Reasoning with Uncertainties Over Existence Of Objects

Ngo Anh Vien and **Marc Toussaint**

Institute for Parallel and Distributed Systems, University of Stuttgart, 70569 Germany
{vien.ngo,marc.toussaint}@ipvs.uni-stuttgart.de

Abstract

In this paper we consider planning problems in relational Markov processes where objects may "appear" or "disappear", perhaps depending on previous actions or properties of other objects. For instance, problems which require to explicitly generate or discover objects fall into this category. In our formulation this requires to explicitly represent the uncertainty over the number of objects (dimensions or factors) in a dynamic Bayesian networks (DBN). Many formalisms (also existing ones) are conceivable to formulate such problems. We aim at a formulation that facilitates inference and planning. Based on a specific formulation we investigate two inference methods—rejection sampling and reversible-jump MCMC—to compute a posterior over the process conditioned on the first and last time slice (start and goal state). We will discuss properties, efficiency, and appropriateness of each one.

Introduction

In many interesting real world robotics problems the robot is faced with uncertainty over whether an object exists or not. In the following we give some motivating example tasks, where the interesting aspect is that they require explicit reasoning about the uncertainty over existence and properties of objects:

– We need to hit a nail into a wall. We do not know which objects there might exist in the room, hidden in drawers or behind doors or other objects, and whether they are *hammer-like* (stone, metal bar, huge screw driver, . . .). We would have to explicitly reason about what is the probability that a hidden object exists in this-or-that drawer, what is the probability that it is hammer-like, and eventually which actions are suitable to reduce these uncertainties to solve the problem.

– We want to find as many objects as possible and pile them. For this we need to consider probabilities of finding objects when opening drawers, whether they might be "pilable", and which actions will lead to the discovery of suitable objects.

– We need to carry a box that is closed. If the box contains a too heavy object we might fail. We will have to reason about which objects might be contained in the box, what their weight is and which actions will reduce uncertainty and avoid failure.

Why is this hard, interesting, and non-trivial? First of all, the dimensionality of the state space depends on the number of objects. Therefore the agent has uncertainty about the dimensionality of the state space. Relational or factored MDPs can be thought of as Dynamic Bayes Networks (DBN). In the relational case, each property or relation of objects is being represented by a random variable in each time slice (Lang and Toussaint 2009). If we have uncertainty over existence of objects, we actually have uncertainty over which random variables exist in the DBN. Doing inference in such a DBN is an interesting matter. There are some formalism covering uncertainty over *state dimension* or *number of objects*. Infinite models such as infinite HMM (iHMM) (Beal, Ghahramani, and Rasmussen 2001), infinite factored HMM (ifHMM)(Gael, Teh, and Ghahramani 2008), infinite POMDP (Doshi-Velez 2009), and infinite DBN (Doshi et al. 2011) use Dirichlet Processes (DP) (Teh et al. 2004) and Indian Buffet Processes (IBP) (Griffiths and Ghahramani 2011) to define a prior over the cardinality of latent variables. Is this the same as having uncertainty over the existence of (latent) variables? In principle it is related when collapsing all variables to one big variable the uncertainty is simply over its cardinality. The previous work uses an IBP prior over uncertainty of cardinality of latent variables, but not number of objects or random variables and their relation. Moreover, the possibility of an object's appearance at one time slice depends only on objects and actions in the previous time slice (Markov property) not the whole time slices like in IBP, and they are not exchangeble like the assumption of IBP. This clearly yields a huge difference. BLOG (Bayesian Logic) (Milch et al. 2005), its nonparametric extension (Carbonetto et al. 2005) and formalisms like that dig deep into logic to describe probability distributions over worlds where the number of objects is unknown. Similarliy, the Church language (Goodman et al. 2008), a Lisp-like functional programming language, describes generative models as stochastic functions (e.g. a Bayes net with arbitrary symbolic nodes and arrows). But we do not see the possibility that this could ever be efficient for inference in planning. We would hope to find something simpler; a more direct extension of Markovian models.

