Where are my colleagues? Tracking and Counting Multiple Persons using Lifted Marginal Filtering.

Stefan Lüdtke, Max Schröder, Frank Krüger and Thomas Kirste
Mobile Multimedia Information Systems Group
Institute of Computer Science, University of Rostock
18051 Rostock, Germany
{stefan.luedtke2, max.schroeder, frank.krueger2, thomas.kirste}@uni-rostock.de

ABSTRACT

Tracking multiple targets with anonymous sensors (e.g. presence sensors) leads to a combinatorial explosion in the number of possible siuations (hypotheses) that need to be tracked, due to the uncertainty of the association of identities to observed tracks. We propose a novel Bayesian filtering algorithm that can solve this problem by employing a compact state representation. A single *lifted* state represents a uniform distribution over all possible identity-track associations. The state representation and dynamics is based on Multiset Rewriting Systems and Lifted Probabilistic Inference. We show that Bayesian filtering using this representation is possible without resorting to ground states. This is demonstrated for a person tracking scenario in an office environment where up to seven persons are observed with presence sensors. Our approach naturally allows to simultaneously track persons and estimate their total number. The number of hypotheses is several orders of magnitude smaller than using a ground state representation.

ACM Classification Keywords

I.2.3 Deduction and Theorem Proving: Uncertainty, "fuzzy," and probabilistic reasoning

Author Keywords

data association; multiple object tracking; Bayesian filtering; abstraction; Rao-Blackwellization; lifted probabilistic inference

INTRODUCTION

Tracking persons in indoor environments is necessary for applications such as providing assistance or disaster management. Bayesian filtering algorithms typically used for tracking suffer from the *state space explosion problem*: If the sensor measurements do not provide any information of the identity of the agent that produced the measurement, filtering algorithms have to keep track of all possible assignments of sensor measurements to agent identities [4]. In this work, we present a solution to this problem, based on a *lifted* filtering algorithm. The algorithm represents all possible permutations of agents and tracks as a single, parametric state.

iWOAR '17, September 21–22, 2017, Rostock, Germany

ACM ISBN 978-1-4503-5223-9/17/09...$15.00
https://doi.org/10.1145/3134230.3134237

Figure 1. Floor plan of the office environment. Red dots denote PIR sensors in the corridor, blue dots denote PIR sensors in the other rooms. Adopted from [8].

The scenario we are considering is an office environment that is equipped with passive infrared (PIR) presence sensors (cf. Figure 1). Several persons move independently in this environment. The sensors do not supply any information about the identities of an observed person, and do not allow any conclusions about the number of persons in the sensing area. Due to these sensors, we cannot distinguish the following states (cf. Figure 2, left):

- Alice is at room A, and Bob is at room B
- Bob is at room A, and Alice is at room B

We call an explicit representation of a state that is possible at a specific point in time *hypothesis*[1]. In general, for n observed persons, there are $n!$ associations of identities to observations. This combinatorial explosion makes conventional Bayesian filtering, where each of the associations is represented by a separate hypothesis, infeasible. The specific tracking scenario considered here has already been investigated by Krüger et al. Their algorithm needs to represent each idendity-observation association explicitly, and thus suffers from the combinatorial explosion[2] [8, 9].

[1]For example, each particle of a particle filter is a hypothesis.

[2]Although they avoid the combinatorial explosion in some cases by distinghishing between the different agents based on prior knowledge about the agents' goals.

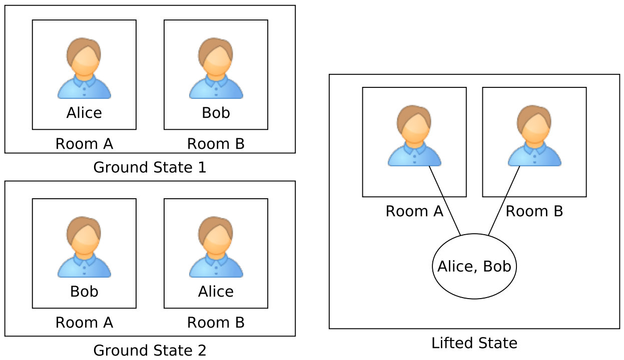

Figure 2. General idea of our algorithm. Instead of representing the identity permutations explicitly as multiple hypotheses (left), we represent them parametrically as a single hypothesis (right). The circle in the lifted state represents a uniform distribution of permutations of the names.

