Relational Markov Decision Processes: Promise and Prospects

Saket Joshi
Cycorp Inc., Austin, TX 78731

Roni Khardon
Tufts University
Medford, MA 02155

Prasad Tadepalli
Oregon State University
Corvallis, OR 97331

Alan Fern
Oregon State University
Corvallis, OR 97331

Aswin Raghavan
Oregon State University
Corvallis, OR 97331

Relational Markov Decision Processes (RMDPs) offer an elegant formalism that combines probabilistic and relational knowledge representations with the decision-theoretic notions of action and utility. In this paper we motivate RMDPs to address a variety of problems in AI, including open world planning, transfer learning, and relational inference. We describe a symbolic dynamic programming approach via the 'template method' which addresses the problem of reasoning about exogenous events. We end with a discussion of the challenges involved and some promising future research directions.

Introduction

The past decade has seen significant advances in expressive probabilistic and relational knowledge representations and inference on the one hand and effective decision-theoretic planning on the other. The formalism of Relational Markov Decision Processes (RMDPs) brings the two paradigms together by combining expressive knowledge representation with decision-theoretic notions of actions and utilities.

To motivate the need for such a rich formalism, consider the problem of building a general-purpose household robot. A necessary condition for such a robot is that it should be able to understand and do a variety of tasks, e.g., from opening the doors to making coffee. In other words, its vocabulary must include everyday objects such as cups and tables and the relationships between them, thus ruling out systems based on propositional languages (Boutilier *et al.* 2001). Importantly, it must be able to plan and execute a sequence of actions, and must respond rationally to exogenous events such as hearing a door bell when making coffee. Unlike the current planning systems that make the closed world assumption, the robot needs to be able plan in open worlds, where the set of objects in the world are variable and unknown, e.g., a cup falls on the floor and breaks into many pieces, or a neighbor brings a plate full of cookies (Talamadupula *et al.* 2010; Joshi *et al.* 2012). It should be able to reason about the consequences of actions, and make inferences about the goals and potential future actions of other agents based on observations (Lang and Toussaint 2010). Indeed, it is hopelessly inadequate to either divorce action from inference, or ignore the realities of open, relational, stochastic, and multi-agent worlds. The robot must be able to learn new knowledge and generalize and transfer to other related settings, which means that the knowledge must be represented in a general enough form (Proper and Tadepalli 2009). For example, while a cup may not be similar to a vase, a broken cup is very much similar to a broken vase and must be dealt with in similar ways.

In the current paper, we focus on one particular problem, namely planning in the presence of exogenous events. We are particularly motivated by applications to "service domains," such as taxi service, or inventory control, where the stochasticity arises mainly due to the exogenous service requests. The problem is to derive optimal plans that allow the agent to effectively respond to these requests in a timely manner over the long term. The main difficulty with the exogenous events is that they are not directly addressed by the standard MDP framework. They are instead treated as stochastic outcomes of the agent's actions, which is unnatural. Further, the propositional formulation of the MDP framework prevents us from taking advantage of the similarity of different objects in the way they generate exogenous events. We argue that a more natural modeling and analysis of exogenous events would lead to more efficient and practical algorithms.

Our approach to the problem of solving RMDPs with high exogeneity is based on Relational Symbolic Dynamic Programming (RSDP) which builds a symbolic representation of the optimal value function over relational states using generalized first order decision diagrams (GFODDs). Unlike the propositional approach which requires grounding of the dynamics to each domain size, the relational (first order logic) approach (Boutilier *et al.* 2001) generalizes to all sizes. To our knowledge, the only work to have approached first order treatment of exogenous events is (Sanner and Boutilier 2009; Sanner 2008). While this work is very ambitious in that it attempted to solve a very general class of problems using approximate policy iteration and heuristic simplification, it is not clear when the particular combination of these ideas is applicable.

Instead of trying to solve general RMDPs with arbitrarily complicated exogeneities, in this paper, we argue for focusing on a common but well-constrained kind of exogeneity, which is centered around individual objects such as customers in the aforementioned service domains. It seems reasonable to assume that the customers behave independently of each other and impact the service agent only through their collective demands on its resources. Hence, in general, we model the state transition in each step as first taking the agent's action, and then following a sequence of "object-centered exogenous events" in any order. While the exact solution to this problem is still likely to be complex, we argue that a symbolic lower bound approximation can be computed, and that this provides a useful solution. In particular, we develop and evaluate a new "template-based" algorithm that substitutes and reasons about a generic exogenous event, and uses it to calculate a symbolic approximation of the value function. We show that the approximation is monotonic, which provides a guarantee that the resulting greedy policy will at least achieve the computed value. We have verified in two inventory control domains that our new algorithm scales better than propositional approaches, and produces a size-independent solution of high quality.

