Gradient-Based Relational Reinforcement-Learning of Temporally Extended Policies

Charles Gretton
charles.gretton@nicta.com.au

Abstract

[1] We consider the problem of computing general policies for decision-theoretic planning problems with temporally extended rewards. We consider a gradient-based approach to relational reinforcement-learning (RRL) of policies for that setting. In particular, the learner optimises its behaviour by acting in a set of problems drawn from a target domain. Our approach is similar to *inductive policy selection* because the policies learnt are given in terms of relational control-rules. These rules are generated either (1) by reasoning from a first-order domain description, or (2) more or less arbitrarily according to a taxonomic concept language.

The cost of decision-theoretic planning in individual problems is substantial. State-of-the-art solution algorithms target either state-based (tabular) or factored propositional problem representations, thus they succumb to Bellman's curse of dimensionality – i.e. the complexity of computing the optimal policy for a problem instance can be exponential in the dimension of the problem (Littman, Goldsmith, & Mundhenk 1998). A research direction which has garnered significant attention recently is that of *generalisation in planning*. The idea is that the cost of planning with propositional representations can be mitigated by technologies that plan for a domain rather than for individual problems. These approaches yield *general policies* which can be executed in any problem state from the domain at hand. In practice general policies are expressed in first-order/relational formalisms. Proposals to date suggest general policies can be achieved by either (1) reasoning from the domain description (Boutilier, Reiter, & Price 2001; K. Kersting, M. V. Otterlo, & L. D. Raedt 2004; Sanner & Boutilier 2005; Karabaev & Skvortsova 2005; Wang, Joshi, & Khardon 2007), or (2) by developing planners that can *learn from experience* (Khardon 1999; Martin & Geffner 2000; C. Guestrin *et al.* 2003; Hernandez-Gardiol & Kaelbling 2003; Kersting & Raedt 2004; Fern, Yoon, & Givan 2006). There has also been some work in combining the two (Gretton & Thiébaux 2004).

Reasoning approaches can achieve optimal general policies without recourse to individual problems. On the downside they rely on expensive theorem proving and cannot give guarantees about the quality and generality of policies they compute for domains where the value of a state is drawn from an infinite set. For example, this is the case in the blocks-world because the value of a state given any policy is the expected number of actions it takes for that policy to achieve the goal, which in turn is related to the number of blocks. Inductive learning approaches have a significant advantage over reasoning approaches because they avoid theorem proving and do not rely on an exhaustive domain description. They rarely achieve optimality, however are able to compute good policies with very little effort. On the downside, we are not aware of any technique for learning a good general policy, represented in terms of the true (or a reasonable approximation of the true) state values of that policy, for the decision-theoretic case.

Orthogonal to generalisation in planning, there have also been significant developments towards propositional planning with *temporally extended rewards*. In this case, rather than accepting the standard scenario where reward is allocated to individual states, reward is allocated to sequences of states called *rewarding behaviours*. Typical examples of rewarding behaviours occur where we reward the maintenance of some property, the periodic achievement of some objective, the achievement of an objective after a trigger has occurred (and not expired), or the first achievement of an objective. These rewards are not supported in a reasonable way where problems are modelled using Markov decision processes (MDPs), the standard problem representation formalism. In particular, for an MDP we say both dynamics and reward are *Markovian* because, at any time both the effects of an action and the reward allocated are determined completely by the state the process is in. Moreover, although it may be possible in principle to manually compile temporally extended rewards into an MDP, by adding propositions that capture temporal events, the original structure is lost on an MDP solution algorithm that is not aware of the temporal interpretation of some state characterising propositions. In order to address weaknesses in the MDP model where temporally extended rewards are involved, formalisms and solution methods have been proposed for decision processes with non-Markovian rewards (NMRDPs) (Thiébaux *et al.* 2006). For an NMRDP, the problem dynamics is Markovian, and reward is a compact temporal logic specification

[1] This paper is a short summary of my ICAPS-07 paper with the same title.

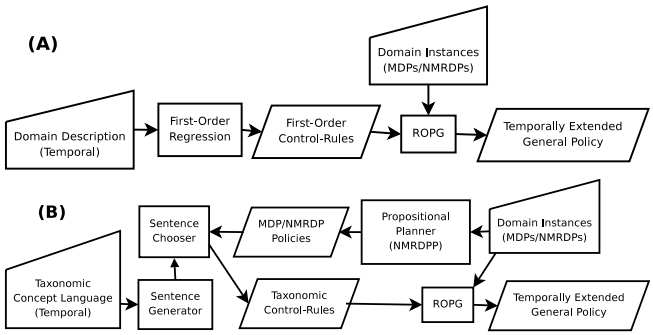

Figure 1: Data-flows for the two settings in which we use ROPG. (A) Demonstrates the case where control-rules are generated using first-order regression from an NMRDP domain description, and (B) the case where control-rules are generated according to the grammar of a taxonomic concept language.

of temporally extended rewards. NMRDP solution methods exploit the temporal logic specification of the rewarding behaviours to efficiently translate NMRDPs into equivalent MDPs amenable to MDP solution methods. Consequently, NMRDP solution techniques still succumb to Bellman's curse.

