Using Bisimulation for Policy Transfer in MDPs

Pablo Samuel Castro and **Doina Precup**
School of Computer Science, McGill University, Montreal, QC, Canada
pcastr@cs.mcgill.ca and dprecup@cs.mcgill.ca

Abstract

Knowledge transfer has been suggested as a useful approach for solving large Markov Decision Processes. The main idea is to compute a decision-making policy in one environment and use it in a different environment, provided the two are "close enough". In this paper, we use bisimulation-style metrics (Ferns et al., 2004) to guide knowledge transfer. We propose algorithms that decide what actions to transfer from the policy computed on a small MDP task to a large task, given the bisimulation distance between states in the two tasks. We demonstrate the inherent "pessimism" of bisimulation metrics and present variants of this metric aimed to overcome this pessimism, leading to improved action transfer. We also show that using this approach for transferring temporally extended actions (Sutton et al., 1999) is more successful than using it exclusively with primitive actions. We present theoretical guarantees on the quality of the transferred policy, as well as promising empirical results.

Introduction

Autonomous intelligent agents are often faced with the problem of making decisions with favorable long-term consequences in the presence of stochasticity. In this paper, we consider this problem in the context of Markov Decision Processes (MDPs) (Puterman, 1994), in which the agent has to find a way of behaving that maximizes its long-term expected return. Much of the work on using MDPs in AI and operations research focuses on solving a single problem. However, in practice, AI agents often exist over a longer period of time, during which they may be required to solve several, related tasks. For example, a physical robot may be in use for a period of several years, during which it has to navigate to different locations, pick up different objects, etc. Typically, these tasks are distinct, but share important properties (e.g., the robot may be located in one specific building, where all its tasks take place). 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. Several approaches for knowledge transfer in MDPs have been proposed; Taylor & Stone (2009) provide a comprehensive survey. Broadly speaking, the goal of knowledge transfer is two-fold. On one hand, it should speed up the process of solving new tasks. On the other hand, it should enable solving tasks that are very complex in the given (raw) representation. The first goal has been emphasized more in reinforcement learning, while the second goal is more prevalent in general machine learning.

In this paper we focus on transferring knowledge in MDPs that are specified fully by their states, actions, rewards and model of state transition probabilities. The knowledge to be transferred is in the form of a *policy*, i.e. a way of behaving for the agent. The goal is to specify a transfer method with strong guarantees on the expected returns of this policy in the new MDP. In particular, we focus on bisimulation metrics (Ferns et al., 2004; Taylor et al., 2009), which measure the long-term behavioral similarity of different states. States which are "close" in terms of these metrics also have similar expected returns (Ferns et al., 2004). However, bisimulation suffers from three drawbacks. The metrics are very expensive to compute; the optimal policy or value function are irrelevant to the metrics; and their estimates tend to be too pessimistic. We present a variant of the bisimulation metrics which overcomes these problems and improves the empirical behavior significantly, while still retaining good theoretical properties (in some cases).

Previous work has also illustrated the fact that using temporally extended actions (and their models) can significantly improve knowledge transfer in MDPs (e.g., Perkins & Precup, 1999, Andre & Russell, 2002; Ravindran & Barto, 2003; Konidaris & Barto, 2007). Intuitively, it is easier to transfer high-level controls rather than low-level primitive actions. For instance, someone giving driving directions will use high-level specifications (such as street names and physical landmarks), and will not mention lower-level controls, such as how to drive a car. This overcomes many of the difficulties that arise when comparing dynamics on a primitive-action level: different individuals will have differences in how they drive, but the high-level description will "smooth" them out. We establish bisimulation metrics for MDPs with temporally extended actions, using the framework of options (Sutton et al., 1999). All theoretical results hold in this case as well, and options provide better empirical behavior.

We first introduce notation and discuss related work, then present knowledge transfer using bisimulation metrics and discuss its theoretical properties; the approximation to the bisimulation metrics follows. The use of bisimulation with options is introduced, followed by an empirical illustration.

