Human Activity and Context Recognition using Lifted Marginal Filtering

Stefan Lüdtke, Kristina Yordanova and Thomas Kirste
Institute of Computer Science
University of Rostock, Germany
{stefan.luedtke2,kristina.yordanova,thomas.kirste}@uni-rostock.de

***Abstract*—Computational causal behavior models can be used for joint human activity recognition and reasoning about the context of an activity, like the location of used objects, which is relevant for assistive systems. Such models are computationally expensive due to the large number of different states that need to be considered. However, the distribution of these states is often highly symmetrical. Lifted Marginal Filtering (LiMa) is an inference algorithm that maintains a suitably factorized state distribution, such that symmetrical factors can be represented compactly. In this paper, we show for the first time the application of LiMa to a complex real-world activity recognition setting based on real IMU data. This is achieved by introducing an operation that prevents the distribution representation to grow indefinitely, by projecting the distribution back to an exchangeable distribution. We show that LiMa needs fewer states to represent the exact filtering distribution, and achieves a higher activity recognition accuracy when only limited resources are available to represent the state distribution.**

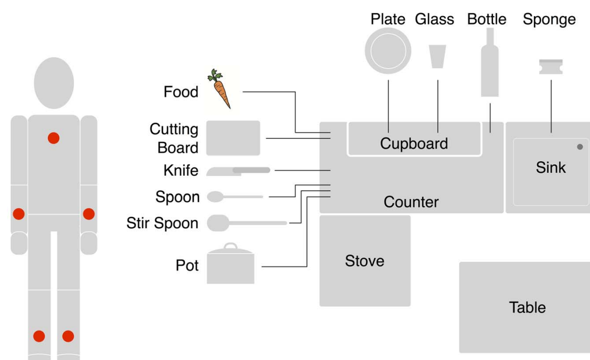

Fig. 1: Instrumentation and trial setting of kitchen task. Reprinted from [7].

I. Introduction

Recognizing activities of daily living (ADLs) and inferring context information from data of inertial measurement units (IMUs) is a challenging task that is highly relevant for providing automatic assistance. Apart from recognizing the activities, an assistive system needs to reason about the *context*, i.e. the state of the environment, like the location or other properties of objects. *Computational Causal Behavior Models* (CCBMs) [7] are probabilistic inference methods where the system dynamics (the *transition model*) are modeled by probabilistic precondition-effect actions, i.e. they incorporate causal information to infer not only performed actions, but also context information.

These models are computationally expensive, due to the large number of discrete environmental states that need to be tracked. However, the tracked states often have some symmetrical structure, due to the fact that the environment state is not observed directly: For example, observing that an object was moved does not allow to discriminate the identity of the object, and thus, multiple states need to be tracked, that are only different in the permutation of objects and locations. More formally, the state distribution exhibits (partial) *exchangeability*. Lifted Marginal Filtering (LiMa) [8] exploits this property, such that fewer states need to be represented explicitly, and thus inference becomes more efficient. The central technical idea is to introduce a suitable factorized representation of the state distribution, such that some factors are *exchangeable* distributions that can be represented more compactly than by complete enumeration (Section III). The algorithm can perform inference directly on the factorized representation without resorting to the original, much larger distribution (Section IV). When the system dynamics break the exchangeability, the state representation automatically adapts to a more explicit representation by *splitting* operations, conceptually similar to operations in lifted probabilistic inference [13].

The technical contribution of this paper is twofold: (i) a method to tackle the problem that the state representation grows indefinitely over time due to repeated splitting operations, by introducing a domain-specific approximation operation that projects the distribution back to an exchangeable distribution (Section IV-C); (ii) empirical evidence that the proposed method allows us to use LiMa for a complex real-world activity and context recognition task (Section V), as opposed to the simulated scenarios considered in prior work [15], [8].

II. PROBLEM STATEMENT

We start by outlining the problem that makes the factorized distribution representation necessary. In this paper, we are concerned with activity and context recognition in a kitchen task (see Figure 1) that consists of the subtasks (i) preparing the kitchen, (ii) cooking, (iii) preparing the table, (iv) eating, and (v) washing the dishes. The data (IMU data of 7 participants performing the task) is annotated with an *action class* (16 possible classes, e.g. taking, moving, cutting, washing), handled object (10 objects) and locations (5 locations).