Therefore, we think that the problem of reasoning with uncertainties over existence of objects is a fundamental type of uncertainty in real worlds that is not really nicely represented in practical models nowadays. And that is a huge potential for research. In this paper, we propose a formulation such that inference can be done efficiently. We introduce the target distribution which is over the unknown quantity of objects which is considered as a random variable. This results in a trans-dimensional target distribution. Then, we can carry out Bayesian inference for the problem of the unknown quantity of objects using reversible-jump MCMC. For experiment, we evaluate this proposed method on basic model problems. These model problems do not yet include action random variables – in this preliminary work we wanted to consider minimalistic scenarios that capture the core challenge of inference in processes where object may appear or disappear depending on random variables in the previous time slice. In these scenarios we condition the first and final time slice (start and goal state), as is standard in planning as inference approaches. The results show that we are successful to compute correct posteriors also over the (dis-) appearance of objects. We compare rejection sampling with a reversible-jump MCMC method that we designed specifically to address the problem of uncertainty over existence of objects.

In the motivating examples above we mentioned the *discovery* of objects. It is in fact an interesting discussion whether such discovery should be modelled as the *creation* of a new object—as we do here—rather than an object that always existed. We dig into this discussion in the last section. The foremost purpose of this preliminary work is to incite a discussion among the AI community about which methods will be most promising to address the problem.

Related Work

The work by Srivastava et. al. (Srivastava, Immerman, and Zilberstein 2009) uses abstraction for state aggregation to represent unknown object quantities. However, this is a deterministic planner which can not deal with inference and uncertainty.

The PHD filter (a.k.a. finite random sets) (Mahler 2007) is a method which allows the on-line estimation of the multi-object state using a sequence of noisy observation. It considers the tracked objects at each time slice as a random set. The multi-target posterior belief density function is computed using finite set statistics (FISST) (Goodman, Mahler, and Nguyen 1997). There are a number of algorithms used to implement FISST such as Sequential Monte Carlo (Sidenbladh 2003; B.-N., S., and A. 2003), Gaussian mixture model (B.-N and Ma 2005). The methods have been used extensively for tracking (Mahler 2000; 2002). These FISST methods have not yet been used to address planning. As these methods have specialized to address tracking problems with full observations we do not see a simple extension for planning as inference in our scenarios.

Finally, birth and death processes (Latouche and Ramaswami 1987) have been considered in the context of continuous-time Markov process over the number of individuals. However, these methods do not consider that individuals may have properties or relations as in relational MDPs.

Problem Formulation

Before defining a concrete model more formally, we first discuss examples. In general we will consider a factored Markov process on n_t random variables (RVs) $s_{t,i}$, for $i = 1, \cdots, n_t$ in time slice t^{th}. Each $s_{t,i}$ is discrete. We will need to define transition probabilities $p(n_t, s_t | n_{t-1}, s_{t-1})$ that also define the distribution over the number of random variables in time slice t depending on the state at time $t - 1$. In this paper we will usually condition on the number of RVs and their values at the the start and final time slice and consider the problem of computing the posterior distribution over number of RVs and their values in all intermediate time slices.

Examples

As a starting point for research we considered basic model problems which are not yet concerned with object manipulation but capture the basic problems involved when reasoning with uncertainty over number of RVs in Markov processes. We call these model problems "bacteria" problems, as they concern a set of objects, the size of which may change depending on whether they split, die, or change their properties. We assume the following three levels of difficulty of the bacteria model problem.

Problem 1: A bacteria can split, die, or stay at next time slice with a given probability: p, d, and $(1 - p - d)$ respectively. In this variant, bacteria have no properties. The state is fully captured by n_t in each time slice.

Problem 2: Bacteria have a single discrete property: color $\in \{1, 2, 3\}$. A bacteria's color may change at each time step. The split and die probabilities depend on the color value:

- Each object splits if $color = 3$ with probability of p into two new objects,

- And dies if $color = 1$ with probability of d.

- The color changes to a uniform other color with probability 0.3 and stays unchanged with probability 0.7.

To exemplify Problem 2, in Fig. 1, we generate 6 feasible graphs using rejection sampling with setting of $T = 5, n_1 = 5, n_5 = 3$, the colors of these bacteria are randomly initialized, $p = 0.1, d = 0.4$.