We propose to solve this problem by using a hypothesis representation that abstracts from the identities, i.e. a single hypothesis represents all possible permutations of tracks and identities. Multiple hypotheses thus are only neccessary to represent uncertainty about the *number* of persons per room. States are modelled as multisets of entities (key-value maps), where the values are pointers to distribution representations (in this case, a uniform distribution of identity-track associations). The general concept is depicted in Figure 2. The dynamics of the system is described by precondition-effect actions. Thus, the algorithm can be understood as a Bayesian filtering algorithm for a Multiset Rewriting System [1] with structured entities. Note that the algorithm still maintains the identity information and does not discard them completely. Therefore, as opposed to other approaches that abstract from the identities (e.g. [3]), the algorithm is also able to process *identifying* observations (e.g. a certain room can only be entered with an ID card). We show how the tracking problem can be solved efficiently with this algorithm. We also investigate the effect of different sensor placements (see Figure 1) on the tracking accuracy. Furthermore, we show that using this algorithm, we can also directly estimate the *number* of persons that are present in the environment (e.g. for disaster management applications).

METHODS

In this section, we give a brief introduction into Bayesian filtering, the basic concept that the Lifted Marginal Filter is built on. Afterwards, we describe the state representation and filtering algorithm of the Lifted Marginal Filter.

Bayesian Filtering

The goal of Bayesian filtering is to estimate the (hidden) state sequence $x_{1:t}$, based on a sequence of noisy observations $y_{1:t}$. That is, we want to estimate the distribution $p(x_{1:t}|y_{1:t})$, called *belief state*. This estimation can be peformed recursively, by decomposing it into a predict

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

and an update step

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

We call $p(x_{t+1}|x_t)$ transition model, and $p(y_{t+1}|x_{t+1})$ observation model. Common approaches to perform this recursive estimation are Hidden Markov Models (where states are discrete and the transition model is a matrix of transition probabilities), and Particle Filters, that represents the belief state by samples.

These methods represent the belief state explicitly, by enumerating all hypotheses and their probability. Thus, they suffer from the combinatorial explosion occuring in tracking scenarios with uncertainty about the agent-observation association.

Lifted Marginal Filtering

In the following, we will briefly describe the Lifted Marginal Filtering state representation and filtering algorithm. Due to lack of space, we omit the formal details here. Instead, we give an intuition on the lifted state representation, how it is used to compactly represent a belief state and how Bayesian filtering can be performed with this representation.

State Representation

The general idea of Lifted Marginal Filtering is to maintain a compact belief state representation, i.e. each *lifted* state represents a set of ground states. More concretely, a lifted state is represented by a multiset of structured entities. Entities are property-value maps. Thus, a concrete assignment of values to all properties corresponds to a ground state. For example, the multiset

$$[\![\, 1\langle \text{Name: } Alice, \text{Location: } L1\rangle, 1\langle \text{Name: } Bob, \text{Location: } L2\rangle \,]\!]$$

represents the ground state with two entities: Alice is at location L1 and Bob is at Location L2.

The *lifting* is performed by using distributions to represent slot values. The distribution used in this scenario is the urn without replacement, denoted by $\mathscr{U}(\cdot)$, which represents a uniform distribution over permutations[3]. Thus, a first approach for modelling a lifted state for the above scenario might look like this:

$$[\![\, 1\langle \text{Name: } \mathscr{U}(\text{Alice, Bob}), \text{Location: } L1\rangle, \; 1\langle \text{Name: } \mathscr{U}(\text{Alice, Bob}), \text{Location: } L2\rangle \,]\!]$$

represents four ground states: “Alice is at L1 and L2”, “Alice is at L1 and Bob is at L2”, “Bob is at L1 and L2”, and “Bob is at L1 and Alice is at L2”. Note that a lifted state can represent infinitely many ground states when we use continous distributions. By using such a representation, we assume that the distribution of slot values is independent of other slot values of this entity. The concept we use here is related to the Rao-Blackwellized Particle Filter [2]: By representing some state variables parameterically, we need fewer explicit samples to represent the whole belief state.