Background

A Markov decision process (MDP) is a 4-tuple $\langle S,A,P,R\rangle$, where S is a finite state space, A is a finite set of actions, $P : S\times A \to Dist(S)$ [1] specifies the next-state transition probabilities and $R : S\times A \to \mathbb{R}$ is the reward function. A policy $\pi : S \to A$ specifies the action choice for each state. The value of a state $s \in S$ under a policy π is defined as: $V^\pi(s) = \mathbb{E}_\pi\{\sum_{t=0}^{\infty}\gamma^t r_t | s_0 = s\}$, 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 $V^*(s) = \max_\pi V^\pi(s)$ and the associated policy π^*. In a finite MDP, there is a unique optimal value function and at least one deterministic optimal policy. The action-value function, $Q^* : S\times A \to \mathbb{R}$, gives the optimal value for each state-action pair, given that the optimal policy is followed afterwards. It obeys the Bellman equations:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'\in S} P(s,a)(s') \max_{a'\in A} Q^*(s',a')$$

Several types of knowledge can be transferred between MDPs. Existing work includes transferring models (e.g., Sunmola & Wyatt, 2006), using samples obtained by interacting with one MDP to learn a good policy in a different MDP (e.g., Lasaric et al., 2008), transferring values (e.g., Ferrante et al., 2008), or transferring policies. In this paper, we focus on the latter approach, and mention just a few pieces of work, most closely related to ours. The main idea of policy transfer methods is to take policies learned on small tasks and apply them to larger tasks. Sherstov & Stone (2005) show how policies learned previously can be used to restrict the policy space in MDPs with many actions. Taylor et al. (2007) transfer policies, represented as neural network action selectors, from a source to a target task. A hand-coded mapping between the two tasks is used in the process. MDP homomorphisms (Ravindran & Barto, 2002) allow correspondences to be defined between state-action pairs, rather than just states. Follow-up work (e.g. Ravindran & Barto, 2003; Konidaris & Barto, 2007) uses MDP homomorphisms and options to transfer knowledge between MDPs with different state and action spaces. Wolfe & Barto (2006) construct a reduced MDP using options and MDP homomorphisms, and transfer the policy between two states if they both map to the same state in the reduced MDP. Unfortunately, because the work is based on an equivalence relation, rather than a metric, small perturbations in the reward or transition dynamics make the results brittle. Soni & Singh (2006) transfer policies learned in a small domain as options for a larger domain, assuming that a mapping between state variables is given. A closely related idea was presented in (Sorg & Singh, 2009) where the authors use soft homomorphisms to perform transfer and provide theoretical bounds on the loss incurred from the transfer.

Knowledge transfer using bisimulation metrics

Suppose that we are given two MDPs $M_1 = \langle S_1,A,P,R\rangle$, $M_2 = \langle S_2,A,P,R\rangle$ with the same action sets, and a metric $d : S_1\times S_2 \to \mathbb{R}$ between their state spaces. We define the policy π_d on M_2 as

$$\forall t \in S_2. \qquad \pi_d(t) = \pi^*(\arg\min_{s\in S_1} d(s,t)) \tag{1}$$

In other words, $\pi_d(t)$ does what is optimal for the state in S_1 that is closest to t according to metric d. Note that the mapping between the states of the two MDPs is defined implicitly by the distance metric d.

Bisimulation for MDPs was defined by (Givan et al., 2003) based on the notion of probabilistic bisimulation from process algebra (Larsen & Skou, 1991). Intuitively, bisimilar states have the same long-term behavior.

Definition 1. *A relation $E \subseteq S\times S$ is said to be a bisimulation relation if whenever sEt:*

1. $\forall a \in A. \qquad R(s,a) = R(t,a)$
2. $\forall a.\forall C \in S/E. \sum_{s'\in C} P(s,a)(s') = \sum_{s'\in C} P(t,a)(s')$

where S/E is the set of all equivalence classes in S w.r.t equivalence relation E. Two states s and t are called bisimilar, denoted $s \sim t$, if there exists a bisimulation relation E such that sEt.

Ferns et al., (2004) defined a bisimulation metric and proved that it is an appropriate quantitative analogue of bisimulation. The metric is not brittle, like bisimulation: if the transitions or rewards of two bisimilar states are changed slightly, the states will no longer be bisimilar, but they will remain close in the metric. A metric d is a bisimulation metric if for any $s,t \in S$, $d(s,t) = 0 \Leftrightarrow s \sim t$.

The bisimulation metric is based on the Kantorovich probability metric $T_K(d)(P,Q)$ applied to state probability distributions P and Q, where d is a semimetric on S. It is defined by the following primal linear program (LP):

$$\max_{u_i, i=1,\cdots,|S|} \sum_{i=1}^{|S|} (P(s_i) - Q(s_i))u_i \tag{2}$$
$$\text{subject to: } \forall i,j. u_i - u_j \le d(s_i,s_j)$$
$$\forall i. 0 \le u_i \le 1$$

Intuitively, $T_K(d)(P,Q)$ calculates the cost of "converting" P into Q under d. The dual formulation, which is typically solved, is a Minimum Cost Flow (MCF) problem (see Ferns et al., 2004 for details).