Our Contribution

We developed ROPG (Relational Online Policy Gradient), an unsupervised RRL approach to computing temporally extended policies for domains with non-Markovian rewards. ROPG itself consists of an online REINFORCE (Williams 1992) style gradient ascent optimisation strategy where *controls* correspond to relational control-rules, and *observations* to the *available controls* (in the sense they prescribe an action) at an underlying state history. Figure 1 summarises the settings where we employed ROPG. The key novelty of our work is the way in which we use relationally defined control-rules as the mechanism to provide ROPG with relational observations and actions while it is learning to act in a set of problems from a given planning domain. In a similar vein to some of the more fruitful techniques for RRL such as inductive policy selection (Yoon, Fern, & Givan 2002; Gretton & Thiébaux 2004), ROPG learns policies in terms of relational control-rules. Each control-rule is a small expression in a first-order language that can be interpreted at an NMRDP state to provide an action prescription. We adapt two very different techniques from the literature for generating relational control-rules, and have described and evaluated each in the full paper. The first technique is based on (Fern, Yoon, & Givan 2006). Relational control-rules are generated more or less arbitrarily according to the grammar of a temporal taxonomic concept language. In order to avoid redundant taxonomic control-rules, and also to avoid overwhelming the learner with too many rules, they are evaluated in small problems and only a small number of rules that behave well according to the optimal decision-theoretic planner NMRDPP (Thiébaux *et al.* 2006) are passed to ROPG. The second technique is based on (Gretton & Thiébaux 2004). In this case we exploit first-order regression to generate control-rules, from a given domain description, that are guaranteed to cover all concepts relevant to any optimal n-state-to-go value function. To this end, we developed a domain description language which accommodates non-Markovian rewards, and also extend the standard definition of first-order regression (Boutilier, Reiter, & Price 2001) to this setting.

We have implemented our approach in C++ including functionality for generating control-rules according to: (1) the extended taxonomic syntax from (Fern, Yoon, & Givan 2006), (2) our own extension of that language with temporal operators, and (3) first-order decision-theoretic regression (Gretton & Thiébaux 2004). Figure 2 demonstrates the convergence of ROPG in the case where control-rules are generated according to each of these mechanisms. In particular, convergence is shown for a set of 15 problems from the deterministic blocks-world domain with the usual Markovian reward scheme, and 11 problems from a stochastic variant of miconic (Koehler & Schuster 2000) (elevator scheduling) with a simple non-Markovian reward when a passenger is served for the first time. Essentially we find that ROPG yields good policies for these problem sets, and that learning with the temporal features is both achievable and helpful. Moreover, ROPG is useful in the case where the target domain is Markovian and non-Markovian. In the full paper we explore how the policies learnt for Figure 2 perform in large problems. We found that such policies did not always generalise. For example, in the blocks-world, the policy that was learnt on the 15 training problems for Figure 2 is only reliable on large (20 and 30 block) problems if the goal is to reverse a stack of blocks on the table.

Summary

ROPG is a version of online policy gradient, and thus learns by acting in problems. ROPG is the first reported technique for direct relational reinforcement-learning of general policies which does not rely on a state-based planning (or learning) mechanism. This means ROPG does not capture action decisions made by an optimal (or good) state-based planning agent which is inevitably better equipped to distinguish states according to propositional features that are not available to a relational learner. Along the same lines, ROPG also addresses some pitfalls of value-based relational reinforcement-learning in the setting of decision-theoretic planning with geometrically discounted future-rewards. Our approach directly learns a policy, thus it does not attempt to classify the infinite states from a target planning domain according to real values. Rather, it classifies those states according to an infinite set of actions prescribed by a small set of relational control-rules. To summarise, ROPG learns a general policy directly by acting in domain instances. Moreover, ROPG is not crippled by a reinforcement-learning scheme which punishes a learner for not mimicking the actions of "problem specialist" in the form of a state-based agent.