[3]Other distributions that are straightforward to handle are gaussian distributions, and urns with replacement. Using these distributions allows to perform lifting in other scenarios.

In the example from the beginning (cf. Figure 2), we actually want to represent only the two states "Alice is at L1 and Bob is at L2", and "Bob is at L1 and Alice is at L2"[4]. This is achieved by storing the distribution representation outside of the multiset. Instead of writing the distribution representation directly into the slot values, we use *pointers* that point to a density representation that is stored outside the multiset state. We call the structure where the density representations are stored *context*. This way, multiple slots can draw values from the *same* distribution. This gives us the possibility to represent slot values of multiple entities, that depend on each other. For our example, this allows us to model the constraint that each entity has a unique name. The example finally looks like this[5]:

$$[\![\, 1\langle \text{Name: } N, \text{Location: } L1\rangle, 1\langle \text{Name: } N, \text{Location: } L2\rangle \,]\!] \\ \{ N \mapsto \mathscr{U}(\text{Alice, Bob}) \}$$

Using this representation, we separate the *structure* of entities from the distribution of possible values. This way, we can group multiple entities that have the same structure, but can have distinct slot values in each ground state. For example, the state

$$[\![\, 3\langle \text{Name: } N, \text{Location: } L1\rangle, 1\langle \text{Name: } N, \text{Location: } L2\rangle \,]\!] \\ \{ N \mapsto \mathscr{U}(\text{A, B, C, D}) \} \quad (3)$$

represents all states where three entities are at L1 and one entity is at L2, but their identities are arbitrary permutations.

The belief state (a distribution over ground states) can be represented by a distribution over lifted states, because each lifted state itself represents a distribution over ground states. This means the belief state can be represented very compactly, compared to an explicit representation of ground states.

State Dynamics

By representing the belief state as a distribution over lifted states, the question is whether we can define a Bayesian filtering algorithm for this representation. It is crucial for the efficiency of our approach that the filtering algorithm can operate on the lifted state representation (grounding all states would result in the combinatorical explosion that we are trying to avoid with our approach). We can indeed define such an algorithm. We will describe the predict and update steps of the algorihm in the following.

The transition model is defined by precondition-effect actions. A precondition can be understood as a condition on an entity, e.g. the presence of a certain slot, or a specific slot value. The effects can manipulate the state arbitrarily: Create and remove entities, manipulate slots, and manipulate the context. There are cases where the preconditions are indeterminate with respect to an entity, because a slot value of an entity is not grounded, but a distribution that can either satisfy or not satisfy the precondition, depending on the value that is drawn from the distribution. In this case, we perform an operation called *splitting*. That is, we split the lifted state in two lifted states, one where the precondition is satisfied, and one where it is not satisfied. This requires manipulating the corresponding distribution representations. Whether this operation is possible depends on the concrete distribution, but for the distribution used here (urns without replacements), this is rather straightforward. For example, splitting the state from Equation 3 on the precondition "Name=A & Location=L2"[6] results in two states. The state where the precondition is satisfied:

$$[\![\, 3\langle \text{Name: } N, \text{Location: } L1\rangle, 1\langle \text{Name: } N', \text{Location: } L2\rangle \,]\!] \\ \{ N \mapsto \mathscr{U}(\text{B, C, D}), N' \mapsto \delta(A)^{7} \}$$

and the state where the precondition is not satisfied:

$$[\![\, 1\langle \text{Name: } N', \text{Location: } L1\rangle, 2\langle \text{Name: } N, \text{Location: } L1\rangle, \\ 1\langle \text{Name: } N, \text{Location: } L2\rangle \,]\!] \\ \{ N \mapsto \mathscr{U}(\text{B, C, D}), N' \mapsto \delta(A) \}$$

Note that the first state has a probability of $1/4$ and the second state has a probability of $3/4$ because the probability of person A being at location L2 is $1/4$, because this is exactly the situation described by the state in Equation 3 (due to the uniform distribution of the urn). Thus, we can decide on the applicability of an action in a state, without completely grounding the state. The transition model is obtained by computing maximally parallel compound actions (MPCAs), similar to the Multiset Rewriting System described by Barbuti et al. [1]. That is, entities that satisfy a precondition are *bound* to this action, and cannot be used to satisfy the preconditions of another action. A MPCA is a set of actions such that no further action is applicable. Note that there can be multiple MPCAs. The probability of each MPCA (i.e. each state transition) is based on the weights of the (atomic) actions and the number of ways the entities can be bound to the preconditions of the MPCAs (i.e. the multiplicity of this MPCA). Finally, performing the predict step means applying all possible MPCAs to each state. The posterior probability of each state is the product of the prior and the probability of the MPCA. Afterwards, we merge equal states by summing up their probability, similar to the marginal filter [10].