Theorem 2. (From Ferns et al., 2004) *Let M be the set of all semimetrics on S and define $F : M \to M$ by $F(d)(s,s') = \max_{a\in A}(|R(s,a) - R(s',a)| + \gamma T_K(d)(P(s,a),P(s',a)))$. Then F has a least-fixed point, $d_\sim$, and $d_\sim$ is a bisimulation metric.*

Phillips (2006) used bisimulation metrics to transfer policies in MDPs, assuming that a state mapping between the MDPs is given. Here, we relax this requirement and let the mapping be determined automatically from bisimulation.

When transferring knowledge between MDPs, we are only interested in computing the distance between the states in S_1 and the states in S_2, but not between states of the same

[1] $Dist(X)$ is the set of distributions over the set X

MDP. Because of this, the primal LP (2) can be rewritten as:

$$\max_{u_i,i=1,\cdots,|S_1|,v_i,i=1,\cdots,|S_2|} \sum_{i=1}^{|S_1|} P(s_i)u_i - \sum_{j=1}^{|S_2|} Q(s_j)v_j$$
$$\text{subject to: } \forall i,j.u_i - v_j \leq d(s_i,s_j)$$
$$\forall i. -1 \leq u_i \leq 1$$

Let $V_1^*(Q_1^*)$ and $V_2^*(Q_2^*)$ denote the optimal policies (optimal Q-values) for M_1 and M_2, respectively. The following lemmas are necessary for Theorem 5.

Lemma 3. $\forall s \in S_1, t \in S_2, |V_1^*(s) - V_2^*(t)| \leq d_\sim(s,t)$.

Proof: See the proof of Theorem 5.1 in (Ferns et al., 2004)

Lemma 4. *For all $t \in S_2$ let $s_t = \arg\min_{s\in S_1} d_\sim(s,t)$ and $a_t^T = \pi^*(s_t)$. Then $|Q_2^*(t,a_t^T) - V_1^*(s)| \leq d_\sim(s,t)$.*

Proof.

$$\begin{aligned}
&|Q_2^*(t,a_t^T) - V_1^*(s)| = |Q_2^*(t,a_t^T) - Q_1^*(s,a_t^T)| \\
&\leq \max_{a\in A} \Big\{ \big|R(t,a_t^T) - R(s,a_t^T)\big| \\
&\quad + \gamma \left| \sum_{t'\in S_2} P(t,a_t^T)(t')V_2^*(t') - \sum_{s'\in S_1} P(s,a_t^T)(s')V_1^*(s') \right| \Big\} \\
&\leq \max_{a\in A} \big\{ \big|R(t,a_t^T) - R(s,a_t^T)\big| + \gamma T_K(d_\sim)(P(t,a),P(s,a)) \big\} \\
&= d_\sim(s,t)
\end{aligned}$$

where the second to last line follows from the fact that V_1^* and V_2^* together constitute a feasible solution to primal LP $T_K(d_\sim)$ by Lemma 3. □

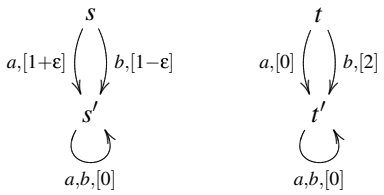
We can now use the last lemmas to bound the loss incurred when using the transferred policy.

Theorem 5. *For all $t \in S_2$ let $a_t^T = \pi_\sim(t)$. Then: $|Q_2^*(t,a_t^T) - V_2^*(t)| \leq 2\min_{s\in S_1} d_\sim(s,t)$.*

Proof: Let $s_t = \arg\min_{s\in S_1} d_\sim(s,t)$. We have:

$$\begin{aligned}
|Q_2^*(t,a_t^T) - V_2^*(t)| &\leq |Q_2^*(t,a_t^T) - V_1^*(s_t)| + |V_1^*(s_t) - V_2^*(t)| \\
&\leq 2d_\sim(s_t,t) \qquad \text{(by Lemmas 4 and 3)} \\
&= 2\min_{s\in S_1} d_\sim(s,t) \qquad \square
\end{aligned}$$

The following simple example proves that the above bound is tight. Consider the following two systems:



The two available actions are a and b. The numbers in brackets indicate the reward received when following that branch. The optimal action for state s is a, yielding $V_1^*(s) = 1+\varepsilon$, while the optimal action for state t is b, yielding $V_2^*(t) = 2$. Since s' and t' are bisimilar states, s and t have the same probability of transitioning to all bisimulation equivalence classes. Thus, $d_\sim(s,t) = 1+\varepsilon$ and $d_\sim(s',t) = 2$; so the policy from s will be transferred to t, yielding $Q_2^*(t,a_t^T) = 0$. We then have $|Q_2^*(t,a_t^T) - V_2^*(t)| = 2 \leq 2(1+\varepsilon) = 2d_\sim(s,t)$.