The states that need to be tracked for activity and context recognition contain the location and state (clean/dirty, cooked, ...) of each object and the agent. Due to the fact that context information is not observed by the sensors directly, typically there is a very large number of states that have non-zero probability at a given time. In a previous study, the state space for this scenario consisted of 146 million states [12].

However, many of these states are similar, except for the permutation of objects: Suppose we observe that the agent is *taking* an object, then *moves*, and then placing an object down, and suppose that two objects A and B are available. Based on the observations, we cannot discriminate whether A or B was moved, and thus we need to track (at least) two states. For 10 objects, there are $10! = 3,628,800$ permutations. In the real scenario, not all permutations can actually occur. Furthermore, some actions require a specific object (e.g. only the pot can be used for cooking), thus when observing such an action, the object identity can be inferred. Still, abstracting from permutations can potentially reduce the required number of states by a large amount.

III. FACTORIZING MULTISET DISTRIBUTIONS

In this section, we will present the main technical idea of LiMa: a suitable factorization of a distribution of states, such that some factors are *exchangeable* and can be represented compactly. For more details, we refer to [8].

In our approach we use multisets to represent states. Multisets naturally allow to describe situations where multiple similar objects are present, i.e. multiset-based states allow the compact representation of similar situations that we are interested in.

Formally, a multiset $s \in \mathcal{S}$ is a function of *entities* $e \in \mathcal{E}$ to multiplicities (how often this entity occurs in the multiset), i.e. $S = \mathcal{E} \mapsto \mathbb{N}_{\geq 0}$. We use *structured* entities here, i.e. entities are key-value maps. For example, suppose that at some point in the kitchen task, three objects exist: A spoon, a plate, and a pot. The spoon and the pot are dirty, the plate is clean. This situation is represented by the following multiset[1]:

$$\begin{aligned} s = \llbracket\, & 1\langle \text{N: } Spoon, \text{D: } \top \rangle, 1\langle \text{N: } Pot, \text{D: } \top \rangle, \\ & 1\langle \text{N: } Plate, \text{D: } \bot \rangle \,\rrbracket \end{aligned} \tag{1}$$

Due to uncertainty in the sensor data, we need to maintain a *distribution* $p(S)$ of such states over time. This can simply be done by maintaining a set of tuples (s_i, p_i) to represent a categorical distribution. However, the number of tuples can easily grow very large, as outlined in Section II.

Suppose we can decompose a multiset state s into two parts r and v, such that there is a bijection from s to tuples (r, v). Then, $p(S)$ can be factorized as:

$$p(S) = p(R, V) = p(R)\, p(V|R) \tag{2}$$

This concept, called Rao-Blackwellization [5], is independent of multisets. The idea here is that v is actually a *vector* of *exchangeable* random variables (RVs) $v_1, \dots, v_n$. A set of RVs $v_1, \dots, v_n$ is *exchangeable* when for all permutations π of $\{1, \dots, n\}$, $p(V_1{=}v_1, \dots, V_n{=}v_n) = p(V_1{=}v_{\pi(1)}, \dots, V_n{=}v_{\pi(n)})$. Distributions of exchangeable RVs can be represented more efficiently than by complete enumeration, by *sufficient statistics* [9].

The question now is how to decompose a multiset state s into such a tuple (r, v). Here, the fact that we use structured entities allows for such a decomposition: We separate the *structure* of the multiset (how many entities are there, and what are their properties) from the values of the properties. For example, the state given in Equation 1 can be decomposed as follows:

$$\begin{aligned} r =& \llbracket\, 1\langle \text{N: } n_1, \text{D: } d_1 \rangle, 1\langle \text{N: } n_2, \text{D: } d_2 \rangle, \\ & 1\langle \text{N: } n_3, \text{D: } d_3 \rangle \,\rrbracket \\ v =& (Spoon, \top, Pot, \top, Plate, \bot) \end{aligned} \tag{3}$$

Given that $p(S)$ is a categorical distribution with finite support, $p(R)$ is also a finite categorical distribution, as multiple states s are mapped to a single structure r. We assume that $p(V|R)$ is a product of m exchangeable distributions such that $p(V|R) = \prod_i p_i(V_i|R)$, where V_i are the subset of RVs corresponding to p_i. We call $\rho(p_i)$ the *representation* of p_i. For example, a uniform distributions of permutations of 3 objects A_1, A_2, A_3 (i.e. an urn without replacement where 3 elements are drawn) can be represented as "$\mathcal{U}(A_1, A_2, A_3)$".