The update step can be any function that manipulates a state distribution. In our case, we discard all states that are not compatible with the observations: That is, all states where at least one entity is at a room where no entity was observed, are discarded. The distribution of the remaining lifted states is normalized afterwards.

EXPERIMENTAL EVALUATION

The overall goal of the experiments was to demonstrate the capabilities of Lifted Marginal Filtering in the specific scenario. More specifically, we will anwer two questions: (1)

[4] The other two states "Alice is at L1 and L2" and "Bob is at L1 and L2" are impossible, because a person can only be at one location at a time.

[5] N is the *density label* that points to the distribution representation $\mathscr{U}(Alice, Bob)$ stored in the context.

[6] We do not attempt to introduce a formal syntax for preconditions here. The precondition "Name=A & Location=L2" should be read as: "Requires an entity that has a slot *Name* with value *A* and slot *Location* with value *L2*".

[7] $\delta(\cdot)$ denotes the Kronecker delta.

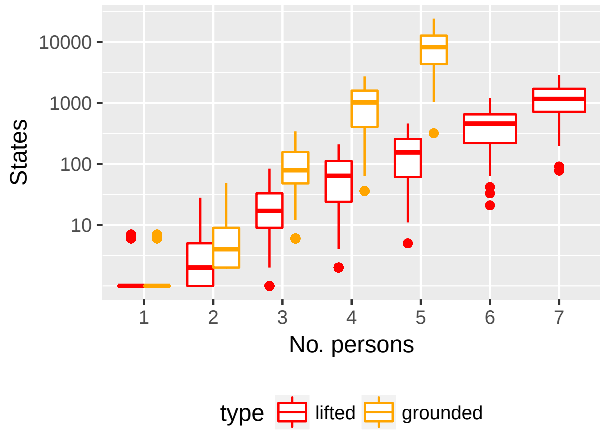

Figure 3. Mean number of hypotheses stored explicitly during filtering for different number of people. Filtering the ground state representation has been infeasible for six and seven agents.

Can the algorithm reduce the number of hypotheses necessary for Bayesian filtering in this scenario, and (2) can the algorithm solve the simultaneous tracking and counting problem.

Empirical Data

We will evaluate our algorithm with an office scenario dataset [6]. The scenario consists of persons that move around an office. The task is to track the persons, i.e. estimate the position of each person over time. The floor plan is depicted in Figure 1. Experiments with one to seven people that move around these rooms have been executed. For each number of persons, five iterations have been executed, resulting in 35 iterations. The data has been manually annotated in one-second time steps.

Based on these annotations, we simulated observation data of presence sensors that only recognize if a person was present or not, but not which person or the number of persons that are present. Apart from this uncertainty, this simulated data is always correct, i.e. if a person was in a room at a specific time step, this is always correctly recognized by the simulated sensors.

Model

We model this scenario in our approach as follows. Each state contains one entity for each person. Each entity has a *location* and a *name* slot. The location is represented explicitly, and the name is represented by an urn without replacement.

The initial location distribution has been estimated empirically based on the recorded data. Depending on the performed experiment (see below), the initial belief state assigns a non-zero probability only to the correct number of agents, or it is a distribution over the number of agents[8].

The transition model is described by two types of actions, $move(from, to)$ and *stay*. The probability of each action has been estimated empirically from the recorded data. Due to the transition semantics of our approach (transitions are maximally parallel compound actions), all agents must perform a move or a stay action simultaneously. For example, for two agents that are both at the same room, and an environment consisting of two rooms, there are 3 possible MPCAs: both stay, both move, and one of them moves. Note that we do not have to distinguish which of the agents moved, due to the parametric description of the agents' names.

Experimental Design

Here, we describe the experiments we performed in order to analyze the capabilities of the filtering algorithm. To compare results, we use the root mean squared error (RMSE).