Problem 3: This problem makes the same assumptions as problem 2, but in addition objects can have relations. We consider a binary valued relation *pair*. The split probability depends on the value of this relation: two bacteria can only split if they are *paired*. Each time slice now includes additional $O(n^2)$ RVs to represent the value p_{ij} of the pair relation. The *pair* transition probabilities are

$$P(p_{ij}^t = 1 | p_{ij}^{t-1} = 0, c_i = 3, c_j = 3) = \delta \,, \qquad (1)$$

$$P(p_{ij}^t = 0 | p_{ij}^{t-1} = 0, c_i = 3, c_j = 3) = 1 - \delta \,, \quad (2)$$

$$P(p_{ij}^t = 1 | p_{ij}^{t-1} = 1) = 1 \,. \qquad (3)$$

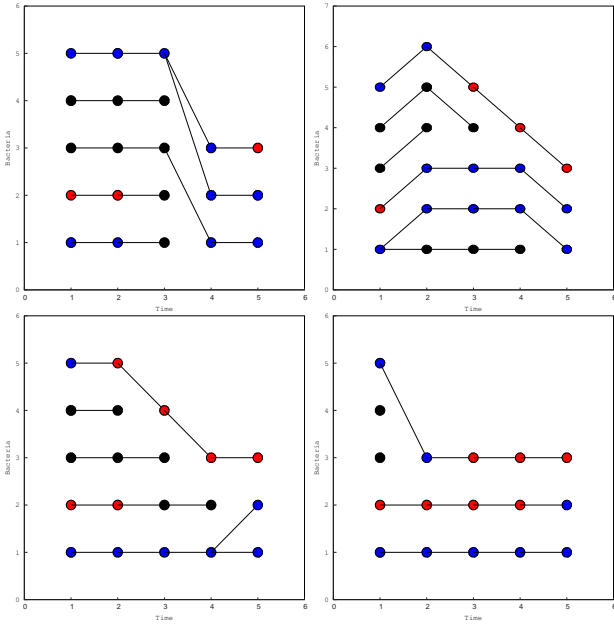Figure 1: Example of problem 2. Colors $\{1,2,3\}$ = {Black, Red, Blue}

Split operator above changes to *spawn*. If $p_{ij}^{t-1} = 1$, there is a new node spawn with probability p.

Using BLOG framework (Milch et al. 2005), the generative process of Problem 3 can be described in first-order logic in Table 1.

However, BLOG is not efficient for doing inference planning in sequential decision problems. In Table 2, we use a NID framework of Pasula et. al. (Pasula, Zettlemoyer, and Kaelbling 2007) to represent stochastic world dynamics having uncertainty over existence of objects.

Formulation

The examples above describe basic model processes. We not formalize the problem of inference more concretely. We assume we are given the transition model Θ and evidences at the first and last time slice $\{1, T\}$. For instance, the transition model Θ in the above problems is given by $\{p, d, \delta\}$ (probability of spliting, dying, and pairing). The task is to infer the posterior over number of RVs and their values at time slice $\{2, \cdots, T-1\}$. More specifically, we encode each time slice t with information $I_t = \{n_t, \mathbf{L}_t, \mathbf{V}_t\}$, where n_t is the number of nodes, $\mathbf{L}_t = \{l_1, \cdots, l_{n_t}\}$ is the set of links for each node (each link l_i is a set of the nodes at next time slice connected to node i), and $\mathbf{V}_t = \{v_1, \cdots, v_{n_t}\}$ is the set of the nodes' values. We denote a dynamic graph as $G = \{\mathbf{N}, \mathbf{L}, \mathbf{V}\}$ which has a set of node $\mathbf{N} = \{n_1, \cdots, n_T\}$, $\mathbf{L} = \{\mathbf{L}_1, \cdots, \mathbf{L}_{T-1}\}$, and $\mathbf{V} = \{\mathbf{V}_1, \cdots, \mathbf{V}_{T-1}\}$. The general problem is to estimate $p(G|\Theta, I_1, I_T)$. Note that in the model problems above, the link structure really becomes part of the random variables: which random variable is coupled to which in the previous time slice depends on which random operators have occured during the transition. There-fore we included $\mathbf{L}$ in the formulation.

