Learning Skill and Representation Hierarchies for Effective Control Knowledge Transfer

Mehran Asadi ASADI@CSE.UTA.EDU
Vinay Papudesi PAPUDESI@CSE.UTA.EDU
Manfred Huber HUBER@CSE.UTA.EDU
Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019 USA

Abstract

Learning capabilities of computer systems still lag far behind biological systems. One of the reasons can be seen in the inefficient re-use of control knowledge acquired over the lifetime of the artificial learning system. To address this deficiency, this paper presents a learning architecture which transfers control knowledge in the form of behavioral skills and corresponding representation concepts from one task to subsequent learning tasks. The presented system uses this knowledge to construct a more compact state space representation for learning while assuring bounded optimality of the learned task policy by utilizing a representation hierarchy.

To demonstrate this control knowledge transfer, a sequence of experiments in a video game domain is presented, illustrating its ability to reduce representational complexity and to enhance learning performance.

1. Introduction

Learning capabilities in biological systems far exceed the ones of artificial agents, partially because of the efficiency with which they can transfer and re-use control knowledge acquired over the course of their lives.

To address this, knowledge transfer across learning tasks has recently received increasing attention (Ando & Zhang, 2004; Taylor & Stone, 2005; Marthi et al., 2005; Marx et al., 2005). The type of knowledge considered for transfer includes re-usable behavioral macros, important state features, information about expected reward conditions, and background knowledge. Knowledge transfer is aimed at improving learning performance by either reducing the learning problem's complexity or by guiding the learning process.

The work presented here focuses on the construction and transfer of control knowledge in the form of behavioral skill hierarchies and associated representational hierarchies in the context of a reinforcement learning agent. In particular, it facilitates the acquisition of increasingly complex behavioral skills and the construction of appropriate, increasingly abstract and compact state representations which accelerate learning performance while ensuring bounded optimality.

Recent work in Hierarchical Reinforcement Learning (HRL) has led to approaches for learning with temporally extended actions using the framework of Semi-Markov Decision Processes (SMDPs) (Sutton et al., 1999), for learning subgoals and hierarchical action spaces (Barto & Mahadevan, 2003), and for learning abstract representations (Kim & Dean, 2003). However, most of these techniques only address one of the aspects of transfer and do frequently not directly address the construction of action and representation hierarchies in life-long learning.

The approach to hierarchical learning and knowledge construction presented here provides an integrated system which forms increasingly complex behavioral skills and corresponding state representation concepts that provide formal properties for new tasks. In particular, it forms a state hierarchy that encodes the functional properties of the skill hierarchy, providing a compact basis for learning that ensures bounded optimality.

2. Hierarchical Knowledge Transfer

In the approach presented here, skills are learned within the framework of Semi-Markov Decision Processes (SMDP) where new task policies can take ad-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

vantage of previously learned skills, leading from an initial set of basic actions to the formation of a skill hierarchy. At the same time, abstract representation concepts are derived which capture each skill's goal objective as well as the conditions under which use of the skill would predict achievement of the objective. The latter captures important aspects of the state in the context of the specific skills and is similar to "affordances" (Gibson, 1977) which are considered important representational abstractions in biological staged development. Figure 1 shows the approach.

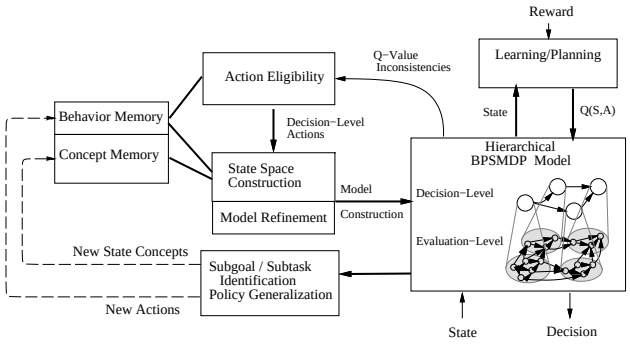

Figure 1. System Overview of the Approach for Hierarchical Behavior and State Concept Transfer.