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T} \sum_{l=1}^{|Locations|} (truth_{l,t} - estimate_{l,t})^2}{T * |Locations|}} \quad (4)$$

$truth_{l,t}$ and $estimate_{l,t}$ denote the true and estimated number of persons at room l at time step t. That is, we only compare if the correct *number* of persons have been estimated, but we do not consider the persons' *identities* (because the observations do provide any information about the identities).

The following research questions have been adressed:

Q1 Which effect has the lifted state representation on the number of hypotheses stored explicitly, compared to a ground state representation?

Q2 What effects do the sensor placement and prior knowledge of the number of persons have on the error in the estimated number of persons per room?

The rationale behind Q2 is as follows: In a real-world scenarios, presence sensors might only be installed in some rooms due to cost and privacy issues. We therefore investigate the performance of our approach with different sensor placements. Furthermore, the number of persons might not be known a priori, e.g. because persons might enter and leave the observed area.

For answering Q1, we assessed the filtering algorithm for each of the 35 data sets using a lifted as well as the ground state representation, and compared the mean number of hypotheses for a fixed number of persons. For this experiment, the true number of persons is known a priori, and only the hallway sensors are used (as these factors are not relevant for assessing Q1).

To answer Q2, we varied the prior knowledge about the number of persons, and the used sensor data. Two types of sensor placement have been evaluated: (1) PIR sensor only in the corridor (red dots in Figure 1), (2) PIR sensors in all rooms (blue and red dots in Figure 1). Three levels of prior knowledge about the number of agents have been evaluated: (1) number of agents known a priori, (2) number of agents known up to ± 1 of the true number, (3) number of agents unknown, i.e. one to seven agents possible. These six parameter combinations have been assessed on all 35 data sets. We evaluated the RMSE (cf. Equation 4), and the error in the estimated number of persons.

[8] We use Jeffreys' prior for scale parameters [5], which is $p(n) \propto 1/n$.

RESULTS

Figure 3 shows the number of hypotheses of the Lifted Marginal Filter and a conventional Bayesian filter with ground state representation. Both approaches need an exponential number of states to represent the belief state. This is due to the exponential number of possible multiplicities of persons per room, i.e. both approaches need to represent the situations: "two agents are at location A, one agent is at location B", and "one agent is at location A, two agents are at location A" explicitly. However, the belief state representation is several orders of magnitude smaller when the lifted state representation is used, as all identity permutations are represented by a single lifted state. The effect gets more pronounced for data sets with a larger number of agents. For 2 (3, 4, 5) agents, the grounded representation needs 1.8 (4.8, 14.4, 53.3) times the number of hypotheses than the lifted representation. The reason for this behaviour is the factorial blowup the ground belief state representation experiences due to the factorial number of identity-entity associations. This result shows that Lifted Marginal Filtering can indeed maintain a compact belief state representation in this scenario, compared to an algorithm that employs a ground hypotheses representation (Q1). Furthermore, the advantage of the lifted representation gets even more pronounced for larger scenarios.

For answering Q2, we calculated the RMSE (cf. Equation 4) for each sensor configuration, and each prior knowledge level about the number of agents. The results are shown in Figure 5. An interesting observation is the fact that the sensor configuration has the largest effect on the RMSE. Observing all rooms (instead of just the corridor) naturally leads to a lower RMSE, because more information on room occupation are available. However, even when only the corridor is observed, we still achieve an RMSE of < 0.5. The second interesting fact is that prior knowledge about the number of agents does not increase the accuracy substantially. Thus, person tracking in this scenario is possible even without knowing the number of persons in advance.

Figure 4 shows the absolute error in the estimated total number of persons. In the left plot, only the corridor has been observed, and in the right plot, all rooms have been observed. In all cases, the estimate becomes more accurate over time. This is due to the fact that over time, sensor observations occur that can only be explained by a specific number of persons. These observations rule out hypotheses with the wrong number of agents. When all rooms are observed, the error always becomes zero after some time, i.e. only hypotheses with the correct number of agents are present. Thus, the evaluation shows that simultaneously tracking persons and estimating their number using Lifted Marginal Filtering is indeed possible for this scenario.