We are interested in the distribution or the expected number of nodes and their values at each time slice: $p(n_t, \mathbf{V}_t|\Theta, I_1, I_T)$, or $E(n_t|\Theta, I_1, I_T)$. This is given at any time slice t as

$$p(n_t, \mathbf{V}_t|\Theta, I_1, I_T) = \sum_{\mathbf{N}_{-t}} \sum_{\mathbf{V}_{-t}} \sum_{\mathbf{L}} p(\mathbf{N}, \mathbf{L}, \mathbf{V}|\Theta, I_1, I_T) \tag{4}$$

and

$$p(n_t|\Theta, I_1, I_T) = \sum_{\mathbf{N}_{-t}} \sum_{\mathbf{V}} \sum_{\mathbf{L}} p(\mathbf{N}, \mathbf{L}, \mathbf{V}|\Theta, I_1, I_T) \tag{5}$$

and

$$E(n_t|\Theta, I_1, I_T) = \sum_{n_t=0}^{\infty} n_t p(n_t|\Theta, I_1, I_T)$$

$$= \sum_{n_t=0}^{\infty} n_t \sum_{\mathbf{N}_{-t}} \sum_{\mathbf{V}} \sum_{\mathbf{L}} p(\mathbf{N}, \mathbf{V}, \mathbf{L}|\Theta, I_1, I_T) \tag{6}$$

where the probability $p(\mathbf{N}, \mathbf{L}, \mathbf{V}|\Theta, I_1, I_T) = p(G|\Theta, I_1, I_T)$ is the joint probability which can be computed tractably. However, computing the above quantity over possible values of n_t (its value's domain is unbounded), $\mathbf{L}_t$ (depending on n_t), and $\mathbf{V}_t$ (also depending on n_t) is intractable. In order to evaluate this quantity we propose to use sampling approaches. If we can sample M graphs $\{G_i\}_{i=1}^{M}$ from the distribution $p(G|\Theta, I_1, I_T)$ then (4) can be estimated as

$$p(n_t, \mathbf{V}_t|\Theta, I_1, I_T) = \frac{1}{M} \sum_{i=1}^{M} \mathbf{1}_{(n_t, \mathbf{V}_t)}(G_i) \tag{7}$$

where $\mathbf{1}_{(n_t, \mathbf{V}_t)}(G_i)$ is an indicator function to verify if the number of nodes and their values of graph G_i are n_t and $\mathbf{V}_t$, respectively. The target distribution $p(G|\Theta, I_1, I_T)$ can be evaluated easily as a product of factors $p(I_t|Pa(I_t))$

$$p(G|\Theta, I_1, I_T) = \prod_{t=2}^{T} \prod_{i=1}^{n_t} p(I_t^i|Pa(I_t^i)) \tag{8}$$

```
type Bacteria; type Color
#{Bacteria}:
    ∼ NumBacteria()
#{Color}:
    ∼ NumColor()
Color(b):
    ∼ UniformColor()
Paired(b₁, b₂, t):
    then ∼ PairedTransition(δ)
State(b, t):
    If IsColor(b, 1) ∧ DieDistribution(d)
        then = null
    Else ∼ ColorTransition(b)
State(b₁, b₂, t):
    If Paired(b₁, b₂, t)
        then = InitNewBacteria(b, t + 1)
    Else = null
```

Table 1: BLOG model for the bacteria evolution problem.

(***Die* operator**)

$Evolve(X) : \{IsBacteria(X)\}$

      $Color(X, 1)$

$$\rightarrow \begin{cases} d & : Set(X, null) \\ 1-d & : \text{no change} \end{cases}$$

(***stay still* operator**)

$Evolve(X) : \{IsBacteria(X)\}$

      $Color(X, 2) \lor Color(X, 3)$

$$\rightarrow \{ \ 1 \quad : \text{no change}$$

(**Color transition**)

$Evolve(X) : \{IsColor(X)\}$

$$\rightarrow \begin{cases} 0.8 & : SetColor(X, Color(X)) \\ 0.1 & : SetColor(X, \text{other color 1}) \\ 0.1 & : SetColor(X, \text{other color 2}) \end{cases}$$

(**Establish *paired* relation**)

$Evolve(X) : \{IsPair(X)\}$

      $X = IsPairOf(b_1, b_2), Color(b_1, 3),$

      $Color(b_2, 3), Paired(X) = 0$

$$\rightarrow \begin{cases} \delta & : SetPaired(X, 1) \\ 1-\delta & : SetPaired(X, 0) \end{cases}$$

