Lifted Filtering via Exchangeable Decomposition

Stefan Lüdtke[1], **Max Schröder**[1], **Sebastian Bader**[1], **Kristian Kersting**[2], **Thomas Kirste**[1]
[1]Institute of Computer Science, University of Rostock, Germany
[2]Computer Science Department and Centre for Cognitive Science, TU Darmstadt, Germany
{stefan.luedtke2, max.schroeder, sebastian.bader, thomas.kirste}@uni-rostock.de
kersting@cs.tu-darmstadt.de

Abstract

We present a model for exact recursive Bayesian filtering based on lifted multiset states. Combining multisets with lifting makes it possible to simultaneously exploit multiple strategies for reducing inference complexity when compared to list-based grounded state representations. The core idea is to borrow the concept of Maximally Parallel Multiset Rewriting Systems and to enhance it by concepts from Rao-Blackwellization and Lifted Inference, giving a representation of state distributions that enables efficient inference. In worlds where the random variables that define the system state are exchangeable – where the identity of entities does not matter – it automatically uses a representation that abstracts from ordering (achieving an exponential reduction in complexity) – and it automatically adapts when observations or system dynamics destroy exchangeability by breaking symmetry.

1 Introduction

Modeling dynamical systems is fundamental for the understanding of complex phenomena in a variety of AI tasks. *Multiset Rewriting Systems (MRSs)* provide a convenient mechanism to represent dynamic systems that consist of multiple interacting entities which can be grouped into "species". MRSs can be used to model biochemical reactions [Barbuti *et al.*, 2011], population dynamics in ecological studies [Pescini *et al.*, 2006], network protocols [Cervesato *et al.*, 1999], etc. Or consider *human activity recognition (HAR)* [Bulling *et al.*, 2014], the target of our present paper, and assume that we are interested in reconstructing structured activities of one or more human protagonists (pursuing everyday activities) from noisy or ambiguous sensor data. Here, multisets naturally arise when we want to represent states that encompass multiple persons, or multiple objects that are handled by persons, which can not be discriminated by observations. In HAR, one established method for deriving state distributions from sequences of observations is *recursive Bayesian state estimation* (RBSE), for instance based on various kinds of enhanced hidden Markov models [Bui *et al.*, 2002; Liao *et al.*, 2007] or particle filtering [Fox *et al.*, 2003; Krüger *et al.*, 2014]. RBSE iteratively computes the posterior $p(S_t \mid y_{1:t})$ for time t from the previous posterior $p(S_{t-1} \mid y_{1:t-1})$ at time $t-1$ and an observation y_t.

In application domains such as chemistry, cell biology or ecology, MRSs are typically used for simulation studies. However, MRSs should in principle also allow the online integration of sensor data as required by HAR and as provided by recursive Bayesian state estimation. However, when we try to apply established methods to systems whose dynamics is represented by MRSs, we find that they cannot efficiently represent the exact posterior, in terms of the amount of storage required.

In this paper, we present a novel algorithm for exact RBSE for systems whose state space and dynamics can be represented by MRSs, which we call *Lifted Marginal Filtering* (LiMa). The central technical idea of LiMa is to introduce a suitable factorized representation of distributions over multisets which allows to represent some factors in a parametric way rather than by samples or by complete enumeration, similar to *Rao-Blackwellization* as used in particle filters [Doucet *et al.*, 2000]. Interestingly, the prediction and update steps of RBSE can be performed directly on the factorized representation, without resorting to the original, much larger distribution, by exploiting exchangeability [Niepert and Van den Broeck, 2014]. In certain cases (computing applicable actions, and when conditioning on *identifying* observations), the representation of the factorized distribution needs to be manipulated by *splitting* operations, similar to splitting used in exact lifted inference algorithms [Poole, 2003].

Depending on the underlying model, this approach reduces the number of explicitly represented states s_t by orders of magnitude in comparison to an approach using samples or enumeration of the s_t. This is possible because LiMa combines two effects: The multiset state representation, which allows to exploit exchangeability (Lifted Inference), and the factorization of the multiset state distribution, which allows manipulating the state distribution on a parametric level (Rao-Blackwellization). To the best of our knowledge, this is the first attempt to provide RBSE for systems with MRS dynamics and the first attempt that allows to perform prediction and update directly (at least partially) on the factorized representation.

The paper is organized as follows: We continue by briefly introducing *Probabilistic Maximally Parallel Multiset*

Rewriting Systems (PMPMRS) in Section 2. The factorization of distributions over multisets is introduced in Section 3. In Section 4, we show how RBSE can be performed on the factorized representation, without needing to enumerate the original states. Our approach is demonstrated on two multi-agent activity recognition tasks in Section 5.