The remaining question is how to associate property values in r with the RVs $V_1, \dots, V_n$. We propose a labeling mechanism that provides this association. The following example illustrates this, for a more formal definition we refer to [8]:

[1]We use $\llbracket \cdot \rrbracket$ to denote multisets, and $\langle \cdot \rangle$ to denote partial functions (maps).

Consider the situation shown in Equation 3. However, suppose we now only know that two of the objects are dirty and one is clean, but not their identities. This means we need to maintain three states to represent $p(S)$. Representing this situation in the factorized representation, however, requires only a single tuple $(r, \rho(p(V|r)))^2$:

$$\begin{aligned} r =& [\![\, 2\langle \text{N: } \mathbf{n}, \text{D: } \mathbf{d_1} \rangle, 1\langle \text{N: } \mathbf{n}, \text{D: } \mathbf{d_2} \rangle \,]\!] \\ \rho(p(V|r)) =& \langle \mathbf{n}\text{: } \mathcal{U}(\mathit{Spoon}, \mathit{Pot}, \mathit{Plate}), \\ & \mathbf{d_1}\text{: } \delta_\top, \mathbf{d_2}\text{: } \delta_\bot \rangle \end{aligned} \tag{4}$$

Here, $\mathbf{n}$, $\mathbf{d_1}$ and $\mathbf{d_2}$ are labels (or pointers) that associate properties in r with the corresponding factor of $p(V|r)$. As all of these factors are exchangeable, this unambiguously defines the distribution of each property. For simplicity, we will abbreviate the entities with e_c and e_d (with indices indicating whether they are clean or dirty), such that we can write $r = [\![\, 2e_d, 1e_c \,]\!]$.

We call such a tuple $l = (r, \rho(p(V|r)))$ a *lifted state*. Each lifted state describes a distribution of (ground) states that all have the same structure r. In the example, the lifted state represents a uniform distribution of three ground states, where either the spoon, the pot, or the plate is clean, and the other two objects are dirty.

IV. Lifted Marginal Filtering

In the following, we show how this state representation can be used for *Bayesian filtering* [14]: BF iteratively computes the filtering distribution $p(L_{t+1}|y_{1:t+1})$ for time $t+1$ from the previous distribution $p(L_t|y_{1:t})$ at time t and an observation y_{t+1}. This calculation is typically decomposed into two steps: The *predict* step calculates the distribution after applying the *transition model* $p(L_{t+1}|L_t)$:

$$p(L_{t+1}) = \sum_{l_t} p(L_{t+1}|L_t = l_t)\, p(L_t = l_t|y_{1:t}) \tag{5}$$

Afterwards, the posterior distribution is calculated by employing the *observation model* $p(y_{t+1}|L_{t+1})$:

$$p(L_{t+1}|y_{1:t+1}) = \frac{p(y_{t+1}|L_{t+1})\, p(L_{t+1}|y_{1:t})}{p(y_{t+1}|y_{1:t})} \tag{6}$$

A. Probabilistic Multiset Rewriting Systems

BF can be performed directly on lifted multiset states, by using a *multiset rewriting system* to describe the transition model. Thus, LiMa can be seen as some variant of a CCBM (i.e. the transition model is described symbolically by precondition-effect actions).

More specifically, we use a *probabilistic multiset rewriting system* (PMRS) [2]. An *action* a is a triple (c, f, w), where c is a list of *preconditions*, f is an *effect function* and w is a *weight*. As we use structured entities, the preconditions are formulated in terms of *constraints* on the entities (i.e. as boolean functions of entities), e.g. testing for existence of a property or for a specific value of a property. An action can have more than one precondition, i.e. more than one entity can participate in an action.