(**Establish *paired* relation**)

$Evolve(X) : \{IsPair(X)\}$

      $X = IsPairOf(b_1, b_2),$

      $\neg Color(b_1, 3) \lor \neg Color(b_2, 3), Paired(X) = 0$

$$\rightarrow \{ \ 1.0 \quad : SetPaired(X, 0)$$

(**Establish *paired* relation**)

$Evolve(X) : \{IsPair(X)\}$

      $X = IsPairOf(b_1, b_2), Paired(X) = 1$

$$\rightarrow \{ \ 1.0 \quad : SetPaired(X, 1)$$

(***paired* transition**)

$Evolve(X) : \{IsPair(X)\}$

      $X = IsPairOf(b_1, b_2), Paired(X) = 1$

$$\rightarrow \begin{cases} p & : InitNewBacteria(b_3) \\ 1-p & : \text{no change} \end{cases}$$

Table 2: NID model for the bacteria evolution problem.

Proposed Approach

In this section, we propose two solutions to sample G from its target distribution: *rejection sampling*, and *reversible-jump MCMC*. The first one is straightforward, then served as a base-line solution.

Rejection Sampling

We start with information I_1 at the first time slice, then use the generative model Θ to evole until the terminal time slice T. If the information at T matches with I_T, the sample is accepted, otherwise rejected. Next sampling round repeats.

Reversible-Jump MCMC

Rejection sampling is notorious for its inefficiency in acceptance rate. This problem is even more severe when T is a long horizon, each node can take a lot of possible values and relations, then the acceptance rate is closer to zero. In this section, we propose an alternative Bayesian method. This method introduces a trans-dimensional target distribution and proposes to use one trans-dimensional MCMC algorithm which is Reversible-Jump MCMC.

Reversible-Jump MCMC (RJMCMC) was proposed by Green in 1995 (Green 1995) which is an extension to the Metropolis-Hasting algorithm (Hastings 1970). Using RJMCMC, we can directly compute the posterior $p(\mathbf{N}, \mathbf{L}, \mathbf{V}|\Theta, I_1, I_T)$ via reversible jumps through spaces of graphs having the different number of nodes, links, and values. We initialize the RJMCMC algorithm with an arbitrary graph $G^0 = \{\mathbf{N}^0, \mathbf{L}^0, \mathbf{V}^0\}$. For each iteration k, we choose a move type m from a set of reversible moves $\mathcal{M}$. We use three types of move: *birth* (a new node at a time slice is born), *death* (a node at a time slice is dead), and *update* (links at a time slice are updated). Each move is applied on a randomly selected time slice t. Assume that a new proposal graph G^* is generated from this move. It is accepted according to an acceptance probability which is computed slightly similar to the classical Metropolis-Hastings algorithm's. A generic RJMCMC for the growing DBN problem is as in Algorithm 1.

Algorithm 1 Reversible-jump MCMC.

Require: Initialize $(\mathbf{N}^0, \mathbf{L}^0, \mathbf{V}^0)$.
  for $k = 0 \rightarrow k_{max}$ **do**
    Sample $u \sim \mathcal{U}(0, 1)$.
    if $u < b_k$ **then**
      Do *birth* move.
      Accept the move with probability in Eq.10.
    else if $u < b_k + d_k$ **then**
      Do *death* move.
      Accept the move with probability in Eq.11.
    else
      Do *update* move.
    end if
  end for

We use two reversible moves (*birth* and *death*) and update moves. For each time slice, we assume the number of node n_t has a prior $p(n_t)$ according to a truncated-Poission distribution. Similar to the setting in (Green 1995), we set

$$\begin{aligned} b_k &= c_{max} \cdot \min\{1, p(n_t + 1)/p(n_t)\} \\ d_k &= c_{max} \cdot \min\{1, p(n_t - 1)/p(n_t)\} \end{aligned} \quad (9)$$

where the constant c_{max} is set such that $b_k + d_k \leq 0.9$, for all $k = 0, \cdots, k_{max}$.