Relational Symbolic Dynamic Programming

We consider RMDPs that consist of states S which are finite logical interpretations and action templates $A(x)$ where x can be instantiated by objects yielding ground actions. Different outcomes of a stochastic action $A(x)$ are represented as deterministic action variants $A_j(x)$ which are chosen with $prob(A_j(x)|A(x))$. We assume that the next state and the rewards are deterministic functions of the current state and the deterministic action variant.

Relational Expressions and GFODDs. The key idea of symbolic dynamic programming is to directly manipulate the expressions that represent reward and transition functions to derive symbolic value functions over interpretations. The RSDP algorithm of (Joshi *et al.* 2011) generalizes (Boutilier *et al.* 2001) and employs GFODD representations to do this.

For pedagogical reasons, a GFODD can be treated as an expression $f(x)$, similar to an open formula in first order logic, which can be evaluated in interpretation I once we substitute the variables x with concrete objects in I. A closed expression ($\text{aggregate}_x f(x)$) aggregates the value of $f(x)$ over all possible substitutions of x to objects in I. In this paper we focus on average and max aggregation. E.g., in an inventory control (IC) domain we might use the expression: "$\max_t \text{avg}_s$ (if $\neg empty(s)$ then 1, else if $tin(t, s)$ then 0.1, else 0)". This awards a 1 for any non-empty shop and at most one shop is awarded a 0.1 if there is a truck at that shop.

Relational Value Iteration. As input, the algorithm gets closed GFODDs V_n, R, and open GFODDs for the probabilistic choice of actions $prob(A_j(x)|A(x))$ and for the dynamics of deterministic action variants. It then implements the following symbolic value iteration operator, represented as $V_{n+1} = RSDP^1(V_n) = \max_A \max_x R \oplus \gamma \oplus_j (prob(A_j(x)) \otimes Regr(V_n, A_j(x)))$

Here, $\oplus$, $\otimes$ and $\max$ represent point-wise sum, multiplication, and max of functions, γ is the discount factor, and $Regr$ is a regression of a function over a deterministic action variant. Importantly, the variables in different parts of $\oplus$ are standardized apart, i.e., $max_x f(x) \oplus max_x g(x) = max_x max_y f(x) \oplus g(y)$, which is crucial for correctness.

Handling Exogenous Events. We next show how to extend RSDP to handle exogenous events, which we treat as nature's actions, and assume to be "object-centered." In particular, we assume **A1:** for every object i in the domain we have action $E(i)$ that acts on object i and the conditions and effects of $\{E(i)\}$ are such that they are mutually non-interfering. In other words, given any state s, all the actions $\{E(i)\}$ are applied simultaneously, and this is equivalent to their sequential application in any order. For our analysis we make three further modeling assumptions. **A2**: each exogenous action $E(i)$ only effects unary predicates of object i which we label "special"; **A3:** the special unary predicates do not occur as preconditions of agent actions; and **A4:** the reward function is a closed expression of the form $\max_x \text{avg}_y R(x, y)$, and any special predicates in $R(x, y)$ are only applied to y.

We use the same GFODD-based representation to capture the dynamics of exogenous actions $E(i)$ as we do for the agent actions. Each potential exogenous action may have several action variants, minimally, a success variant $E_{succ}(i)$ which might mean that the corresponding event has occurred and a failure variant $E_{fail}(i)$ which means that it has not occurred.

The Template Method. Naive approaches to extend RSDP will use the explicitly ground $E(i)$ in calculating the value function but this does not yield an abstract solution. On the other hand, exact solutions require counting formulas and are very complex (Sanner 2008; Sanner and Boutilier 2009). In contrast, our *template method* provides a completely abstract approximate RSDP solution for the exogenous event model.

The template method first runs the following 4 steps, denoted $RSDP^2(V_n)$, and then applies $RSDP^1$ to the result. The final output of our approximate Bellman backup, is $V_{n+1} = RSDP^1(RSDP^2(V_n))$.

1. **Skolemization:** Let a be a Skolem constant not in V_n. Partially ground V to get $V = \max_x V(x, a)$
2. **Regression:** The function V is regressed over every deterministic variant $E_j(a)$ of the exogenous action centered at a to produce $Regr(V, E_j(a))$.
3. **Add Action Variants:** The value function is updated $V = \oplus_j(prob(E_j(a)) \otimes Regr(V, E_j(a)))$. Importantly, in contrast with the $RSDP^1$ step, here we do not standardize apart the functions when performing $\oplus_j$. This leads to a pessimistic approximation of the value function, as it could overly constrain the action choices.
4. **Reintroduce Avg Aggregation:** An inductive argument based on our assumptions implies that the form of V is guaranteed to be $\max_x W(x, a)$. Return $V = \max_x \text{avg}_y W(x, y)$.

Thus, the algorithm grounds V using a generic object for exogenous actions, it then performs regression for a single generic exogenous action, and then reintroduces the aggregation. We make the following performance guarantee.