A shortcoming of the bisimulation approach is that it requires both systems to have the same action sets. This not only restricts the target domain to those that have equal action sets as the target, but it also means the transfer will not work if the target domain has a different ordering for the actions. To overcome this problem, Taylor et al. (2009) introduced lax bisimulation metrics, d_L. The idea is to have a metric for state-action pairs rather than just for state pairs. Given two MDPs $M_1 = \langle S_1, A_1, P_1, R_1\rangle$ $M_2 = \langle S_2, A_2, P_2, R_2\rangle$, for all $s \in S_1$, $t \in S_2$, $a \in A_1$ and $b \in A_2$, let $d_L((s,a),(t,b)) = |R_1(s,a) - R_2(t,b)| + \gamma T_K(d_L)(P_1(s,a),P_2(t,b))$. From the distance between state-action pairs we can then define a state lax-bisimulation metric. We use the same symbol d_L for the state lax-bisimulation metric, but the arguments will resolve any ambiguity. For all $s \in S_1$ and $t \in S_2$, the metric is defined as: $d_L(s,t) = \max\big(\max_{a\in A_1}\min_{b\in A_2} d((s,a),(t,b)),$ $\max_{b\in A_2},\min_{a\in A_1} d((s,a),(t,b))\big)$ The transferred policy can be computed as follows.

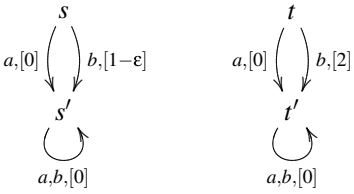
Algorithm 1 laxBisimTransfer(S_1, S_2)

```
Compute d_L
for all t ∈ S_2 do
    s_t ← arg min_{s∈S_1} d_L(s,t)
    b_t = min_{b∈A_2} d_L((s_t, π*(s_t)),(t,b))
    π_L(t) ← b_t
end for
return π_L
```

In other words $\pi_L(t)$ finds the closest state $s \in S_1$ to t under d_L and then chooses the action b from t that is closest to $\pi^*(s)$. With little effort we obtain the following result.

Theorem 6. *For all $t \in S_2$, let $a_t^T = \pi_L(t)$; then $|Q_2^*(t,a_t^T) - V_2^*(t)| \leq 2\min_{s\in S_1} d_L(s,t)$.*

The following example demonstrates that the bound of Theorem 6 is tight.



We can see that $d_L(s,t) = d_L((s,b),(t,b)) = 1+\varepsilon$ and $d_L(s',t) = 2$, so the policy from s will be used to transfer to t. Since $\pi^*(s) = b$ and $a = \arg\min_{c\in\{a,b\}} d((s,b),(t,c))$, $\pi^T(t) = a$, yielding $Q_2^*(t,a_t^T) = 0$. Thus, $|Q_2^*(t,a_t^T) - V_2^*(t)| = 2 \leq 2(1+\varepsilon) = d_L(s,t)$.

Speeding up the computation

A problem that haunts bisimulation methods is the computation time. Because many MCF problems have to be solved, the time it takes to compute these metrics is prohibitive. Although we are considering all actions in the source system,

we really only transfer the optimal ones. This suggests a simple way to modify the lax bisimulation approach to speed up the computation of the metric by only considering the optimal actions in the source system. Given two MDPs $M_1 = \langle S_1, A_1, P_1, R_1 \rangle$ $M_2 = \langle S_2, A_2, P_2, R_2 \rangle$, for all $s \in S_1$, $t \in S_2$ and $b \in A_2$, where $a_s^* = \pi^*(s)$,

$$d_{\approx}(s,(t,b)) = |R_1(s,a_s^*) - R_2(t,b)| + \gamma T_K(d_L)(P_1(s,a_s^*), P_2(t,b))$$

Again, we use the same symbol $d_{\approx}$ for the state metric, but the arguments will resolve any ambiguity:

$$\forall s \in S_1, \forall t \in S_2,\ d_{\approx}(s,t) = \max_{b \in A_2} d_{\approx}(s,(t,b))$$

We can now use Algorithm 1 again, with the new metric $d_{\approx}$, to obtain the transferred policy $\pi_{\approx}$. In other words $\pi_{\approx}(t)$ finds the closest state $s \in S_1$ to t under $d_{\approx}$ and then chooses the action b from t that is closest to $\pi^*(s)$.