As we use lifted states, preconditions can be indeterminate with respect to some state: For example, the lifted state in Equation 4 is indeterminate for the precondition "Name=Spoon", as the precondition is true only for some part of the support of the corresponding distribution. In this case, we need to perform a *splitting* operation [8] , which splits the lifted states in multiple lifted states such that the precondition is determinate for each split product.

Actions are applied to states by *binding* entities from the state to the preconditions. We call a pair of action a and a list of bound entities i an *action instance*. The effect manipulates the state based on the bound entities.

For example, consider the action *wash* that can be applied to any object and has the effect that the object is clean afterwards. This action has two valid action instances (i.e. each bound entity satisfies the precondition, and $i \subseteq r$) for the state in Equation 4: $a_1 = (wash, e_c)$ and $a_2 = (wash, e_d)$. Applying these action instances to the state l leads to the *successor states*, $apply(a_1, l) = l'_1 = [\![\, 2e_c, 1e_d \,]\!]$ and $apply(a_2, l) = l'_2 = [\![\, 3e_c \,]\!]$.

The *multiplicity* $\mu_l(a)$ of an action instance a is the number of of ways the bound entities in a can be chosen from l. The normalized product of multiplicity and weight defines the probability $p(a|l)$ that a specific action instance a is applied in l. The action distribution $p(a|l)$ defines the *transition model* as follows:

$$p(l'|l) = \sum_{\{a|apply(a,l)=l'\}} p(a|l) \tag{7}$$

B. Approximate Filtering

Even when using the lifted state representation derived in Section III, the number of states that need to be tracked can become very large. This issue is tackled by approximating $p(L_t|y_{1:t})$ by a set of weighted samples (called *particles*). The resulting algorithm is called *particle* filter [1]. Here, we use a variant of the particle filter for categorical domains called *marginal filter* (MF) [12]. The MF works by computing Equations 5 and 6 exactly, and then sampling n particles from the posterior $p(L_{t+1}|y_{1:t+1})$ without replacement. We use the resampling strategy presented in [11], which is optimal (in the sense of least squared error) and unbiased.

C. Merging

Over time, the lifted state distribution can degenerate due to symmetry breaks (splitting operations), such that

[2] δ_x is the Kronecker delta, i.e. the distribution that is 1 for x and 0 otherwise.

it does not exhibit any exchangeability, and resorts to the ground state representation. To prevent this, we introduce a *merging* operation, that projects the value distribution back to an exchangeable distribution. The merging operation we use in this paper simply projects all values of the *name* property back to a uniform distribution of permutations.

The question is when this operation should be sensibly applied during filtering: In the worst case, the next transition requires a split, such that no reduction in representation size is obtained, and information about object identity that is necessary in the transition has been discarded.

However, in the scenario we are considering, we know that after a specific action was performed, some object identities are not needed again: After cooking, it is never necessary to distinguish the pot from the other objects, after drinking we do not need to distinguish the glass, and so on. Thus, we can create a *domain-specific* merging operation that is applied whenever we can safely "forget" object identities.

V. Experimental Evaluation

The overall goal of the experiments was to show how the lifted state representation can be beneficial (as compared to the ground state representation used in the conventional marginal filter) in a *real-world setting*, as opposed to the artificial or simulated scenarios investigated previously. More specifically, two related aspects were investigated: (**Q1**) Can inference complexity be reduced by using a lifted state representation, in terms of representation size of the filtering distribution; and (**Q2**) can this lead to a higher estimation accuracy, given a fixed computational effort (in terms of allowed number of particles)?

Note that activity recognition accuracy in comparison to other (data-driven) methods is not of major concern in this study: As argued above, fine-grained causal models have advantages in itself (inference of context information, reduced need for training data), that justify that activity recognition accuracy of these models is not significantly better than of data-driven approaches.

A. *Trial Setting and Model*

The evaluation is performed on a dataset obtained from subjects performing kitchen activities, as described in Section II. This domain was chosen as it is sufficiently complex and has a non-trivial causal structure.