2 Background

In the following, we briefly introduce some background on multiset rewriting systems (MRSs).

Multisets: Let $\mathcal{E}$ be a set of species. A multiset (over $\mathcal{E}$) is a map $s : \mathcal{E} \to \mathbb{N}$ from species to multiplicities (natural numbers). Let s, s' be two multisets, let multiset union $s \uplus s'$, multiset difference $s \mathbin{\dot\cup} s'$ and multiset subset relation $s \sqsubseteq s'$ be defined the obvious way. For species $s_1, \ldots, s_k \in \mathcal{E}$ and multiplicities $n_1, \ldots, n_k \in \mathbb{N}$ where $n_i > 0$ we write $[\![\, n_1 e_1, \ldots, n_k e_k \,]\!]$ to denote a multiset, where the multiplicity of e_i is n_i and the multiplicities of all species not listed is 0.

Actions: For now, it is sufficient to consider a rewriting rule as a pair of multisets (p, e), where p are the prerequisites and e is the add list. The action (also known as *rewriting rule*) (p, e) is applicable to a multiset m if $p \sqsubseteq m$. The result of applying (p, e) to m is the multiset $(m \mathbin{\dot\cup} p) \uplus e$.

Compound Actions: In scenarios where multiple entities (inter-)act simultaneously, multiple actions may take place between consecutive time steps. This intuition is captured by *compound actions*, that describe the transition semantics of *maximally parallel multiset rewriting systems* (MPMRSs) [Barbuti *et al.*, 2011]: Each individual takes part in an action if possible, and all actions are performed in parallel. Specifically, a compound action is a multiset of actions. The compound action c is *applicable* in a state s if all prequisites are present in s, i.e. $\biguplus_{(p_i,e_i)\in c} p_i \sqsubseteq s$. The compound action c is maximal if no action a can be added such that $c \uplus [\![\, 1a \,]\!]$ is still applicable. The result s' of applying a compound action c to a state s is the union of the effects of the individual actions: all prerequisites are removed from s, and all add lists are inserted, i.e., $s' = (s \mathbin{\dot\cup} p') \uplus e'$ with $p' = \biguplus_{(p_i,e_i)\in c} p_i$ and $e' = \biguplus_{(p_i,e_i)\in c} e_i$.

Sampling: *Probabilistic* MRSs assign a *weight* to each action. Given a state s, these weights, and the number of possibilities the compound actions can be instantiated, define a distribution of compound actions $p(C \mid S)$. More details are provided in [Barbuti *et al.*, 2011]. Using this distribution, it is easy to draw sample trajectories: Given a state s, we calculate all applicable compound actions and their probabilities, sample a single compound action from $p(C \mid S{=}s)$, and apply it to s. This process is iterated for the resulting state s'.

3 Factorizing Distributions over Multisets of Structured Entities

In this and the following section, we present the main technical contribution: The factorization of distributions of multisets over structured entities, and a probabilistic maximally parallel MRS (PMPMRS) that operates directly on this factorized, parametric representation.

3.1 Problem Statement

We start by outlining the problem that makes the factorized representation necessary. For RBSE, we are not interested in *sampling* trajectories of states (as outline above), but in maintaining a distribution of multiset states.

The question is how to efficiently represent such a distribution $p(S{=}s)$. If the set of species $\mathcal{E}$ is small and finite, the number of multisets over $\mathcal{E}$ with nonzero support will typically also be small. Therefore, we can simply maintain a set of tuples (s_i, p_i) that represent a categorical distribution. However, if $\mathcal{E}$ is large or even continuous and thus also the number of multisets over $\mathcal{E}$, storing the resulting large or infinite number of tuples (s_i, p_i) becomes infeasible.

The conventional solution for distributions over *metrical* random variables is to use *parametric* distributions that can be represented and manipulated efficiently on the syntactical level, i.e. by storing and manipulating just the parameters of the distribution. For example, a Kalman filter uses $p(S) \propto \mathcal{N}(\mu, \sigma^2)$ to represent a distribution over continuous states by storing and manipulating μ and σ^2. Unfortunately, multisets do not necessarily possess a metrical structure that allows to use parametric distributions. In our approach, we decompose the multisets into a metrical part that allows for parametric distributions, and a remaining, discrete part that that can be represented by a categorical distribution with small (or at least finite) support. We achieve this by introducing *structured* entities that allow for such a decomposition.