Lemma 7. *For all $s \in S_1$, $t \in S_2$ $|V_1^*(s) - V_2^*(t)| \leq d_{\approx}(s,t)$.*

Proof: $|V_1^*(s) - V_2^*(t)| = |Q_1^*(s,a_s^*) - Q_2^*(t,a_t^*)|$

$\leq |R(s,a_s^*) - R(t,a_t^*)| + \gamma T_K(d_{\approx}(P(s,a_s^*), P(t,a_t^*))$

by a similar argument as for Lemma 4

$= d_{\approx}((s,a_s^*),(t,a_t^*)) \leq d_{\approx}(s,t)$ □

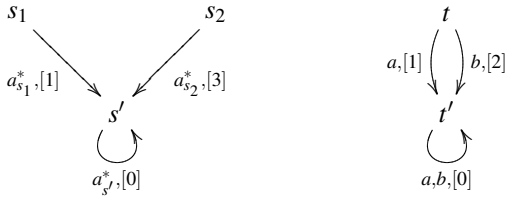
Corollary 8. *For all $s \in S_1$, $t \in S_2$ and $b \in A_2$, $|Q_2^*(t,b) - V_1^*(s)| \leq d_{\approx}(s,t)$.*

Corollary 9. *For all $s \in S_1$, $t \in S_2$, let $a_t^T = \pi_{\approx}(t)$. Then $|Q_2^*(t,a_t^T) - V_1^*(s)| \leq d_{\approx}(s,t)$.*

The last results yield the following theorem.

Theorem 10. *For all $t \in S_2$ let $a_t^T = \pi_{\approx}(t)$, then $|Q_2^*(t,a_t^T) - V_2^*(t)| \leq 2\min_{s \in S_1} d_{\approx}(s,t)$.*

This last result confirms our claim that we really need only consider the optimal actions in the source MDP. However, there is still a problem inherent to the previous transfers, illustrated below. We only indicate the optimal actions in the source system.



We can see that $d_{\approx}(s_1,t) = 1$, $V^*(s_1) = 1$, $d_{\approx}(s_2,t) = 2$ and $V^*(s_2) = 3$, but $\pi_{\approx}(t) = \arg\min_{c \in \{a,b\}} d_{\approx}((s_1,a_{s_1}^*),(t,c)) = a$, yielding $V^T(t) = 1 < 2 = V^*(t)$. This illustrates an underlying problem: when performing the transfer the target system is trying to find the state in the source system which it can most closely simulate, regardless of the actual value this produces. From Corollary 9 we obtain another interesting result.

Corollary 11. *For all $s \in S_1$, $t \in S_2$, let $a_t^T = \pi_{\approx}(t)$. Then $Q_2^*(t,a_t^T) \geq V_1^*(s) - d_{\approx}(s,t)$.*

Algorithm 2 pessTransfer(S_1, S_2)

```
Compute d≈
for All t ∈ S2 do
    for All s ∈ S1 do
        LB(s,t) ← V1*(s) − d≈(s,t)
    end for
    s_t ← arg max_{s∈S1} LB(s,t)
    b_t = min_{b∈A2} d≈(s_t,(t,b))
    π_Pess(t) ← b_t
end for
return π_L
```

With this result we now have a lower bound on the value of the action transferred which takes into consideration the value function in the source system. This suggests a different algorithm to obtain transferred policy π_{Pess}, shown directly above.

In other words $\pi_{Pess}(t)$ uses the source state with the highest guaranteed lower bound on the value of its optimal action. This overcomes the problem of the last example.

