adapted scenario. The scenario starts with all agents being in room A. The position of coffee machines and coffee capsules can vary for the different simulation runs.

Simulating the dataset (instead of performing a real-word experiment) allows to easily generate large amounts of data with a full coverage of parameter combinations (like the number of agents, the position of coffee machines and coffee capsules). We modeled this scenario as a planning domain and then drew samples of plans that reach a goal state.

The dataset has been generated as follows: A domain is described in a PDDL-like action language. In this domain, we defined different problems (represented by an initial state and a goal) by varying the following properties of the initial state:

- The number of agents (1 to 6)
- The position of the coffee machines (3 different positions)
- The position of the coffee capsules (2 different positions)

This results in 36 different problem definitions. For each problem, we sampled 20 plans (sequences of actions), where the probability of selecting an action depends on the goal distance of the resulting state (this ensures that all plans reach the goal state). For each plan, the corresponding observation sequence has been calculated, resulting in a total of 720 plans and observation sequences.

An example of a plan and a corresponding observation sequence is depicted in Listing 1. In the upper part of Table 1, properties of the dataset for each number of agents are listed. The number of grounded actions (actions with a specific assignment, e. g. `(goto person1 H B)`) and the number of binary state predicates (the predicates that are either true or false in each state, e. g. `(at A person1)`) are listed as an indicator of the state space size.

The dataset is published on the author's website. Furthermore, we offer a web service that allows to sample additional plans and observations for this scenario. [1]

```
(take-capsule person1)     1 0 0 0 0 0
(take-capsule person3)     1 0 0 0 0 0
(take-capsule person2)     1 0 0 0 0 0
(goto person2 A H)         1 0 0 0 0 1
(goto person2 H B)         1 1 0 0 0 0
(goto person3 A H)         1 1 0 0 0 1
(goto person3 H B)         1 1 0 0 0 0
(goto person1 A H)         0 1 0 0 0 1
(replenish person2 ca)     0 1 0 0 0 1
(goto person1 H C)         0 1 1 0 0 0
(goto person3 B H)         0 1 1 0 0 1
(goto person3 H C)         0 1 1 0 0 0
(replenish person1 cb)     0 1 1 0 0 0
(get-coffee person1 cb)    0 1 1 0 0 0
(replenish person3 cb)     0 1 1 0 0 0
(get-coffee person2 ca)    0 1 1 0 0 0
(get-coffee person3 cb)    0 1 1 0 0 0
```

Listing 1: Example of a plan (left) and the corresponding observation sequence (right) of our dataset. In this example, three agents are present, coffee machines are at rooms B and C, and all capsules are at room A.

5.3 Comparision of Grounded and Lifted Probabilistic Inference Approaches

In the following, the Lifted Probabilistic Inference algorithm sketched in Section 4 is compared with a conventional inference algorithm based on grounded states. Especially, the performance of the algorithms for different state space sizes has been investigated.

As a benchmark, we used the dataset described in Section 5. For all 720 observation sequences, activity recognition has been performed using Bayesian Filtering.

Here, we have been particularly interested in the number of states considered during Bayesian Filtering as an initial measure of performance (the algorithms become infeasible if a very large number of states has to be considered). Figure 2 shows the number of states taken into account during Bayesian Filtering using the grounded as well as the lifted state representation for the problem with both the two coffee machines as well as the capsules being located in room A with up to 4 agents. A random observation sequence from the dataset was used for the inference for each number of agents. The result is shown twice: with a linear scale on the left part of the figure and with a logarithmic scale on the right. As can be easily seen from the trajectories in both plots, the grounded approach always requires much more states than the lifted one.

For this scenario, the number of agents is the factor determining the size of the state space. Therefore, we calculated the mean maximal number of states visited during the Bayesian filtering for every number of agents. The result is depicted in Figure 3 (linear scale on the left, logarithmic scale on the right). Furthermore, the number of reachable

[1] We will publish the dataset and the web service when the paper gets accepted. We are happy to provide them for the reviewers if needed.

