Using bisimulation for policy transfer in MDPs

(Extended Abstract)

Pablo S. Castro
McGill University
Montreal, QC, Canada
pcastr@cs.mcgill.ca

Doina Precup
McGill University
Montreal, QC Canada
dprecup@cs.mcgill.ca

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Theory

Keywords

Markov Decision Processes, Bisimulation, Policy transfer

1. MAIN RESULTS

Much of the work on using Markov Decision Processes (MDPs) in artificial intelligence (AI) focuses on solving a single problem. However, AI agents often exist over a long period of time, during which they may be required to solve several related tasks. This type of scenario has motivated a significant amount of recent research in *knowledge transfer* methods for MDPs. The idea is to allow an agent to continue to re-use the expertise accumulated while solving past tasks over its lifetime (see Taylor & Stone, 2009, for a comprehensive survey).

We focus on transferring knowledge in MDPs that are fully specified by their state set S, action set A, reward function $R : S \times A \to \mathbb{R}$ and state transition probabilities $P : S \times A \to Dist(S)$ (where $Dist(S)$ is the set of distributions over the set S). A policy π is a function from states to actions, $\pi : S \to A$. The value of a state $s \in S$ under policy π is defined as $V^{\pi}(s) = \mathbb{E}_{\pi}\{\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s\}$, where r_t is the reward received at time step t, and $\gamma \in (0,1)$ is a discount factor. Solving an MDP means finding the optimal value function $V^*(s) = \max_{\pi} V^{\pi}(s)$, and the associated policy π^*. The action-value function, $Q^* : S \times A \to \mathbb{R}$ gives the expected return for each state-action pair, if they are followed by the optimal policy thereafter.

Let $M_1 = \langle S_1, A_1, P_1, R_1 \rangle$ and $M_2 = \langle S_2, A_2, P_2, R_1 \rangle$ be two MDPs and let $V_1^*(Q_1^*)$ and $V_2^*(Q_2^*)$ denote their respective optimal value functions. Our goal is to provide methods for transferring a policy from ont MDP to the other, which ensuring strong theoretical guarantees regarding the expected return of the transferred policy in the new MDP.

Our methods are based on bisimulation metrics, introduced by Ferns, Panangaden & Precup (2004) . Bisimulation is a notion of behavioral equivalence between states, in which two states are equivalent, loosely speaking, if they have the same immediate rewards, and the same probabilities of transitioning to equivalent states. Bisimulation metrics turn the equivalence, which can be brittle, into a more robust estimate. Ferns et al have shown that two states from an MDP that are close in the metric also have close optimal values. Bisimulation metrics can be computed by means of an iterative algorithm and the accuracy of the computed metric depends on the number of iterations.

In this abstract we propose methods for using bisimulation-style metrics for policy transfer between MDPs. The methods differ in terms of theoretical guarantees as well as empirical performance and computational efficiency. A detailed presentation of the proofs, as well as more experiments, is available in a technical report (Castro & Precup, 2010).

Bisimulation-metric algorithm: Suppose that $A_1 = A_2$ and let $d_{\sim} : S_1 \times S_2 \to \mathbb{R}$ denote the bisimulation metric between the state sets of the two MDPs. For any $t \in S_2$, let the policy $\pi_{\sim}$ on M_2 be defined as: $\pi_{\sim}(t) = \pi^*(\arg\min_{s \in S_1} d_{\sim}(s,t))$.

THEOREM 1.1. *For all $t \in S_2$ let $a_t = \pi_{\sim}(t)$. Then:*

$$|Q_2^*(t,a_t) - V_2^*(t)| \leq 2 \min_{s \in S_1} d_{\sim}(s,t),$$

and this bound is tight.

Lax-bisimulation algorithm: A shortcoming of the previous approach is that it requires both MDPs to have the same action sets. Lax bisimulation metrics (Taylor, Panangaden & Precup, 2009) are very similar to bisimulation metrics but work with state-action pairs instead. The lax bisimulation distance between states then takes into account the best matching of actions. Let d_L denote the lax bismulation distance between states. For any $t \in S_2$ let $s_t = \arg\min_{s \in S_1} d_L(s,t)$ be the closest state to t. We define the transferred policy as: $\pi_L(t) = \min_{b \in A_2} d((s_t, \pi^*(s_t)), (t,b))$.

THEOREM 1.2. *For all $t \in S_2$ let $a_t = \pi_L(t)$. Then $|Q_2^*(t,a_t) - V_2^*(t)| \leq 2 d_L(s_t,t)$, and this bound is tight.*

Pessimistic algorithm: We can speed up the computation of the metric by only considering the optimal actions in the source system, yielding a new state distance function $d_{\approx}$ and corresponding policy $\pi_{\approx}$. The following result gives a lower bound on the value of the transferred action:

THEOREM 1.3. *For all $s \in S_1$, $t \in S_2$, let $a_t = \pi_{\approx}(t)$. Then $Q_2^*(t,a_t) \geq V_1^*(s) - d_{\approx}(s,t)$.*

For any $t \in S_2$ let $s_t = \arg\max_{s \in S_1} \{V_1^*(s) - d_{\approx}(s,t)\}$. We define the transferred policy as $\pi_{Pess}(t) = \min_{b \in A_2} d_{\approx}((s_t, \pi^*(s_t)), (t,b))$.