3.2 Structured Species

An *entity* (a species with internal structure) is a map of property names $\mathcal{K}$ to values $\mathcal{V}$, i.e. a partial function $\mathcal{E} = \mathcal{K} \nrightarrow \mathcal{V}$. This is a necessity for the scenarios we are considering, as they contain entities with multiple, possibly continuous, properties. For example, an entity might have a continuous location, e.g. a real number. Using *flat* (unstructured) species, this would require to introduce an infinite number of possible species, and potentially also to an infinite number of actions (as each action prerequisite must be a specific species). Using structured entities, the action prerequisites can be expressed more succinctly as constraints on the entities' properties, as described below. Another reason for using structured entities is that they allow for factorizing the distributions of multisets.

A multiset over entities $\mathcal{E}$ is a state of our MRS. Specificaly, we call a multiset $s \in \mathcal{S}$ a *ground state*. For example, the following state describes a situation in a person tracking scenario where two entities, named Alice and Bob, are at the continuous locations 1.3 and 2.1[1]:

$$[\![\, 1\langle \text{N: } \textit{Alice}, \text{L: } 1.3\rangle, 1\langle \text{N: } \textit{Bob}, \text{L: } 2.1\rangle \,]\!] \qquad (1)$$

3.3 Factorising Multiset Distributions

Suppose we can decompose the state s into two parts t and v, such that there is a bijection from s to tuples (t, v). Then, a state distribution can be factorized as

$$p(S) = p(T, V) = p(T)\,p(V \mid T). \qquad (2)$$

[1]For simplicity, the location is a continuous univariate number; in realistic scenarios we might e.g. use 2D locations.

This idea is independent of multisets and is called Rao-Blackwellization [Doucet *et al.*, 2000]. The question is how to decompose multiset states efficiently, such that the decomposition leads to a more efficient representation of the distribution: The factor $p(V \mid T)$ is handled parametrically, while $p(T)$ has a smaller support than $p(S)$.

Decomposition: The structured species described in Section 3.2 allow 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. More formally, the decomposition is performed as follows: A structure $t \in \mathcal{T}$ (a multiset over $\mathcal{E}_\tau$) is a multiset of entities where the property values are variables, instead of specific values (called *entity structures* $\mathcal{E}_\tau$). The values $v \in \mathcal{W}$ are a list of specific values (of the properties). For example, the state from Equation 1 is decomposed into:

$$\begin{aligned} t =& [\![1\langle \text{N: } n_1, \text{L: } l_1 \rangle, 1\langle \text{N: } n_2, \text{L: } l_2 \rangle]\!] \\ v =& (\text{Alice}, 1.3, \text{Bob}, 2.1) \end{aligned} \tag{3}$$

Given t and v, the state s can be constructed by assuming a canonical order of entities in t (e.g. the lexicographic order) and by replacing each variable by the corresponding value in v (i.e the i-th variable is replaced by the i-th value). This describes a bijection between (t, v) and s, which allows us to represent $p(s)$ in a factorized way, according to Equation 2.

Distributions of structure and values: Performing RBSE inference requires that the distributions $p(T)$ and $p(V \mid T)$ can be represented finitely. For $p(T)$, this is straightforward: Given that $p(S)$ is a categorical distribution with finite support, we can also represent $p(T)$ as a finite categorical distribution, simply because multiple elements of $\mathcal{S}$ are mapped to a single element of $\mathcal{T}$.

We assume that $p(V \mid T)$ is a product of m parametric distributions with parameters θ such that $p(V \mid T) = \prod_i p_i(V_i \mid T, \theta_i)$. Every p_i describes the distribution of one or multiple properties and can be represented by the parameters θ_i, and an indicator signifying which parametric form the distribution has. We call $\rho(p_i) \in \mathcal{R}$ the *representation* of p_i, and $\mathcal{R}$ the representation space. For example, $\mathcal{U}(A, B)$ represents an urn without replacement, containing the elements A and B. How to represent the complete value distribution, i.e. $\rho(p(V \mid T))$, will be discussed next.

Property-Distribution Association: The remaining question is how to associate property values in t with the random variables in $p(V \mid T{=}t)$. At first, this may seem obvious: We order the entities in t, and associate the i-th property with random variable v_i, as suggested by Equation 3. However, this is not sufficient in many cases, as there might be *non-local* dependencies of multiple values: For example, in Equation 3, the distributions of the names of both entities are not independent, assuming that all names are unique. Thus, we must make sure that exactly the name properties are distributed according to a joint distribution (e.g. an urn without replacement). To succinctly represent which property values are associated with which distributions, we propose a labeling mechanism that provides this association. We introduce this mechanism by describing $\mathcal{E}_\tau$ and the representation of the value distribution, $\rho(p(V \mid T))$ in more detail.