The dataset is available at [6] and has been used in multiple previous studies [7], [12]. Out of the 30 IMU signals sampled at 120 Hz, 180 features such as variance and energy were computed with a window size of 128 samples and 75% overlap. Afterwards, a principal component analysis was performed, and the 21 principal components with the largest eigenvalues were selected.

TABLE I: Factors and levels of experimental design.

Factor	Levels
Model	QDA, HMM, MF, LiMa
Observations	Action Classes, IMU data
Subject	$1, \ldots, 7$
Run	$1, \ldots, 10$
Particles	25, 50, 100, 200, 500, 1000, 2000, unlimited

In the resulting dataset, each action has a distinct *duration* distribution, which is not necessarily a geometric distribution, thus requiring to model the duration distribution explicitly, by concepts similar to hidden semi-Markov models [16]. This, however, adds another layer of complexity and additional parameters, which we wanted to avoid in this work. Therefore, we reduced the dataset such that each action lasts for exactly one timestep, by sampling one observation for each segment where the same action is executed.

We modeled the domain as a PMRS: Each object shown in Figure 1, as well as the agent, is modeled as an entity, and a set of PMRS actions has been created to model the causal structure of the domain. The action weights are chosen according to a goal distance heuristic (where the goal is that the meal has been finished and all objects are washed), based on the task script [7].

We investigated two different observation models $p(y_t|l_t)$: (i) *Crisp* observations of the actual *action class* c_t, i.e. $p(y_t|l_t) = \mathbf{1}[a_t{=}c_t]$ where a_t is the action executed in l_t; and (ii) we used the preprocessed real sensor data as observations, and assumed $p(y_t|l_t)$ to be a multivariate normal distribution, conditional on each action class, i.e. $p(y_t|l_t) \sim \mathcal{N}(\mu_{a_t}, \Sigma_{a_t})$.

B. *Experimental Design*

We used a factorial experimental design with the factors depicted in Table I. For each configuration, 10 filtering runs have been performed to assess the variability due to randomness in the resampling step. We compared our approach (LiMa) with a previous inference algorithm (MF [11]) that is identical to LiMa, except that it maintains the ground state representation $p(S)$ directly instead of the lifted state representation. Furthermore, experiments with two baseline classifiers, quadratic discriminant analysis (QDA) and a hidden Markov model (HMM), have been performed.

The evaluation is *not* performed as a cross validation, which would have been methodologically infeasible considering the construction of the causal model. Therefore, to not place baseline models at a disadvantage, they are also built on the complete data.

VI. Results

Figure 2 shows the number of particles necessary to exactly represent the filtering distribution $p(S_t|y_{1:t})$ over

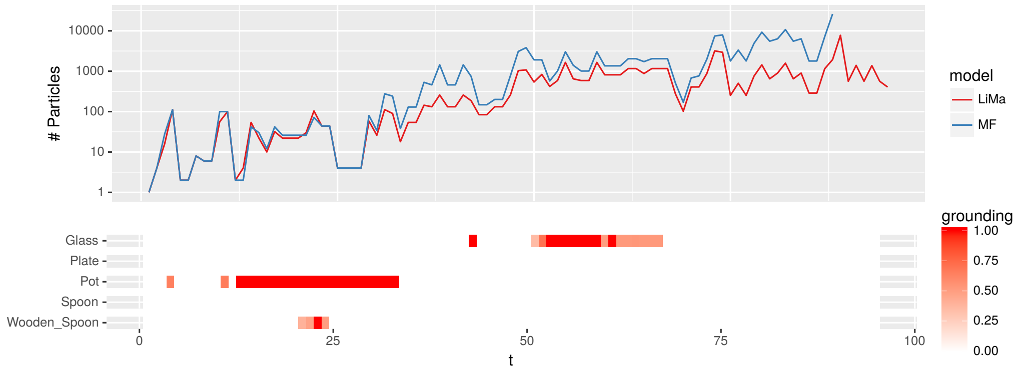

Fig. 2: Number of particles required during exact filtering (i.e. no pruning), using crisp action class observations, and grounding level of each object (fraction of particles in which the object is represented explicitly) for subject 6. Ground filtering is stopped at $t = 89$, because it exceeds 50,000 particles.