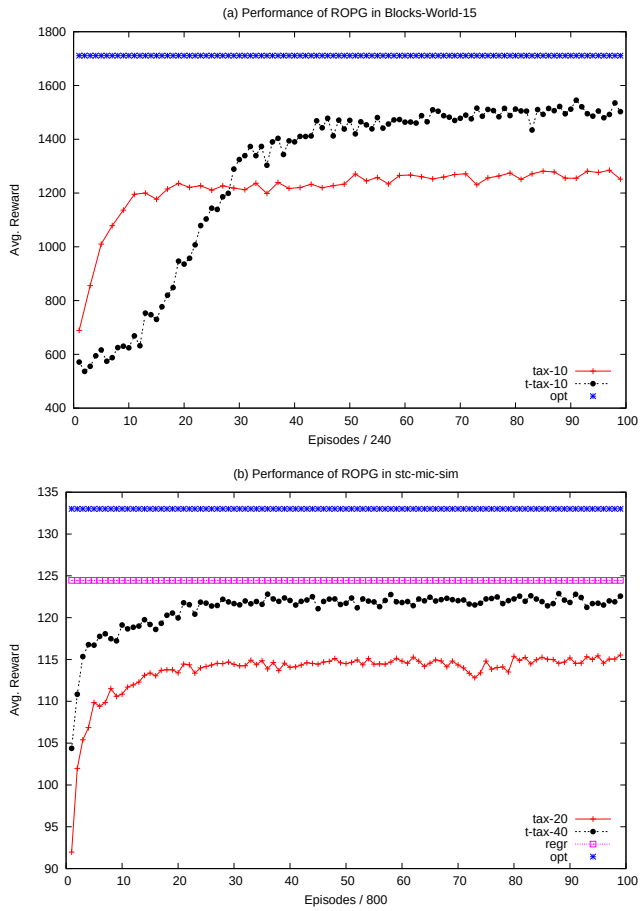

Figure 2: For control-rules generated according to a temporal taxonomic syntax (*t-tax-N*) and the extended taxonomic syntax (*tax-N*), we report average discounted reward as ROPG undertakes episodes in problems from (a) blocks-world with up to 5 blocks and (b) a stochastic version of a simple miconic (elevator scheduling) domain with up to 7 people and floors. *N* reports the number of control-rules that were given to the learner. We also report (*opt*) the optimal performance according to NMRDPP, and (*regr*) the performance of the best policy obtained by ROPG with control-rules generated by regression from a domain description. We omit *regr* in the case of blocks-world because control-rules based on regression do not work in that domain [Gretton and Thiébaux, 2004].

References

Boutilier, C.; Reiter, R.; and Price, B. 2001. Symbolic dynamic programming for first-order MDPs. In *IJCAI-01*.

C. Guestrin; D. Koller; C. Gearhart; and N. Kanodia. 2003. Generalizing plans to new environments in relational MDPs.

Fern, A.; Yoon, S.; and Givan, R. 2006. Approximate policy iteration with a policy language bias: Solving relational Markov decision processes. *J. Artif. Intell. Res. (JAIR)* 25.

Gretton, C., and Thiébaux, S. 2004. Exploiting first-order regression in inductive policy selection. In *UAI*.

Hernandez-Gardiol, N., and Kaelbling, L. P. 2003. Envelope-based planning in relational mdps. In *NIPS-17*.

K. Kersting; M. V. Otterlo; and L. D. Raedt. 2004. Bellman goes relational. In *ICML*, 59.

Karabaev, E., and Skvortsova, O. 2005. A Heuristic Search Algorithm for Solving First-Order MDPs. In *UAI*.

Kersting, K., and Raedt, L. D. 2004. Logical markov decision programs and the convergence of logical td(lambda). In *ILP*, 180–197.

Khardon, R. 1999. Learning action strategies for planning domains. *Artificial Intelligence* 113(1-2):125–148.

Koehler, J., and Schuster, K. 2000. Elevator control as a planning problem. In *AIPS*.

Littman, M. L.; Goldsmith, J.; and Mundhenk, M. 1998. The computational complexity of probabilistic planning. *J. Artif. Intell. Res. (JAIR)* 9:1–36.

Martin, M., and Geffner, H. 2000. Learning generalized policies in planning using concept languages. In *KR*, 667–677.

Sanner, S., and Boutilier, C. 2005. Approximate linear programming for first-order mdps. In *UAI*.

Thiébaux, S.; Gretton, C.; Slaney, J.; Price, D.; and Kabanza, F. 2006. Decision-theoretic planning with non-markovian rewards. *J. Artif. Intell. Res. (JAIR)* 25:17–74.

Wang, C.; Joshi, S.; and Khardon, R. 2007. First order decision diagrams for relational MDPs. In *IJCAI-07*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8:229–256.

Yoon, S. W.; Fern, A.; and Givan, R. 2002. Inductive policy selection for first-order mdps. In *UAI*, 569–576.