RELATED WORK

Multiple object tracking with anonymous sensor is a well-known problem in the literature [11, 3]. They either represent all possible data associations explicitly and thus suffer from the combinatorial explosion [11], or they discard the identity information completely [3].

Huang et al. [4] employ propose to represent a distribution over permutations (each permutation represents a data association) by its first few fourier coefficients. This way, they can compactly represent an exponential number of hypotheses. Our work is different in two aspects. Their representation is only compact because higher-order fourier coefficients are discarded, i.e. by approximation. We propose an *exact* algorithm. Furthermore, their algorithm is only concerned with the combinatorial explosion resulting from permutation effects. While we use our algorithm only for permutations here, it can in principle handle other forms of symmetries, depending on the distributions that are maintained in the context.

The state representation and dynamics of Lifted Marginal Filtering is similar to the maximally parallel semantics for Multiset Rewriting Systems proposed by Barbuti et al. [1]. However, their system does not allow structured entities, and they do not devise a filtering algorithm.

Another concept that is related to our algorithm is Lifted Probabilistic Inference [7]. These approaches are concerned with symmetries in graphical models. They aim at exploiting these symmetries by grouping similar random variables, and performing inference over these groups (just as our algorithm groups similar states). However, they do not explicitly support sequential inference in dynamic domains, i.e. Bayesian filtering consisting of a predict-update cycle.

CONCLUSION

We introduced the problem of tracking and counting multiple persons based on anonymous sensors in partially observed environments. We showed that the combinatorial explosion in the number of hypotheses that is inherent to this problem can be handled by a novel filtering algorithm that compactly represents sets of hypotheses. Using this algorithm, the belief state can be represented more compactly, enabling more efficient filtering. Furthermore, we showed that estimating number of agents can directly be integrated into the filtering algorithm, without additional effort. Situations with up to 7 agents could be handled exactly by our algorithm. Thus, for typical households we are able to give a good estimation about the number and position of persons, based solely on presence sensors. Although not explicitly covered here, the algorithm is a very general solution to the problem. For example, it is straightforward to process *identifying* observations (via splitting operations), which is not possible with representations that do not model the identity information at all.

To employ the algorithm to larger problem domains, approximation methods have to be investigated. A simple form of approximation is *pruning*, i.e. discarding unlikely hypotheses. Another aspect that has not been investigated here is *merging*, the opposite operation to splitting. Merging multiple states results in a single lifted state that represents them. Merging can be performed exactly or approximately (by assuming independence between slot values that is not supported by the actual ground states). Investigating the effect of these methods regarding accuracy and belief state size is a topic for future research. Another interesting aspect is to also represent the *entitiy multiplicities* in a parametric way, such that the two

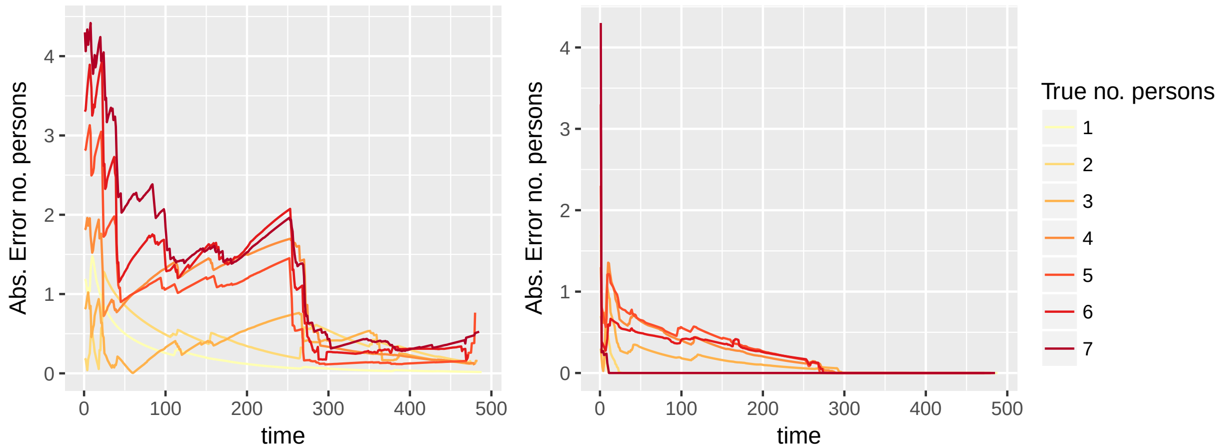

Figure 4. Error of estimated number of persons for corridor sensors only (left) and all sensors (right), without prior knowledge about the true number of persons. For both sensor configurations, the estimate becomes more accurate over time. Aggregated over all five iterations.