An *entity structure* is a map from property names $\mathcal{K}$ to labels $\mathcal{D}$, i.e. $\mathcal{E}_\tau = \mathcal{K} \nrightarrow \mathcal{D}$. The distribution $p(V \mid T)$ is represented as a map of labels $\mathcal{D}$ to distribution representations $\mathcal{R}$. Note that these labels are essentially pointers. We call the representation of $p(V \mid T)$ the *context* $c_t = \rho(p(V \mid T))$ of t.

This mechanism allows us to easily represent correlations of properties, even when the properties belong to different entities, without the need to define an order of the entities in t. This is illustrated in the following example.

Example: Suppose we know that two persons are present in a situation, and we have normally distributed location estimates for both persons (e.g. based on a measurement), but we do not know which specific person corresponds to which location estimate. This situation can be described by the following factorized state representation, i.e. the pair of structure t and context c_t:

$$\begin{aligned} t &= [\![1\langle \text{N: } \mathbf{N}, \text{L: } \mathbf{L_1} \rangle, 1\langle \text{N: } \mathbf{N}, \text{L: } \mathbf{L_2} \rangle]\!] \\ c_t &= \langle \mathbf{N} : \mathcal{U}(\text{A,B}), \mathbf{L_1} : \mathcal{N}(1.3, 2.0), \mathbf{L_2} : \mathcal{N}(2.1, 1.0) \rangle \end{aligned} \tag{4}$$

Note how in the example, we see how the same distribution $\mathbf{N}$ can be referenced multiple times in t, which means that the corresponding properties are distributed according to a joint distribution. On the other hand, properties that reference *different* distributions are independent. In the example, the name of an entity is independent of the entity's location.

Exchangeability: Using the labeling mechanism, we do not rely on an order of entities, i.e. the order of the entities in t is arbitrary. Therefore, we require all joint distributions in the context to be *exchangeable*, i.e. $p(x_1, x_2) = p(x_2, x_1)$. From the view of Lifted Inference, the context thus represents an exchangeable decomposition [Niepert and Van den Broeck, 2014] of the value distribution. This property is the reason that allows efficient filtering, as will be explained in Section 4.

Lifted State: Using our techniques, we can represent the categorical distribution $p(S) = p(T, V)$ by: (i) A categorical distribution of structures $p(T)$ (i.e. a set of tuples (t, p)) and (ii) for each t, a context c_t, representing $p(V \mid T)$. Instead of storing a set of tuples (t, p) and a context c_t for each t, equivalently, we can directly store a set of triples (t, p, c_t). However, this is simply a categorical distribution of pairs (t, c_t). Thus, the distribution $p(S)$ can be represented by a categorical distribution of such pairs (t, c_t). We call $l = (t, c_t)$ a *lifted state*, and $p(L)$ a *lifted state distribution*.

Each lifted state l describes a distribution over $\mathcal{S}$ where all s have the same structure t and all v are distributed according to $p(V \mid T{=}t)$. Note that the structures t in a lifted state distribution $p(L)$ need not be distinct (due to splitting, see Section 4.3). Thus, a lifted state distribution $p(L{=}l)$ with $l = (t, c_t)$, $\rho(p_i(V_i \mid t)) \in \text{range}\, c_t$ and $p(V \mid t) = \Pi_i p_i(V_i \mid t, c_t)$ describes a distribution of states s as follows[2]:

$$p(S{=}s) = p(T{=}t, V{=}v) = \sum_{\{l_i = (t_i, c_i) \mid t_i = t\}} p(L{=}l_i)\, p(V{=}v \mid t_i) \tag{5}$$

[2]For $p(V \mid T) = \prod_i p_i(V_i \mid T)$, we assume that the V_i are assigned to the p_i according to their labels: All V_i that have label d_j in t are distributed according to the distribution with label d_j in c_t.

4 Lifted Filtering via Exchangeable Decomposition

In the following, we present a RBSE algorithm that uses the factorized multiset distribution to represent the current state distribution. Given a prior distribution of states $p(S_t \mid y_{1:t})$, the calculation of the posterior distribution after observing y_{t+1}, i.e. $p(S_{t+1} \mid y_{1:t+1})$ can be decomposed into the following two steps: The *predict* step calculates the distribution after applying the *transition model* $p(S_{t+1} \mid S_t)$, i.e. $p(S_{t+1} \mid y_{1:t}) = \sum_{s_t} p(S_{t+1} \mid S_t{=}s_t)\, p(S_t{=}s_t \mid y_{1:t})$. Afterwards, the posterior distribution is calculated by employing the *observation model* $p(y_{t+1} \mid S_{t+1})$:

$$p(S_{t+1} \mid y_{1:t+1}) = \frac{p(y_{t+1} \mid S_{t+1})\, p(S_{t+1} \mid y_{1:t})}{p(y_{t+1} \mid y_{1:t})} \qquad (6)$$

Interestingly, these steps can be performed directly on the lifted states. This is possible because for multiset rewriting, it is only necessary to know *how many* entities have a certain property, but their specific order or identity is not relevant. It has been shown that exactly such exchangeability properties (reflected by exchangeability of the value distribution) allow efficient Lifted Inference [Niepert and Van den Broeck, 2014].