Theorem 1 *Under assumptions* **A1, A2, A3, A4** *we have for all n:* $V_n \leq V_{n+1} \leq T[V_n] \leq V^*$*, where* $T[V]$ *is the* *true Bellman backup.*

The above theorem shows that our algorithm computes a monotonic lower bound of the true value function, ensuring that the value of the greedy policy wrt V_n is at least V_n. In other words, V_n provides an immediate certificate on the quality of the resulting greedy policy.

Without further logical simplification, the RSDP approaches including ours produce increasingly complex GFODDs that can easily overwhelm the system. We developed new evaluation and model-checking reduction algorithms for GFODDs that simplify diagrams and speed up the run time. These reductions employ a focus set of examples to check which parts of the GFODDs are exercised, and prune the edges which are not exercised by any of the examples.

Experimental Results. Our experiments in two simple versions of "inventory control" domain showed that our approach is efficient, and produced policies that are competitive with those found by the propositional approaches. More importantly, because they are independent of problem size, they scale more easily to large problem sizes.

The first version of the domain included shops with only two levels of inventories and the same rate of consumption. This version satisfied all assumptions (**A1** $\cdots$ **A4**). We observed that while the propositional systems could not handle more than 9 shops, our system was able to scale up to 20 shops with a policy which is statistically indistinguishable from the optimal policy. The second version had shops with 3 inventory levels and one of two possible rates of consumption. This version violated our assumption **A3**. Although our policy was statistically slightly inferior to the optimal in this case, our system was able to scale to 20 shops, while the propositional systems failed beyond 5 shops.

Conclusions and Future Work

We argued that RMDPs provide an elegant formalism that supports planning, reasoning, and action execution in a rich probabilistic relational language. We showed that at least in some cases, our GFODD-based RSDP algorithm is efficient and produces good policies which are size-independent. On the other hand, our algorithm is not exact and only guarantees a monotonic lower bound of the optimal result. Since the implementation also uses model-checking reductions to prune the GFODDs, the guarantees are further weakened and become essentially statistical.

A cautious lesson one can draw from this is that approximations are essential. Although many domains have simple dynamics which is compactly described, their value functions are not necessarily compact, even when they are symbolically expressed in elegant notation. The key questions are what knowledge representation best supports such algorithms and, when it fails, what to approximate and how. We believe that RSDP methods have much to tell us about when the value function is relatively compact, and when it is getting too complicated to represent exactly. It is also possible that while the value function for the whole RMDP is quite complex, it might contain several "sub-RMDPs" which may have more compact value functions. The area of hierarchical reinforcement learning is founded on this insight.

Another area which seems ripe for exploration is RSDP approaches for policy search. Indeed, GFODDs have been used to represent and learn policies directly, and it has been often argued that learning policies is simpler and better than learning value functions (Wang *et al.* 2008). It is yet unclear how to do this well, and whether and when policy-iteration and other policy search-based methods would yield superior results on problems of practical interest.

More recent successes in domains like Go suggest that it is important to include real-time search in our algorithmic tool-box (Silver *et al.* 2012). Approaches such as Monte-Carlo tree search completely ignore the structured representation of the problem and treat each search node as atomic. Most successful learning is problem-specific, and is driven by feature engineering rather than a principled approach beginning with the domain-dynamics. It seems that there is much room for novel algorithms that combine symbolic inference with search, sampling, and learning.

Finally, the relational representation provides an excellent opportunity for studying domain reformulation. Often different formulations of the same problem can lead to very different representations of the value functions and policies. Automating such reformulations might be a powerful way to further scalability, for example, via the induction of new predicate definitions.

Acknowledgments

This work was supported by NSF under grant numbers IIS-0964705 and IIS-0964457.

References

C. Boutilier, R. Reiter, and B. Price. Symbolic dynamic programming for first-order MDPs. In *IJCAI*, 690–700, 2001.

S. Joshi, K. Kersting, and R. Khardon. Decision theoretic planning with generalized first order decision diagrams. *AIJ*, 175:2198–2222, 2011.

S. Joshi, P. Schermerhorn, R. Khardon, and M. Scheutz. Abstract planning for reactive robots. In *ICRA*, 4379–4384, 2012.

T. Lang and M. Toussaint. Planning with noisy probabilistic relational rules. *JAIR*, 39:1–49, 2010.

S. Proper and P. Tadepalli. Transfer learning via relational templates. In *ILP*, 2009.

S. Sanner and C. Boutilier. Practical solution techniques for first-order MDPs. *AIJ*, 173:748–788, 2009.

S. Sanner. *First-order decision-theoretic planning in structured relational environments*. PhD thesis, University of Toronto, 2008.

D. Silver, R. S. Sutton, and M. Müller. Temporal-difference search in computer Go. *MLJ*, 87(2):183–219, 2012.

K. Talamadupula, J. Benton, S. Kambhampati, P. Schermerhorn, and M. Scheutz. Planning for human-robot teaming in open worlds. *TIST*, 1(2):14, 2010.

C. Wang, S. Joshi, and R. Khardon. First-Order decision diagrams for relational MDPs. *JAIR*, 31:431–472, 2008.