With a probability of b_k, we propose a *birth* move at a randomly selected time slice at which there is a new node n_{ti*} added according to the probability $q_{birth}(n_{ti*}|\Theta, I_{t-1}, I_t)$. For the example in previous section, we enumerate all possibilities that there are parents who can have more children,

then choose randomly one possibility. It has a reversible distribution $q_{death}(n_{ti}|\Theta, I_{t-1}, I_t)$ that enumerates all possible nodes which has no children can be removed. The newly created node's parent (links) and values are initialized according to $p(n_{ti^*}|Pa(n_{ti^*}), I_{t-1}, I_t)$ and $p(v_{ti^*}|Pa(n_{ti^*}))$. This new graph is accepted with a probability $A_{birth}(G^*, G_k) = \min\{1, \alpha_{birth}\}$, where

$$\alpha_{birth} = \frac{p(G^*|\Theta, I_1, I_T)}{p(G^i|\Theta, I_1, I_T)} \cdot \frac{d_{k+1}}{b_k} \cdot \frac{q_{death}(n_{ti}|\Theta, I_{t-1}^*, I_t^*)}{q_{birth}(n_{ti^*}|\Theta, I_{t-1}, I_t)}$$
(10)

Similarly, with a probability of d_k, we propose a *death* move at a randomly selected time slice at which we delete a node $n_{ti} \sim q_{death}(n_{ti}|\Theta, I_{t-1}, I_t)$ (enumerate all possible nodes which has no children can be removed). This new graph is accepted with a probability $A_{death}(G^*, G^k) = \min\{1, \alpha_{death}\}$, where

$$\alpha_{death} = \frac{p(G^*|\Theta, I_1, I_T)}{p(G_i|\Theta, I_1, I_T)} \cdot \frac{b_{k+1}}{d_k} \cdot \frac{q_{birth}(n_{ti^*}|\Theta, I_{t-1}^*, I_t^*)}{q_{death}(n_{ti}|\Theta, I_{t-1}, I_t)}$$
(11)

And with probabilty $u_k = 1 - b_k - d_k$, we propose a standard Metropolis-Hastings moves to update links at a randomly selected time slice. The Markov chain $\{\mathbf{N}^k, \mathbf{L}^k, \mathbf{N}^k\}_{k=0}^{k_{max}}$ generated as above is proved to be irreducible and aperiodic. Then it asymptotically converges to samples of the true posterior distribution $p(\mathbf{N}, \mathbf{L}, \mathbf{V}|\Theta, I_1, I_T)$.

Experiments

We use the model problems previously introduced for comparisons of the two proposed solutions. To speed up the burn-in stage and avoid local optima, we initialize the RJMCMC algorithm with a sample from the rejection sampling algorithm. The error bars for rejection sampling are the standard deviation of the mean itself from 10000 iterations, for RJMCMC they are the standard deviation of the mean itself averaged from ten 10000-iteration runs (including 5000 iterations for burn-in). Through experiments we expect to answer two questions: First, are the samples of our RJMCMC proposal from the true target distribution? And, is the RJMCMC proposal reversible?

Problem 1 is tested with setting $T = 5, n_1 = 4, n_5 = 2$, $p = 0.3, d = 0.5$. The comparison is shown in Fig. 2. The rejection sampling has an acceptance rate of $1600/10000$.

Problem 2 is tested setting $T = 5, n_1 = 5, n_5 = 3$, the colors of these bacteria are randomly initialized, $p = 0.1, d = 0.4$. The comparisons of the distribution over the number of nodes and of the distribution over the number of values is shown in Fig. 3. The rejection sampling has an acceptance rate of $83/10000$.

Problem 3 is tested setting $T = 5, n_1 = 2, n_2 = 4, p = 0.5, d = 0.2, \delta = 0.5$. The comparisons of the distribution over the number of nodes and of the distribution over the number of values is shown in Fig. 4. The rejection sampling has an acceptance rate of $108/10000$.