Optimistic approach

The idea of the pessimistic approach is appealing as it uses the value function of the source system as well as the metric to guide the transfer. However, there is still an underlying problem in all of the previous algorithms. This is in fact a problem with bisimulation when used for transfer. There is an inherent "pessimism" in bisimulation: we always consider the action that maximizes the distance between two states. This pessimism is what equips bisimulation with all the mathematical guarantees, since we are usually "upper-bounding". However, one may (not infrequently) encounter situations where this pessimism produces a poor transfer. For instance, assume there is a source state s whose optimal action can be transferred with almost no loss as action b in a target state t (i.e. $d_{\approx}(s,(t,b))$ is almost 0); however, assume there is another action c in t such that $d_{\approx}(s,(t,c))$ is very large. This large distance may disqualify state s as a transfer candidate for state t, when it may very well be the best choice! The inherent pessimism of bisimulation would have overlooked this ideal transfer. If we would have taken a more "optimistic" approach, then we would have ignored $d_{\approx}(s,(t,c))$ and focused on $d_{\approx}(s,(t,b))$. This idea motivates the main algorithmic contribution of the paper.

We start by defining a new metric, $d_{Opt}(s,t) = \min_{b \in A_2} d_{\approx}((s,a_s^*),(t,b))$, and use Algorithm 2 but with d_{Opt} instead of $d_{\approx}$ in the computation of $LB(s,t)$ to obtain our transferred policy π_{Opt}. In other words $\pi_{Opt}(t)$ chooses the action with the highest *optimistic* lower bound on the value of the action. Unfortunately, by removing the pessimism we lose the theoretical properties but we expect to obtain better results in practice.

Bisimulation metrics for options

An option o is a triple $\langle I_o, \pi_o, \beta_o \rangle$, where $I_o \subseteq S$ is the set of states where the option is enabled, $\pi_o : S \to Dist(A)$ is the policy for the option, and $\beta_o : S \to [0,1]$ is the probability

Table 1: Running times (in seconds) and $\|V_2^* - V_t^T\|_\infty$

	First instance (4to4)		Second instance (4to4)		Second instance (4to3)		Second instance (3to4)	
Algorithm	Time	$\|V_2^* - V_t^T\|_\infty$	Time	$\|V_2^* - V_t^T\|_\infty$	Time	$\|V_2^* - V_t^T\|_\infty$	Time	$\|V_2^* - V_t^T\|_\infty$
Bisim	15.565	0.952872	-	-	-	-	-	-
Lax-Bisim	66.167	0.847645	128.135	0.749583	67.115	0.749583	100.394	0.749583
Pessimistic	25.082	0.954625	47.723	0.904437	24.125	0.875533	37.048	0.904437
Optimistic	23.053	0.335348	47.710	0.327911	24.649	0.360802	39.672	0.002052

of the option terminating at each state (Sutton et al., 1999). Options are temporally abstract actions and generalize one-step primitive actions. Given that an option o is started at state s, one can define $Pr(s'|s,o)$ as the discounted probability of ending in state s' given that the option started in state s. Similarly, one can define the expected reward received while executing the option as $\mathfrak{R}(s,o)$ (see Sutton et al., 1999 for details).

Definition 12. *A relation $E \subseteq S \times S$ is said to be an option-bisimulation relation if whenever sEt:*

1. $\forall o, \mathfrak{R}(s,o) = \mathfrak{R}(t,o)$
2. $\forall o, \forall C \in S/E. \sum_{s' \in C} Pr(s'|s,o) = \sum_{s' \in C} Pr(s'|t,o)$

Two states s and t are said to be option-bisimilar if there exists an option-bisimulation relation E such that sEt. Let $s \sim_O t$ denote the maximal option-bisimulation relation.

As before, a metric d is an option-bisimulation metric if for any $s,t \in S$, $d(s,t) = 0 \Leftrightarrow s \sim_O t$.

Ferns et al. (2004) pass the next state transition probabilities in to $T_K(d)$. However, we need to use $Pr(\cdot|s,o)$, which is a sub-probability distribution. To account for this, we add two dummy states to the MCF problem, meant to absorb any leftover probability mass. These dummy nodes are still connected like the other nodes, but with a cost of 1 (see Van Breugel & Worrell, 2001 for more details).

Theorem 13. *Let $F(d)(s,t) = \max_{o \in OPT}(|\mathfrak{R}(s,o) - \mathfrak{R}(t,o)| + \gamma T_K(d)(Pr(\cdot|s,o), Pr(\cdot|t,o))$. Then F has a least fixed-point, $d_\sim$ which is an option-bisimulation metric.*

The proof is almost identical to that of Theorem 4.5 in Ferns et al, (2004). As shown there, $d_\sim$ can be approximated to a desired accuracy δ by applying F for $\lceil \frac{\ln \delta}{\ln \gamma} \rceil$ steps. All other metric approximations presented so far generalize to the option setting in a similar, straightforward way.