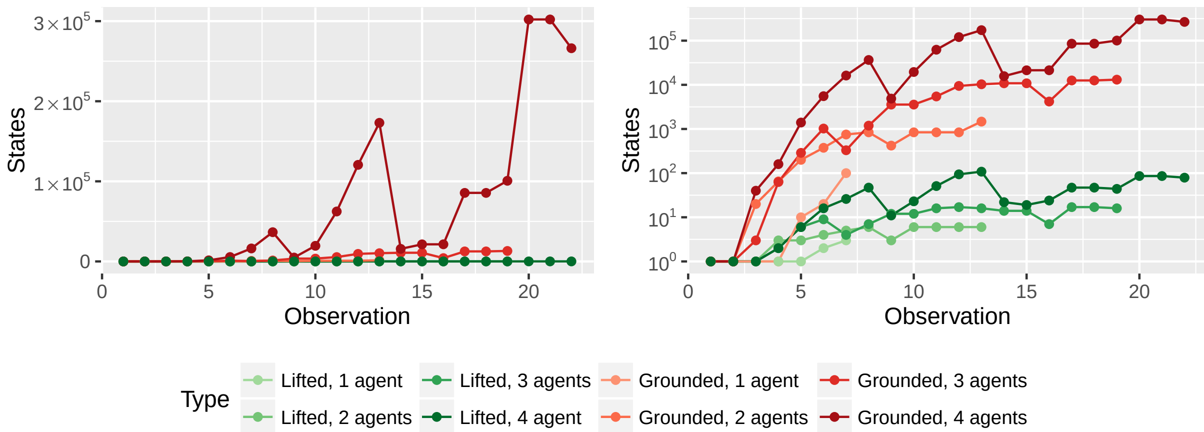

Figure 2: Number of states during Bayesian Filtering for one example of our scenario with up to 4 agents (both coffee machines at room A, capsules at room A). Using a lifted representation of states, the number of states is several orders of magnitude smaller. The same data has been plotted on a linear scale (left) and a logarithmic scale (right).

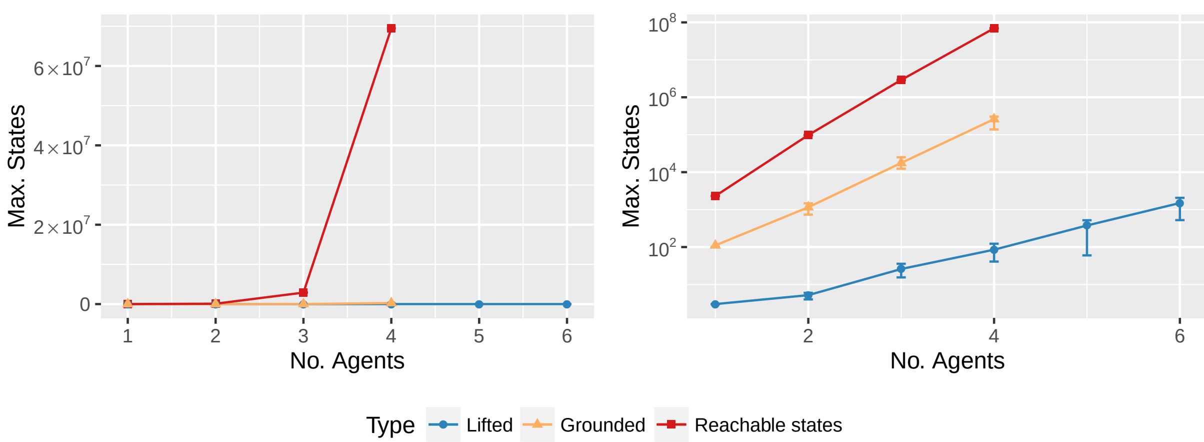

Figure 3: For each number of agents, the maximum number of states has been obtained for all plans. In this graph, the mean maximum number of states for all agents is depicted. Additionally, the size of the reachable state space is shown. The error bars show the interquartile range. The same data has been plotted on a linear scale (left) and a logarithmic scale (right).

	1 agent	2 agents	3 agents	4 agents	5 agents	6 agents
Number of grounded actions	15	30	45	60	75	90
Number of state predicates	21	30	39	48	57	66
Number of reachable states	2316	97,776	2,896,560	69,496,700	*	*
Mean plan length	7.85	16.27	26.89	38.4	50.28	61.85
Mean maximum number of grounded states during inference	111±33.8	1156±359	17625±1024	261134±20267	*	*
Mean maximum number of lifted states during inference	3±0	5.2±1	26±1.5	85.2±59.0	376±404	1502±1193

Table 1: The upper part states the properties of the dataset for each number of agents. The mean is calculated from all parameter combinations with fixed number of agents, i. e. over all 720 plans. The lower part compares the maximum number of states that has to be considered when activity recognition with this data is performed using a conventional Bayesian Filtering mechanism and the Lifted Probabilistic Inference mechanism. (* indicates that the computational effort was too high to be calculated)

grounded states for each number of agents is depicted. Note that this number is much smaller than the (theoretical) number of states, as many states are not reachable using the predefined actions (e. g. a state where a person is at two rooms at once).

The size of the reachable state space grows exponentially with the number of agents, because the number of possibilities n agents can be distributed over 6 rooms grows exponentially in n. This results for our scenario in 69,000,000 reachable states for problems with 4 agents. Because of the exponential growth, the reachable state space size could not be calculated for problems with 5 and 6 agents. The maximum number of states visited during Bayesian Inference also grows exponentially for both inference algorithms. However, for the lifted state representation, the number of states is several orders of magnitude smaller than for the grounded state representation. For example, for problems with 4 agents, the mean maximum number of lifted states is 85, while the mean maximum number of grounded states is 260,000. The maximum number of lifted states for six agents (1,500) is in the same order of magnitude than the number of grounded states for two agents (1,150). Therefore, using a lifted state representation, inference becomes feasible for scenarios with a much larger grounded state spaces. In fact, for the grounded state representation, Bayesian Filtering has been infeasible for problems with 5 or 6 agents due to the large number of states.