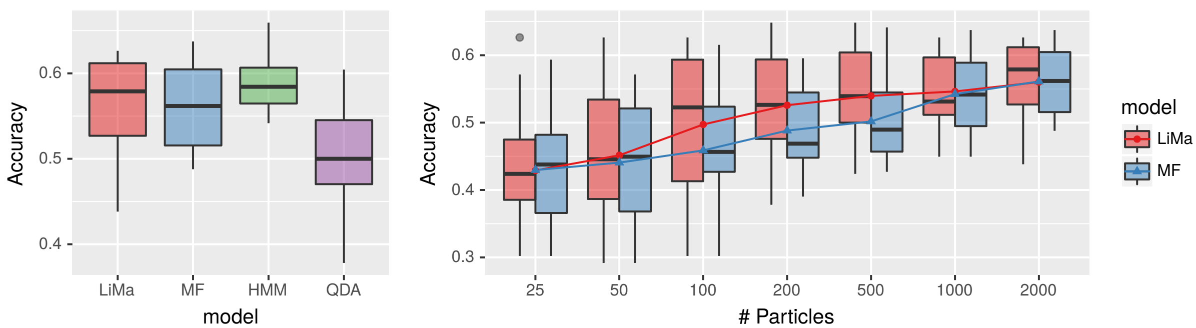

Fig. 3: Left: Comparison of activity recognition accuracy for baseline models, marginal filter (MF) and lifted marginal filter (LiMa), using 2000 particles. Right: Accuracies and their variance for lifted and ground inference. Each point represents the mean accuracy over all 7 datasets and 10 runs, given a limited number of available memory units. The boxes show the quartiles of the accuracies.

time for one specific subject (using the crisp action class observations). The figure also shows, for each object, in what fraction of the lifted states that object is represented explicitly, illustrating when splitting and merging operations have been performed. LiMa always requires at most the same number of particles than MF, and often less.

The figure shows that during inference, there are phases where certain objects need to be identified: Cooking is only possible when the pot is located on the stove, water can only be filled in the glass, and so on. In other phases, no object at all needs to be represented explicitly: For example, in the washing subtask, it is not necessary to know *which* object is washed, we only need do distinguish between clean and dirty objects. In these phases, ground inference is typically of high complexity, as permutation effects lead to a combinatorial explosion in the number of possible states. Lifted inference leads to a reduction in representational complexity here, by representing all object permutations of otherwise identical states in a single particle. In Figure 2, this can be seen from timestep 34 to 42 (preparing the table) and from timestep 70 onwards (washing the dishes).

Figure 3 (left) shows the mean accuracy of the baseline classifiers and the causal models (using 2000 particles). The HMM has a (slightly) better accuracy than the causal models (HMM: 0.59, LiMa and MF: 0.56). However, this 16-state HMM has 272 parameters (16 prior values and 16×16 transition matrix) – as no cross validation is performed in this study, we suspect that the HMM is subject to overfitting, and thus its accuracy is overestimated.

Figure 3 (right) shows the mean accuracy of MF and LiMa (for 7 subjects and 10 runs each) for different numbers of available particles. Both algorithms reach saturation at 1000 or 2000 particles, such that a further increase in the number of particles has no (or only a small) effect on accuracy. However, LiMa reaches saturation with fewer particles, such that the mean accuracy of LiMa is higher when 50, 100, 200 or 500 particles are available.

VII. Related Work

Lifted probabilistic inference [4] describes a class of inference algorithms that exploit symmetries and redundancies of the distributions for efficient inference. Lifted inference can be seen as decomposing a distribution into exchangeable components and handling sufficient statistics of them [9]. In LiMa, the same concepts are used, as we decompose $p(S)$ into products of exchangeable distributions $p(V|R)$ and a remaining part $p(R)$. Ideas from lifted inference have been applied to dynamic domains in the relational Kalman filter [3], which is restricted to Gaussian filtering distributions and a linear transition model.

The primary reason that prevents the direct application of lifted inference algorithms for inference in our causal models are the hard constraints in the transition semantics: Entities can only bind to actions when they satisfy the preconditions, only a single action is performed per timestep, etc.