4.1 Predict

For the predict step, we need to define the transition model $p(S_{t+1} \mid S_t)$. The dynamics of the system is described in terms of compound actions, as introduced in Section 2. However, we have to account for the structured entities (leading to more complex preconditions and effects) and the lifted state representation (requiring *splitting* when the preconditions are indeterminate, explained in Section 4.3). In the following, these concepts are introduced more formally.

Actions describe the behavior of the entities. An action is a pair (c, e, w) consisting of a precondition list $c \in \mathcal{C}$ and an effect function $e \in \mathcal{F}$. A precondition list is a list of constraints on entity structures and their corresponding values, i.e. boolean functions: $\mathcal{C} = [\mathcal{E}_\mathcal{T} \times V \to \{\top, \bot\}]$. The idea of applying an action to a lifted state is to *bind* entities to the preconditions. Specifically, one entity is bound to each element in the precondition list, and entities can only be bound when they satisfy the corresponding constraint. The effect function then manipulates the state based on the bound entities (by inserting, removing, or manipulating entities or the distributions stored in the context). We call such a binding **action instance**, i.e. an action instance is a pair of an action and a list of entity structures. Entity structures can be indeterminate regarding a precondition, as some values v drawn from $p(V \mid T)$ (represented by c_t) may satisfy the precondition, while others do not. This case requires *splitting*, described in Section 4.3. For now, we assume that the preconditions are determinate.

A Compound Action $k \in \mathcal{K}$ is a multiset of action instances. It is applied to a state by composing the effects of the individual action instances. In the following, we are only concerned with applicable maximal compound action (AMCAs, see Section 2), which define the transition model.

Compound action probabilities: Our system is probabilistic, which means that each AMCA is assigned a probability. In general, any function from the AMCAs to positive real numbers which integrates to one is a valid definition of these probabilities, that might be plausible for different domains. Here, we use the probabilities that arise when each entity independently chooses which action to participate in (which is the intended semantics for the activity recognition problems we are concerned with). In this case, the probability of each compound action is based on the number of options specific entities can be bound to the preconditions of the actions, and a weight for each action. For details, we refer to [Barbuti *et al.*, 2011].

Transition model: The distribution of the AMCAs define the distribution of successor states, i.e. the *transition model*. The successor states of l are obtained by applying all AMCAs to l. The probability of each successor state l' is the sum of the probabilities of all AMCAs leading to l':

$$p(L'{=}l' \mid L{=}l) = \sum_{\{k \mid apply(k,l)=l'\}} p(K{=}k \mid L{=}l) \qquad (7)$$

Thus, given a prior lifted state distribution and a set of actions, the following steps have to be performed to obtain the posterior lifted state distribution: (i) Split the lifted states until all preconditions are determinate, see Section 4.3, (ii) compute all action instances of each action, (iii) compute all AMCAs and their probabilities, (iv) apply all AMCAs to the lifted state, (v) calculate the probabilities of the resulting successor states.

Step (ii) can be solved by simple forward constraint satisfaction. Step (iii) can also be implemented by a search-based approach, i.e. building the compound actions by incrementally adding all applicable action instances. Additional efficiency can be gained by noting that multiset insertion is commutative. Therefore, we can define an arbitrary order of the action instances and when incrementally building the compound actions only insert instances that are “not smaller” than the last inserted action instance.

4.2 Update

Given a state distribution $p(S \mid y_t) = p(T, V \mid y_t)$ (which might be represented by a lifted state representation via Equation 5), the update step can be decomposed into the update of the structure, and the update of the value distribution. This becomes obvious by applying the chain rule and Bayes' theorem to the distribution:

$$\begin{aligned} p(S \mid y_t) &= p(T, V \mid y_t) \\ &= p(T \mid y_t)\, p(V \mid y_t, T) \\ &= \underbrace{\frac{p(y_t \mid T) p(T)}{p(y_t)}}_{\text{(a)}} \underbrace{\frac{p(y_t \mid V, T) p(V \mid T)}{p(y_t \mid T)}}_{\text{(b)}} \end{aligned} \qquad (8)$$