Here, the agent maintains and incrementally augments a skill hierarchy and a corresponding representation hierarchy. When a new task is presented, the agent forms actions sets, including one containing actions which are considered relevant and one with all actions not deemed redundant. It then constructs a hierarchy of state representations from the goal and probabilistic *affordance* concepts associated with these skill sets. The state representations are formed here within the framework of Bounded Parameter Markov Decision Processes (BPMDPs) (Kim & Dean, 2003) and include a decision-level model and a more complex evaluation-level model. Learning of the new task is then performed on the decision-level model using Q-learning, while a second value function is maintained on the evaluation-level model. When an inconsistency is discovered between the two value functions, a refinement augments the decision-level model by including the concepts of the action that led to the inconsistency.

Once a policy for the new task is learned, subgoals are extracted from the system model and corresponding subgoal skills are learned off-line. Then goal and probabilistic *affordance* concepts are learned for the new subgoal skills and both, the new skills and concepts are included into the skill and representation hierarchies in the agent's memory, making them available for subsequent learning tasks.

2.1. Learning Transferable Skills

To learn skills for transfer, the approach presented here tries to identify subgoals. Subgoals of interest here are states that have properties that could be useful for subsequent learning tasks. Because the new tasks' requirements, and thus their reward functions, are generally unknown, the subgoal criterion used here does not focus on reward but rather on local properties of the state space in the context of the current task domain. In particular, the criterion used attempts to identify states which locally form a significantly stronger "attractor" for state space trajectories as measured by the relative increase in visitation likelihood.

To find such states, the subgoal discovery method first generates N random sample trajectories from the learned policy and for each state, s, on these trajectories determines the expected visitation likelihood, $C^*_H(s)$. The change of visitation likelihoods along a sample trajectory, h_i, is then determined as $\Delta_H(s_t) = C^*_H(s_t) - C^*_H(s_{t-1})$, where s_t is the t^{th} state along the path. The ratio of this change along the path is then computed as

$$\frac{\Delta_H(s_t)}{\max(1, \Delta_H(s_{t+1}))}$$

for every state in which $\Delta_H(s_t) > 0$. Finally, a state s_t is considered a potential subgoal if its average change ratio is significantly greater than expected from the distribution of the ratios for all states [1]. For all subgoals found, corresponding policies are learned off-line as SMDP option, o_i, and added to the skill hierarchy.

2.2. Learning Functional Descriptions of State

The power of complex actions to improve learning performance has two main sources; (i) their use reduces the number of decision points necessary to learn a policy, and (ii) they usually permit learning to occur on a more compact state representation. To harness the latter, it is necessary to automatically derive abstract state representations that capture the functional characteristics of the actions. To do so, the presented approach builds a hierarchical state representation within the basic framework of BPMDPs extended to SMDPs, forming a hierarchical Bounded Parameter SDMP (BPSMDP). Model construction occurs in a multi-stage, action-dependent fashion, allowing the model to adapt rapidly to action set changes.

The BPSMDP state space is a partition of the original state space where the following inequalities hold for all blocks (BPSMDP states) B_i and actions o_j:

[1] The threshold is computed automatically using a t-test based criterion and a significance threshold of 2.5%.

$$\left| \sum_{s'' \in B_j} F(s''|s, o_i) - \sum_{s'' \in B_j} F(s''|s', o_i) \right| \leq \delta \quad (1)$$

$$|R(s, o_i) - R(s', o_i)| \leq \epsilon \quad (2)$$

where $R(s, o)$ is the expected reward for executing option o in state s, and $F(s'|s, o)$ is the discounted transition probability for option o initiated in state s to terminate in state s'. These properties of the BPSMDP model ensure that the value of the policy learned on this model is within a fixed bound the optimal policy value on the initial model, where the bound is a function of ϵ and δ (Kim & Dean, 2003).

To make the construction of the BPSMDP more efficient, the state model is constructed in multiple steps. First functional concepts for each option, o, are learned as termination concepts $C_{t,o}$, indicating the option's goal condition, and probabilistic prediction concepts ("affordances"), $C_{p,o,x}$, indicating the context under the option will terminate successfully with probability $x \pm \epsilon$. These conditions guarantee that any state space utilizing these concepts in its state factorization fulfills the conditions of Equation 1 for any single action.