Experimental results

To illustrate the performance of the policy transfer algorithms, we used the gridworld task of Sutton et al., (1999), consisting of four rooms connected by four hallways. There are four primitive actions: $\wedge$ (up), $\vee$ (down), $<$ (left) and $>$ (right). When one of the actions is chosen, the agent moves in the desired direction with 0.9 probability, and with 0.1 probability uniformly moves in one of the other three directions or stays in the same place. Whenever a move would take the agent into a wall, the agent does not move.

There are four global options available in every state, analogous to the $\wedge$, $\vee$, $<$ and $>$ primitive actions. We will refer to them as **u**, **d**, **l** and **r**, respectively. If an agent chooses option **u**, then the option will take it to the hallway above its position. If there is no hallway in that direction, then the option will take the agent to the middle of the upper wall. 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 in the right hallway, yielding a reward of 1. All other rewards are 0.

The above topology for the rooms can be instantiated with different numbers of cells. We started with an instance where there are only 8 states: one for each of the rooms, and one for each of the hallways, with the goal in the rightmost hallway. This domain only has 4 options, which are simply the primitive actions. The various metrics ($d_\sim$, d_L, $d_\approx$, and d_{Opt}) were computed between this small problem and each of the larger problems, using a desired accuracy of $\delta = 0.01$, then the policy transfer algorithms were applied. For all experiments we used the CS2 algorithm for solving MCF problems (Frangioni & Manca, 2006) and a discount factor of $\gamma = 0.9$.

We used a domain with 44 states as target for the transfer. In the first instance the target domain only had primitive actions. In the second case, the target had both primitive actions and options. The original bisimulation metric approach could not be run because of the difference in number of options. We also ran experiments where the target system only has 3 rooms (bottom right room was removed), but source still has 4 rooms, and where the source system only has 3 rooms (bottom right room was removed) and target system has 4 rooms. Table 1 displays the running times of the various algorithms, as well as $\|V_2^* - V_2^T\|_\infty$.

If we had a model distribution from which problems were sampled and we wanted to avoid solving the value function for each sampled model, we could use our algorithms to transfer the policy from the small source to just one of the target systems (the mean model, for instance), and use that policy for all of the other samples. Furthermore, this single transferred policy can be used as an initial policy for learning on any of the sampled systems. To illustrate this idea, we sampled 30 models of the large domain from an underlying Dirichlet distribution with parameter $\alpha = (46,1,1,1,1)$, where the first element corresponds to the probability of landing in the intended target state and the other parameters to landing in any of the other possible states. This Dirichlet distribution was used to sample the transition dynamics for each state-action pair in the target system. The policy transferred to the mean model is used as a starting point for Q-learning. More precisely, if the current Q-value estimate for the action suggested by the policy would be best, this action would be taken. Figure 1 shows the performance of the various algorithms in this scenario, averaged over 30 inde-

pendent runs. The optimistic approach is clearly better than the other methods, while the other methods lag behind regular Q-learning. Table 1 shows the advantage of this method in terms of running time and value function bounds as well. As expected, using options is better than using just primitive actions.

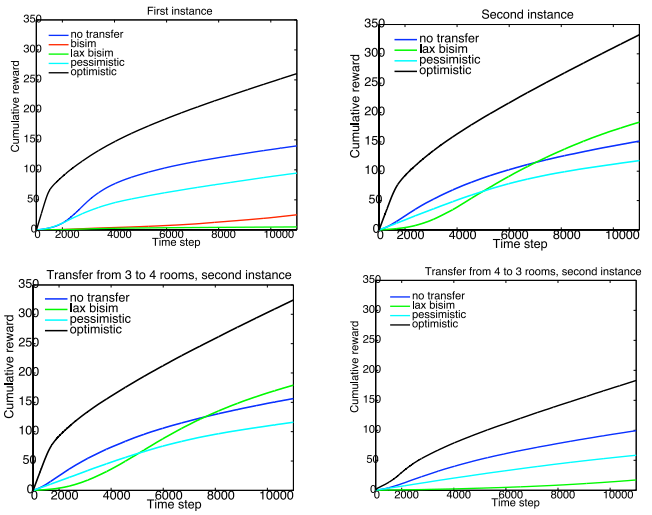


Figure 1: Comparison of performance of transfer algorithms: optimistic method (black) no transfer (dark blue), lax bisimulation (green), pessimistic (light blue) and bisimulation (red). The graphs correspond to different tasks.