For all three problems the mean estimators for the number of objects match within the standard deviations between rejection sampling and RJMCMC. Also, for Problem 2 and 3
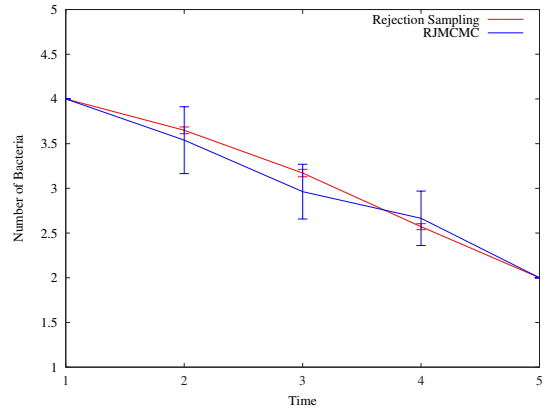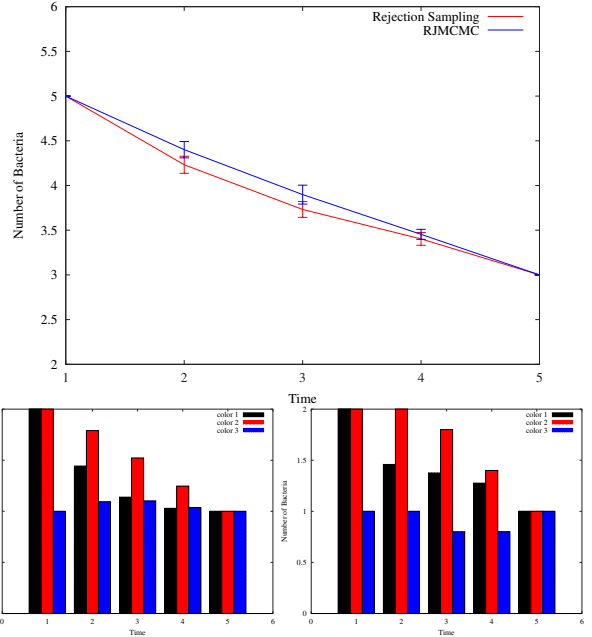

Figure 2: Problem 1: Number of Bacteria vs Time

Figure 3: Problem 2: Number of bacteria vs time and color value histogram (left: rejection sampling, right: RJMCMC).

the color value histograms match—confirming that the samples of our RJMCMC proposal are correcly from the true target distribution. This ascertains that the RJMCMC proposal successfully jumps through spaces of feasible graphs, therefore it is reversible. The color value histograms for Problem 2 and 3 interestingly reflect the intrinsics of the transition probabilities. E.g. in Problem 3, the absence of color 3 at $t = 1$ leads to the need to generate many 3's at $t = 2$ to ensure the necessary growth to reach the final state.

Discussion

We have studied the problem of inference in processes where objects can be created or destroyed, implying the need to reason directly about the uncertainties over existence of ob-
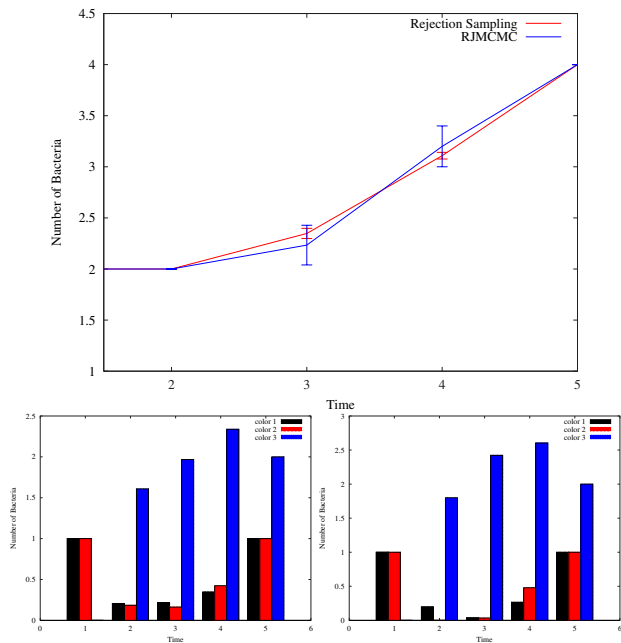
Figure 4: Problem 3: Number of bacteria vs time and color value histogram (left: rejection sampling, right: RJMCMC).

jects. In this preliminary work we proposed a basic formulation of a trans-dimensional distribution over such processes and tested rejection sampling as well as a specific reversible-jump MCMC method to compute the desired posteriors. The results on basic model problems show that RJMCMC is promising and successful in computing the correct posteriors.