The factor (a) corresponds to the update of the structure, and (b) is the update of the value distribution. The values $p(y_t)$ and $p(y_t \mid T)$ are normalization factors that can be obtained by marginalization. Both factors of the update can be computed on the lifted state representation: Step (a) means that the weight of each structure (i.e. the weight of the corresponding lifted state) is multiplied by observation probability $p(y_{t+1} \mid T)$. Step (b) corresponds to syntactically manipulating of each context c_t, i.e. modifying the parametric distribution $p(V \mid t)$. For example, if $P(V \mid T)$ and $p(y_{t+1} \mid V, T)$ follow normal distributions, this corresponds to the standard Kalman filter update.

When $p(V \mid y_{t+1}, T)$ can no longer be represented by a product of exchangeable parametric distributions, we have to *split* the lifted state, described in Section 4.3. For example, regarding the situation in Equation 4, when observing a speficic person (say, Alice) at a specific location, the name and location are no longer independent.

4.3 Splitting

During action instance computation and during the update step, two closely related problems arise: (1) While computing action instances, a precondition can be satisfied for some ground states s represented by a lifted state l, and not satisfied by others which are also represented by l. (2) Due to observations, the resulting ground state distribution previously represented by a lifted state may no longer be representable by a single lifted state. In general, the ground states form partitions based on whether a constraint is satisfied or not (in action instance computation) or whether they can be represented exactly by a single lifted state using an exchangeable parametric distribution for $p(V \mid T)$ in the update step.

We want to compute lifted states that describe the partitions, without requiring a complete enumeration of all ground states first. This is done by manipulating the lifted states by an operation called *splitting*. More specifically, splitting a lifted state l results in a set of lifted states L such that: (1) L describes the same distribution of ground states as l, and (2) for each $l_i \in L$, all ground states s_i described by l_i lie in the same partition. How splitting is done exactly depends on the parametric form of $p(V)$. In the following, we describe splitting for urns without replacement on equality constraints:

Suppose we want to split the property q on the constraint "$q = v$". Intuitively, the strategy is to ground the values of the specific property q. Suppose the values of q are distributed according to an urn without replacement with n different values, $\mathcal{U}(v_1, \ldots, v_n)$. We split this situation into n worlds, where in world i, q has value v_i. For example, splitting the state in Equation 4 results in two lifted states, one where Alice is at the location distributed according to L_1, and one where Alice is at the location distributed according to L_2. Splitting into n lifted states (instead of two lifted states, one where the precondition is satisfied for e and one where it is not satisfied) is necessary to preserve the independence and exchangeability properties of the remaining urn. The general procedure for splitting urns is shown in Algorithm 1. Note that splitting only affects the specific property that we split on, but other properties are still represented in lifted form. The approach is conceptually similar to splitting parametric factors in First-order Variable Elimination [Poole, 2003].

Splitting rules for other kinds of parametric distributions can be developed by following the same idea. In general, we can split a distribution when a corresponding conditional distribution (conditioned on the constraint that we split on) is still exchangeable and can still be represented parametrically.

Algorithm 1 For lifted state l, split the property $q \in \mathcal{K}$ with label $d \in \mathcal{D}$, distributed according to urn u.

```
1:  function SPLIT-URN($l=(t,c)$,$q$,$d$,$u$)
2:      for value $v$ in $u$ do
3:          $u' \leftarrow u$ without $v$
4:          $e' \leftarrow e \oplus \langle q: d' \rangle$
5:          $c' \leftarrow c \oplus \langle d: u', d': \delta_v \rangle$
6:          $t' \leftarrow t \uplus [\![ 1e ]\!] \uplus [\![ 1e' ]\!]$
7:          $l' \leftarrow (t', c')$
8:          $p(l') \leftarrow$ Probability of $v$ in $u$
9:      end for
10:     return Categorical Distribution $p(L')$
11: end function
```

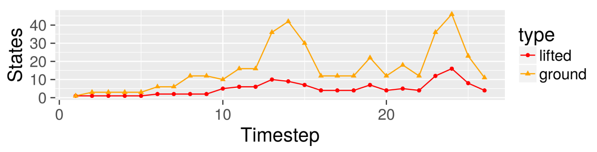

Figure 1: Number of lifted and grounded states during inference for the first scenario.

5 Qualitative Examples

In this section, we provide an intuition for the qualitative effect of the lifted state representation, by comparing LiMa with an *exact propositional* filtering approach that maintains $p(S)$ by complete enumeration. We compare the the cardinality $p(L)$ (handled by LiMa) and $p(S)$ – which is an indicator of space and time complexity. We do not compare LiMa with other approaches, as they are either approximate, or cannot handle identifying observations at all.