6 Conclusion

The field of human behavior recognition is diverse, and many different approaches tailored for specific application scenario exist. However, these mechanisms can only be applied to (very) limited scenarios. One reason is that the participating entities are considered separately during the inference task resulting in high computation costs, even if the observation data do not provide enough information to distinguish between them. We proposed an approach that exploits observation-equivalencies during the inference to reduce computational effort. We call this approach lifted probabilistic inference and presented a notion for lifted states. Furthermore, we provided a benchmark scenario including the corresponding dataset that can be used to evaluate different approaches according to the appearance of observation-equivalent entities. This benchmark is used to investigate the number of states our approach has to consider, compared to those of a conventional approach. The results showed that exploiting observation-equivalent entities has the potential for much more efficient computation.

In the future, we plan to extend our lifted probabilistic inference mechanism so that it can be used in diverse scenarios. Furthermore, the actual efficiency of the mechanism has to be extensively evaluated as well as compared to other approaches. Furthermore, we plan to evaluate the appearance of observation-equivalent entities in real world scenarios, as well as the efficiency of lifted probabilistic inference for real world scenarios.

References

Baker, C. L.; Saxe, R.; and Tenenbaum, J. B. 2009. Action understanding as inverse planning. *Cognition* 113(3):329–349.

Bui, H. H. 2003. A General Model for Online Probabilistic Plan Recognition. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 1309–1315.

De la Torre, F.; Hodgins, J.; Bargteil, A.; Martin, X.; Macey, J.; Collado, A.; and Beltran, P. 2008. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. *Robotics Institute* 135.

Fikes, R. E., and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. In *Proc. of the second Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 608–620. San Francisco: Morgan Kaufmann.

Hiatt, L. M.; Harrison, A. M.; and Trafton, J. G. 2011. Accommodating human variability in human-robot teams through theory of mind. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, 2066.

Krüger, F.; Steiniger, A.; Bader, S.; and Kirste, T. 2012. 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 2012, 1066–1074. Pittsburgh, PA, USA: ACM.

Krüger, F.; Nyolt, M.; Yordanova, K.; Hein, A.; and Kirste, T. 2014. Computational State Space Models for Activity and Intention Recognition. A Feasibility Study. *PLOS ONE* 9(11):e109381.

Krüger, F.; Hein, A.; Yordanova, K.; and Kirste, T. 2015. Recognising the actions during cooking task (cooking task dataset).

Krüger, F. 2016. *Activity, Context, and Plan Recognition with Computational Causal Behaviour Models*. Ph.D. Dissertation, University of Rostock.

Mcdermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL - the planning domain definition language. Technical Report TR-98-003, Yale Center for Computational Vision and Control,.

Mendez-Vazquez, A.; Helal, A.; and Cook, D. 2009. Simulating events to generate synthetic data for pervasive spaces. In *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*.

Monekosso, D., and Remagnino, P. 2009. Synthetic training data generation for activity monitoring and behavior analysis. In *European Conference on Ambient Intelligence*, 267–275. Springer.

Nguyen, N. T.; Bui, H. H.; Venkatsh, S.; and West, G. 2003. Recognising and Monitoring High-Level Behaviours in Complex Spatial Environments. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, II–620–5. IEEE Comput. Soc.

Nyolt, M.; Krüger, F.; Yordanova, K.; Hein, A.; and Kirste, T. 2015. Marginal filtering in large state spaces. *International Journal of Approximate Reasoning* 61:16–32.

Ramírez, M., and Geffner, H. 2011. Goal Recognition over POMDPs: Inferring the Intention of a POMDP Agent. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2009–2014.

Singla, G.; Cook, D. J.; and Schmitter-Edgecombe, M. 2010. Recognizing independent and joint activities among multiple residents in smart environments. *Journal of ambient intelligence and humanized computing* 1(1):57–63.

Synnott, J.; Nugent, C.; and Jeffers, P. 2015. Simulation of smart home activity datasets. *Sensors* 15(6):14162–14179.

Tapia, E. M.; Intille, S. S.; and Larson, K. 2004. Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing*, 158–175. Springer.

van Kasteren, T. L.; Englebienne, G.; and Kröse, B. J. 2011. Human activity recognition from wireless sensor network data: Benchmark and software. In *Activity recognition in pervasive intelligent environments*. Springer. 165–186.

Yordanova, K. 2014. *Methods for Engineering Symbolic Human Behaviour Models for Activity Recognition*. Ph.D. Dissertation, University of Rostock.