Optimistic algorithm: The previous algorithms all suffer from an inherent "pessimism" in bisimulation metrics, which are always

Cite as: Using bisimulation for policy transfer in MDPs (Extended Abstract), Pablo S. Castro and Doina Precup, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1399-1400

driven by the action that maximizes the distance between two states. In practice, this produces guarantees on performance for the worst case, but may produce poor transfer for less pathological problems. To get an optimistic algorithm, we can consider only the optimal actions in the source system, and only their best matches in the target system. This leads to the following heuristic dissimilarity measure:

$$d_{Opt}(s,t) = \min_{b \in A_2} d_{\approx}((s,a_s^*),(t,b))$$

which can be used in the previous approach, instead of $d_{\approx}$.

Approximations: To speed up the computation of bisimulation metrics, we suggest two approximations. First, one can iterate the bisimulation metric computation algorithm only once, using the immediate reward as a myopic distance estimate. In the second approximation we split the reward region into a fixed number of intervals and cluster states according to the reward interval to which they belong, once again iterating the bisimulation metric computation only once. If the reward structure is relatively sparse, few reward intervals are needed, and computation will be faster.

Temporal abstraction: All the metrics and algorithms discussed above generalize easily to work with temporally extended actions, in the options framework (Sutton, Precup & Singh, 1999) . Using options produces better transfer results, as illustrated in previous work, as well as in our experiments.

2. EXPERIMENTAL RESULTS

To illustrate the performance of the various policy transfer algorithms, we used a gridworld navigation task consisting of four rooms in a square (a room in each corner) connected by four hallways (one between each pair of rooms). There are four primitive actions, $\wedge$, $\vee$, $<$ and $>$, along with four analogous options, **u**, **d**, **l** and **r**, available in every state. If an agent chooses option **u**, then the option will take it to the hallway above its position, or to the middle of the upper wall (if there is no hallway in that direction). The option terminates as soon as the agent reaches the respective hallway or position along the wall. All other options are similar. There is a single goal placed in the right hallway, yielding a reward of 1; all other rewards are 0. The source MDP M_1 has only 8 states, one for each room and one for each hallway. Only the primitive actions are enabled. The target domain M_2 has 44 states, and can have either primitive actions only, or both primitives and options (results in Figure 1). We also ran experiments where either MDP has one room removed, whose results are presented in Figure 2.

In the graphs, we show results for standard Q-learning that starts with an agent that uses the transferred policy initially, and overrides it only if the value of some other action (or option) is better. In all results, the optimistic approach (pink line) yields the fastest learning speed, with the bisimulation approximants (black and dotted lines) coming in second. A similar result is obtained using the max-norm distance between the value of the transferred policy and the optimal value in the target system (results not shown here): the optimistic approach outperformed the first three algorithms in this measure, as well as in running time. The results obtained are consistent across all problems.

From Figure 1, we also note that options help transfer and speed up learning as well (not that the right panel has a larger range on the y-axis than the left panel).

3. CONCLUSIONS AND FUTURE WORK

We presented four new bisimulation-based policy transfer algorithms and two approximation ideas for performing policy transfer on MDPs. The initial results shown here are very promising,

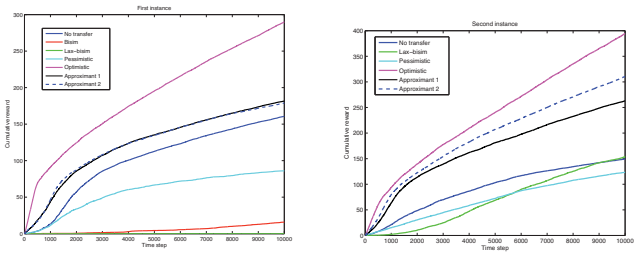

Figure 1: Comparison of performance of transfer algorithms (4 rooms to 4 rooms). Left: Only primitive actions. Right: Both primitive actions and options.

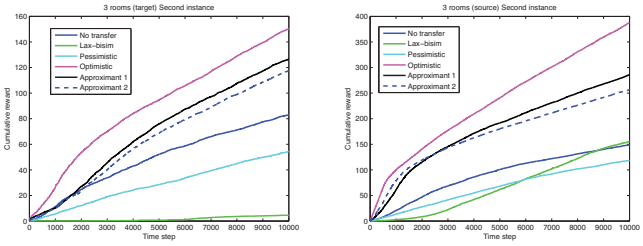

Figure 2: Comparison of performance of transfer algorithms (left: 4 rooms to 3 rooms, right: 3 rooms to 4 rooms)

but more work needs to be done to assess the empirical merit of these methods. The most promising is the optimistic approach. Although it lacks theoretical guarantees, it overcomes the pessimism of bisimulation metrics, and provides much faster computation. The extensions of all these algorithms to using temporally extended actions are straightforward and provide much better empirical results than using just primitive actions.

4. REFERENCES

[1] N. Ferns, P. Panangaden, and D. Precup. Metrics for finite Markov decision processes. In *Proceedings of UAI*, pages 162–169, 2004.

[2] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[3] J. Taylor, D. Precup, and P. Panangaden. Bounding performance loss in approximate MDP homomorphisms. In *Advances in Neural Information Processing Systems 21*, pages 1649–1656, 2009.

[4] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.