We use two scenarios from the HAR domain for evaluation. The first scenario [Schröder *et al.*, 2017] consists of simulated sensor data of three persons acting in an office environment observed by presence sensors. For each person, their name, location and whether they hold an object is modeled. The simulated sensor data do not reveal the *identity* of the persons nor the number of persons per location. At timestep $t = 10$, an *identifying* observation is been made (Alice is at the printer), resulting in a split of the lifted states on the predicate "Name=Alice".

The second scenario is a person tracking task with anonymous sensors (similar to the previous scenario), but uses real sensor data of PIR and light switch sensors. The data consists of 35 observation sequences of 1 to 7 agents (5 each) who

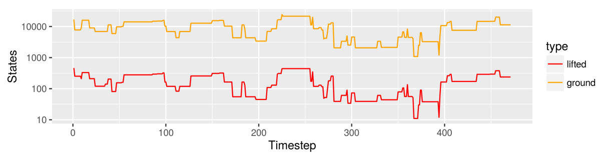

Figure 2: Number of lifted and grounded states during inference for the second scenario.

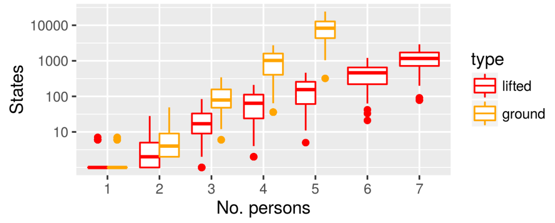

Figure 3: Mean number of states for second scenario for different number of persons.

move between 14 rooms in an office. The dataset is available at [Kasparick and Krüger, 2013]. We model both scenarios by entities whose name properties are distributed according to a finite urn. Both scenarios have a compound action semantics, as all persons can simultaneously move between observations.

Figures 1 and 2 show the number of states necessary to represent the posterior $p(S_t)$ over time. For both scenarios, using the lifted state representation, the number of states represented explicitly is several times smaller. Note that the ground and the lifted states represent exactly the same distribution (see Equation 5). Figure 3 shows the mean number of states occurring during inference for the second scenario, for different numbers of agents. Here, we see that the effect gets more pronounced for larger number of agents. The reason for this is the increase of the number of ground states due to the factorial number of entity-name associations (that are all represented by a single lifted state).

The number of explicit state representations linearly corresponds to runtime, as compound action computation (which is the most computationally expensive part of the filtering algorithm) has to be performed individually for each state. Thus, the empirical results clearly demonstrate that LiMa can be order of magnitude faster.

6 Related Work

Another class of approaches concerned with performing inference over compact representations of probability distributions is known as *Lifted Probabilistic Inference* [De Raedt *et al.*, 2016]. Lifted Inference can be seen as decomposing a distribution into exchangeable components and handling sufficient statistics of them [Niepert and Van den Broeck, 2014]. The exchangeable decomposition is made explicit in our approach – each factor of the value distribution is exchangeable. We are then also only interested in sufficient statistics, namely *how many* entities have a certain property.

Ideas of Lifted Inference have been applied to RBSE (i.e. to dynamic domains) in the Relational Kalman Filter [Choi *et al.*, 2015], an approach that is restricted to a Gaussian state distribution and a linear transition model.

The primary reason that makes it difficult to directly apply Lifted Inference algorithms to MRSs are the hard constraints that are present in the transition semantics: which entity can perform which actions, the fact that each entity must perform exactly one action etc. Interestingly, computing compound actions is a special case of Lifted Weighted Model Counting [Gogate and Domingos, 2012], that additionally considers such hard constrains.

There are a number of other approaches that aim at finding compact descriptions of sets of states in dynamic systems, like *Relational POMDPs* [Sanner and Boutilier, 2009] and *Logical Filtering* [Shirazi and Amir, 2011]. They employ situation calculus to describe states and actions, but are not explicitly concerned with efficient filtering. The idea of using independent factors for RBSE in multi-agent settings is also explored by [Pfeffer *et al.*, 2009], but this approach needs to multiply the factors and then sample from the joint, whereas LiMa only resorts to the joint distribution when necessary.

An RBSE algorithm that, similar to LiMa, uses state descriptions that each represents a set of specific states is the *Relational Particle Filter* [Nitti *et al.*, 2013]. It uses states where some variables are described by specific values and others are represented by parametric distributions. Similar to LiMa, the approach can handle continuous and infinite domains. In cases where we require a split, the algorithm samples from the corresponding distributions, instead of manipulating the exact distributions on a parametric level.