Our original motivation for this work was to address robotic scenarios that involve the (active) discovery of objects by explicitly reasoning about the chances of discovery. Literally taken, discovery and creation are very different things: a discovered object has "always" existed. Translated to a DBN framework, modelling discovery would, taken literally, require to have RVs for all objects that might exist for all times. As these might potentially be infinitely many this naturally leads to ifHMM or IBP type models. However, is this really necessary? With this work we want to propose that modelling discovery as the creation of a new object (which has no history in the DBN) might lead to simpler and more feasible models. Future work should elaborate on the formal differences or equivalences. Most important, our hope is that future application of such approaches in real robotic scenarios will evaluate the practiability of these alternatives (including alternative inference methods) and lead to efficient planning methods for active discovery problems.

Ackowledgement

References

B.-N, V., and Ma, W.-K. 2005. A closed-form solution for the probability hypothesis density filter. In *Proc. Int'l Conf. on Information Fusion*, volume 2, 25–28.

B.-N, V.; S., S.; and A., D. 2003. Sequential monte carlo implementation of the PHD filter for multi-target tracking. In *Proc. Int'l Conf. on Information Fusion*, 792799.

Beal, M. J.; Ghahramani, Z.; and Rasmussen, C. E. 2001. The infinite hidden Markov model. In *NIPS*, 577–584.

Carbonetto, P.; Kisynski, J.; de Freitas, N.; and Poole, D. 2005. Nonparametric Bayesian logic. In *UAI*, 85–93.

Doshi, F.; Wingate, D.; Tenenbaum, J. B.; and Roy, N. 2011. Infinite dynamic Bayesian networks. In *ICML*, 913–920.

Doshi-Velez, F. 2009. The infinite partially observable Markov decision process. In *NIPS*, 477–485.

Gael, J. V.; Teh, Y. W.; and Ghahramani, Z. 2008. The infinite factorial hidden Markov model. In *NIPS*, 1697–1704.

Goodman, N. D.; Mansinghka, V. K.; Roy, D. M.; Bonawitz, K.; and Tenenbaum, J. B. 2008. Church: a language for generative models. In *UAI*, 220–229.

Goodman, I.; Mahler, R. P. S.; and Nguyen, H. T. 1997. *Mathematics of Data Fusion*. Springer.

Green, P. J. 1995. Reversible jump Markov chain monte carlo computation and Bayesian model determination. *Biometrika* 82(4):711–732.

Griffiths, T. L., and Ghahramani, Z. 2011. The indian buffet process: An introduction and review. *Journal of Machine Learning Research* 12:1185–1224.

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109.

Lang, T., and Toussaint, M. 2009. Approximate inference for planning in stochastic relational worlds. In *ICML*, 74.

Latouche, G., and Ramaswami, V. 1987. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Society for Industrial and Applied Mathematics.

Mahler, R. P. S. 2000. Technical report, Lockheed Martin Technical Monograph. An Introduction to Multisource-Multitarget Statistics and its Applications.

Mahler, R. P. S. 2002. An extended first-order Bayes filter for force aggregation. In *SPIE Conference on Signal and Data Processing of Small Targets*, volume 4729.

Mahler, R. P. S. 2007. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems* 43(4):1523–1543.

Milch, B.; Marthi, B.; Russell, S. J.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2005. Blog: Probabilistic models with unknown objects. In *IJCAI*, 1352–1359.

Pasula, H. M.; Zettlemoyer, L. S.; and Kaelbling, L. P. 2007. Learning symbolic models of stochastic domains. *J. Artif. Intell. Res. (JAIR)* 29:309–352.

Sidenbladh, H. 2003. Multi-target particle filtering for the probability hypothesis density. In *Proc. Int'l Conf. on Information Fusion*, 279–290.

Srivastava, S.; Immerman, N.; and Zilberstein, S. 2009. Abstract planning with unknown object quantities and properties. In *SARA*.

Teh, Y. W.; Jordan, M. I.; Beal, M. J.; and Blei, D. M. 2004. Sharing clusters among related groups: Hierarchical dirichlet processes. In *NIPS*.