A BF algorithm that, similar to LiMa, uses particles that each represent a *distribution* of specific states is the relational particle filter [10]. It uses particles where some variables have specific values and others are represented by parametric distributions. Opposed to LiMa, the approach is not concerned with efficient representations of *exchangeable* distributions, and thus cannot efficiently handle the identity permutations occurring in our application domains. Furthermore, in cases where LiMa performs a split, the algorithm samples from the corresponding distributions, instead of manipulating the exact distributions on a parametric level.

VIII. Conclusion

In this paper, we investigated whether LiMa can be applied successfully to a real-world activity and context recognition task. We answered this question affirmatively, by showing that LiMa needs fewer particles to represent the exact filtering distribution (specifically, in phases where ground inference requires the largest number of particles due to permutation effects), and that it can be more accurate than conventional ground inference (MF). Still, LiMa retains the advantages of causal, symbolic models in comparison to data-driven methods (reduced need for training data, reasoning about the context of activities).

However, this study is still only a fist step for using lifted filtering methods in real, online activity and context recognition systems, e.g. as part assistive systems: First, we did not attempt to properly model the duration distribution of the actions, which means that accuracy values obtained here are not directly comparable to results obtained in other studies (this affects all methods equally, therefore model comparison within this study should be valid). How to model duration distributions in symbolic models has already been investigated in previous studies [7] and similar concepts can be applied to LiMa.

Most importantly, the merging operation we used here is domain-specific, requiring prior knowledge about the causal structure of the domain. Developing more general merging operations, that do not require domain knowledge (e.g. by analyzing the causal structure automatically) is a topic for future research.

References

[1] D. Arnaud, N. de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag New York, 2001.

[2] R. Barbuti, F. Levi, P. Milazzo, and G. Scatena. Maximally Parallel Probabilistic Semantics for Multiset Rewriting. *Fundamenta Informaticae*, 112(1):1–17, 2011.

[3] J. Choi, E. Amir, T. Xu, and A. Valocchi. Learning Relational Kalman Filtering. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2539–2546, 2015.

[4] L. De Raedt, K. Kersting, S. Natarajan, and D. Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2):1–189, 2016.

[5] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[6] Frank Krüger, Albert Hein, Kristina Yordanova, and Thomas Kirste. Recognising user actions during cooking task (cooking task dataset) – imu data. University Library, University of Rostock, 2017.

[7] Frank Krüger, Martin Nyolt, Kristina Yordanova, Albert Hein, and Thomas Kirste. Computational State Space Models for Activity and Intention Recognition. A Feasibility Study. *PLOS ONE*, 9(11):e109381, November 2014.

[8] Stefan Lüdtke, Max Schröder, Sebastian Bader, Kristian Kersting, and Thomas Kirste. Lifted Filtering via Exchangeable Decomposition. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018.

[9] Mathias Niepert and Guy Van den Broeck. Tractability through exchangeability: A new perspective on efficient probabilistic inference. In *AAAI*, pages 2467–2475, 2014.

[10] D. Nitti, T. De Laet, and L. De Raedt. A particle filter for hybrid relational domains. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2764–2771. IEEE, November 2013.

[11] Martin Nyolt and Thomas Kirste. On Resampling for Bayesian Filters in Discrete State Spaces. In *Proceedings of the 27th International Conference on Tools with Artificial Intelligence*, pages 526–533, Vietri sul Mare, Italy, November 2015. IEEE Computer Society.

[12] Martin Nyolt, Frank Krüger, Kristina Yordanova, Albert Hein, and Thomas Kirste. Marginal filtering in large state spaces. *International Journal of Approximate Reasoning*, 61:16–32, June 2015.

[13] D. Poole. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 985–991, 2003.

[14] Simo Särkkä. *Bayesian Filtering and Smoothing*, volume 3. Cambridge University Press, 2013.

[15] Max Schröder, Stefan Lüdtke, Sebastian Bader, Frank Krüger, and Thomas Kirste. Sequential Lifted Bayesian Filtering in Multiset Rewriting Systems. In *UAI Workshop: Statistical Relational Artificial Intelligence*, 2017.

[16] Shun-Zheng Yu. Hidden semi-markov models. *Artificial intelligence*, 174(2):215–243, 2010.