Stochastic Relational Processes (SRPs) [Thon *et al.*, 2011] use causal probabilistic logic to describe the transition model of a RBSE algorithm, i.e. by a set of probabilistic precondition-effect rules. Opposed to the factorized multiset states used in LiMa, SRPs use a ground state representation – although a part of the transition model can be calculated in a lifted way (for calculating the successor states, not all ground rules need to be generated).

7 Conclusion

We presented LiMa, a recursive Bayesian state estimation (RBSE) algorithm that uses a Probabilistic Maximally Parallel Multiset Rewriting System to model the underlying state dynamics. It uses a *factorized* representation for distributions of multisets and exploits exchangeability to perform the complete RBSE cycle on this compact representation, without needing to sample from or to enumerate all ground states. Empirical evidence in two activity recognition domains shows that LiMa needs to represent a much lower number of states explicitly.

There are scenarios (e.g. containing many identifying observations) that might require repeated splitting until the state representation is completely grounded, which is a well-known problem in the filtering literature [Boyen and Koller, 1998]. Therefore, future research will concentrate on methods that allow to maintain the factorized representation when the independence assumptions hold only approximately, methods that re-introduce factorized representations (similar to *merging* in Lifted Inference, e.g. [Kersting *et al.*, 2009]), and using other representations for $p(V|T)$, like Sum-Product Networks [Poon and Domingos, 2011] or Exchangeable Variable Models [Niepert and Domingos, 2014]. A further research goal is to investigate whether the multiplicities of entities can also be represented by parametric distributions, which would lead to an even more compact representation of distributions.

Acknowledgements

This work was supported by the *Deutsche Forschungsgemeinschaft* (DFG), grant number INST 264/128-1 FUGG.

References

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

[Boyen and Koller, 1998] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 33–42, 1998.

[Bui *et al.*, 2002] Hung Hai Bui, Svetha Venkatesh, and Geoff West. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research*, 17:451–499, 2002.

[Bulling *et al.*, 2014] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.

[Cervesato *et al.*, 1999] I. Cervesato, N. A. Durgin, P. D. Lincoln, J. C. Mitchell, and A. Scedrov. A Meta-Notation for Protocol Analysis. In *Proceedings of the 12th IEEE Workshop on Computer Security Foundations*, CSFW '99, pages 55–, Washington, DC, USA, 1999. IEEE Computer Society.

[Choi *et al.*, 2015] 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.

[De Raedt *et al.*, 2016] 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.

[Doucet *et al.*, 2000] 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.

[Fox *et al.*, 2003] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE pervasive computing*, 2(3):24–33, 2003.

[Gogate and Domingos, 2012] Vibhav Gogate and Pedro Domingos. Probabilistic theorem proving. *arXiv preprint arXiv:1202.3724*, 2012.

[Kasparick and Krüger, 2013] Martin Kasparick and Frank Krüger. Probabilistic action selection - tracking multiple persons in indoor environments. `http://dx.doi.org/10.18453/rosdok_id00000114`, 2013.

[Kersting *et al.*, 2009] K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 277–284, 2009.

[Krüger *et al.*, 2014] 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.

[Liao *et al.*, 2007] Lin Liao, Donald J Patterson, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, 2007.

[Niepert and Domingos, 2014] Mathias Niepert and Pedro Domingos. Exchangeable variable models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 271–279, 2014.

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

[Nitti *et al.*, 2013] 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.

[Pescini *et al.*, 2006] Dario Pescini, Daniela Besozzi, Giancarlo Mauri, and Claudio Zandron. Dynamical probabilistic P systems. *International Journal of Foundations of Computer Science*, 17(01):183–204, 2006.

[Pfeffer *et al.*, 2009] Avi Pfeffer, Subrata Das, David Lawless, and Brenda Ng. Factored reasoning for monitoring dynamic team and goal formation. *Information Fusion*, 10(1):99–106, 2009.

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

[Poon and Domingos, 2011] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference On*, pages 689–690. IEEE, 2011.

[Sanner and Boutilier, 2009] S. Sanner and C. Boutilier. Practical solution techniques for first-order MDPs. *Artificial Intelligence*, 173(5-6):748–788, April 2009.

[Schröder *et al.*, 2017] 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.

[Shirazi and Amir, 2011] Afsaneh Shirazi and Eyal Amir. First-order logical filtering. *Artificial Intelligence*, 175(1):193–219, 2011.

[Thon *et al.*, 2011] I. Thon, N. Landwehr, and L. De Raedt. Stochastic relational processes: Efficient inference and applications. *Machine Learning*, 82(2):239–272, February 2011.