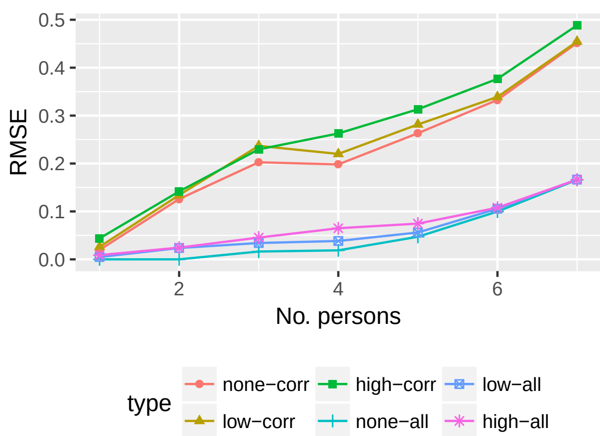

Figure 5. RMSE of estimated position (cf. Equation 4) for different sensor configurations, and different levels of uncertainty about the number of agents. corr: sensors only in corridor, rooms: sensors in all rooms, none: Number of agents is known a priori, low: initial hypotheses of number of agents in range $\pm$ 1 of correct number; high: initial hypotheses of number of agents in range 1 to 7. Aggregated over all five iterations.

states "two agents at location A, one agent at B" and "one agent at A, two agents at B" can be represented by a single parametric state. This would result in a belief state that grows only polynomially with the number of agents.

REFERENCES

1. R. Barbuti, F. Levi, P. Milazzo, and G. Scatena. 2011. Maximally Parallel Probabilistic Semantics for Multiset Rewriting. *Fundamenta Informaticae* 112, 1 (2011), 1–17. DOI: http://dx.doi.org/10.3233/FI-2011-575
2. A. Doucet, N. De Freitas, K. Murphy, and S. Russell. 2000. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 176–183.
3. U. Hanebeck and M. Baum. 2015. Association-Free Direct Filtering of Multi-Target Random Finite Sets with Set Distance Measures. In *18th International Conference on Information Fusion*. IEEE, 1367–1374.
4. J. Huang, C. Guestrin, and L. Guibas. 2009. Fourier Theoretic Probabilistic Inference over Permutations. *Journal of Machine Learning Research* 10 (June 2009), 997–1070.
5. Harold Jeffreys. 1946. An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London a: mathematical, physical and engineering sciences*, Vol. 186. The Royal Society, 453–461.
6. Martin Kasparick and Frank Krüger. 2013. Probabilistic action selection - Tracking multiple persons in indoor environments. (2013). http://dx.doi.org/10.18453/rosdok_id00000114
7. K. Kersting. 2012. Lifted Probabilistic Inference. In *Proceedings of the 20th European Conference on Artificial Intelligence*, Vol. 242. IOS Press, 33–38. DOI: http://dx.doi.org/10.3233/978-1-61499-098-7-33
8. Frank Krüger. 2016. *Activity, Context and Intention Recognition with Computational Causal Behavior Models*. Ph.D. Dissertation. Universität Rostock. http://purl.uni-rostock.de/fodb/pub/53687
9. Frank Krüger, Martin Kasparick, Thomas Mundt, and Thomas Kirste. 2014. Where Are My Colleagues and Why? Tracking Multiple Persons in Indoor Environments. In *10th International Conference on Intelligent Environments (IE), 2014*. Shanghai, China. DOI: http://dx.doi.org/10.1109/IE.2014.35
10. Martin Nyolt, Frank Krüger, Kristina Yordanova, Albert Hein, and Thomas Kirste. 2015. Marginal Filtering in Large State Spaces. *International Journal of Approximate Reasoning* 61 (June 2015), 16–32. DOI: http://dx.doi.org/10.1016/j.ijar.2015.04.003
11. D. Reid. 1979. An Algorithm for Tracking Multiple Targets. *IEEE Trans. Automat. Control* 24, 6 (1979).