Discussion and future work

The results we obtained are very promising; however, more empirical experience, in other domains, is needed. Bowling & Veloso (1999) derive a bound based on the Bellman error quantifying the loss of optimality when using policies of sub-problems to speed up learning. The motivation for considering solutions to sub-problems is similar to the motivation for using options in our case. Their bound is applicable for a particular definition of sub-problem, whereas our bounds are general. Sorg & Singh (2009) use soft homomorphisms to transfer policies, and derive bounds on the loss. The state mapping they suggest is loosely based on MDP homomorphisms and does not have continuity properties with respect to its underlying equivalence, as is the case for bisimulation metrics. Our bounds are tighter because they are state-dependent while their bound is uniform over the state space. We are currently investigating ways to approximate the metrics further, e.g. by approximating the Kantorovich metric, or clustering states before running the MCF solver to reduce the number of constraints and variables. An approach such as suggested in (Castro et al., 2009) could be used to refine the clusters in a principled way. The pessimistic and optimistic methods could be incorporated together into a branch-and-bound approach to search for a good policy that could be transferred. We are currently investigating this idea as well.

Acknowledgements

This work was funded by NSERC and ONR. We thank the anonymous reviewers for their useful comments.

References

Andre, D., and Russell, S. 2002. State abstraction for programmable reinforcement learning agents. In *AAAI*, 119–125.

Bowling, M., and Veloso, M. 1999. Bounding the suboptimality of reusing subproblems. In *IJCAI*.

Castro, P. S.; Panangaden, P.; and Precup, D. 2009. Notions of state equivalence under partial observability. In *IJCAI*.

Ferns, N.; Panangaden, P.; and Precup, D. 2004. Metrics for finite Markov decision processes. In *UAI*, 162–169.

Ferrante, E.; Lazaric, A.; and Restelli, M. 2008. Transfer of task representation in reinforcement learning using policy-based proto-value functions. In *AAMAS*, 1329–1332.

Frangioni, A., and Manca, A. 2006. A computational study of cost reoptimization for min cost flow problems. *INFORMS Journal on Computing* 18(1):61–70.

Givan, R.; Dean, T.; and Greig, M. 2003. Equivalence Notions and Model Minimization in Markov Decision Processes. *Artificial Intelligence* 147(1–2):163–223.

Konidaris, G., and Barto, A. 2007. Building portable options: Skill transfer in reinforcement learning. In *IJCAI*.

Larsen, K. G., and Skou, A. 1991. Bisimulation through probabilistic testing. *Information and Computation* 94(1):1–28.

Lazaric, A.; Restelli, M.; and Bonarini, A. 2008. Transfer of samples in batch reinforcement learning. In *ICML*, 544–551.

Perkins, T. J., and Precup, D. 1999. Using options for knowledge transfer in reinforcement learning. Technical Report UM-CS-1999-034, University of Masschusetts, Amherst.

Phillips, C. 2006. Knowledge transfer in Markov Decision Processes. Technical report, McGill University.

Puterman, M. L. 1994. *Markov Decision Processes*. Wiley.

Ravindran, B., and Barto, A. G. 2002. Model minimization in hierarchical reinforcement learning. In *Fifth Symposium on Abstraction, Reformulation and Approximation*.

Ravindran, B., and Barto, A. G. 2003. Relativized options: Choosing the right transformation. In *ICML*.

Sherstov, A. A., and Stone, P. 2005. Improving action selection in MDP's via knowledge transfer. In *AAAI*.

Soni, V., and Singh, S. 2006. Using homomorphism to transfer options across reinforcement learning domains. In *AAAI*.

Sorg, J., and Singh, S. 2009. Transfer via soft homomorphisms. In *AAMAS*.

Sunmola, F. T., and Wyatt, J. L. 2006. Model transfer for Markov decision tasks via parameter matching. In *the 25th Workshop of the UK Planning and Scheduling Special Interest Group (Plan-SIG 2006)*.

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

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

Taylor, J.; Precup, D.; and Panangaden, P. 2009. Bounding performance loss in approximate MDP homomorphisms. In *Advances in Neural Information Processing Systems 21*.

Taylor, M. E.; Stone, P.; and Liu, Y. 2007. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research* 8:2125–2167.

van Breugel, F., and Worrell, J. 2001. An algorithm for quantitative verification of probabilistic transition systems. In *CONCUR*, 336–350.

Wolfe, A. P., and Barto, A. G. 2006. Defining object types and options using MDP homomorphisms. In *the ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning*.