To construct an appropriate BPMDP for a specific action set $O_t = \{o_i\}$, an initial model is constructed by concatenating all concepts associated with the options in O_t. Additional conditions are then derived to achieve the condition of Equation 1 and, once reward information is available, the reward condition of Equation 2. This construction facilitates efficient adaptation to changing action repertoires.

To further utilize the power of abstract actions, a hierarchy of BPSMDP models is constructed here where the decision-level model utilizes the set of options considered necessary while the evaluation-level uses all actions not considered redundant. In the current system, a simple heuristic is used where the decision-level set consists only of the learned subgoal options while the evaluation-level set includes all actions.

2.3. Learning on a Hierarchical State Space

To learn new tasks, Q-learning is used here at the decision-level of the BPSMDP hierarchy. Because the compact decision-level state model encodes only the aspects of the environment relevant to a subset of the actions, it only ensures the learning of a policy within the pre-determined optimality bounds if the policy only utilizes the actions in the decision-level action set. Since, however, the action set has to be selected without knowledge of the new task, it is generally not possible to guarantee that it contains all required actions. To address this, the approach maintains a second value function on top of the evaluation-level system model. While decisions are made strictly based on the decision-level states, the evaluation-level value function is used to discover value inconsistencies, indicating that a significant aspect of the state space is not represented in the evaluation-level state model. The determination of inconsistencies here relies on the fact that the optimal value function in a BPMDP, V_P^*, is within a fixed bound of the optimal value function, V^*, on the underlying MDP (Kim & Dean, 2003).

Inconsistencies are discovered when the evaluation-level value for a state significantly exceeds the value of the corresponding state at the decision level. In this case, the action producing the higher value is included for the corresponding block at the decision level and the block is refined with this action to fulfill Equations 1 and 2 as illustrated in Figure 2.

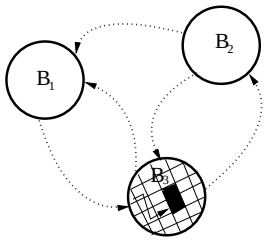

Figure 2. Decision-level model with 3 initial blocks (B_1, B_2, B_3) where block B_3 has been further refined.

As a result, the system is capable of adjusting its state representation on-line to ensure that a policy can be learned which is within a bound of the optimal policy.

3. Experiments

To evaluate the approach, it has been implemented on the Urban Combat Testbed (UTC), a computer game. For the experiments presented here, the agent is given the abilities to move through the environment shown in Figure 3 and to retrieve and deposit objects.

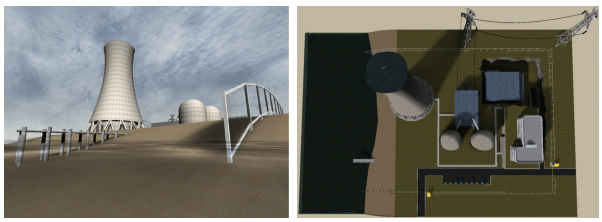

Figure 3. Urban Combat Testbed (UCT) domain.

The state is here characterized by the agent's pose as well as by a set of local object percepts, resulting in an effective state space with $20,000$ states.

The agent is first presented with a reward function to

learn to move to a specific location. Once this task is learned, subgoals are extracted by generating random sample trajectories as shown in Figure 4.

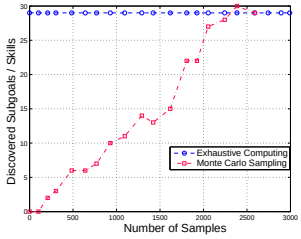

Figure 4. Number of subgoals discovered using sampling.

As the number of samples increases, the system identifies an increasing number of subgoals until, after $2,000$ samples, all 29 subgoals that could be found using exhaustive calculation have been captured.

Once subgoals are extracted, subgoal options, o_i, are learned and termination concepts, C_{t,o_i} and probabilistic outcome predictors, $C_{p,o_i,x}$ are generated using a genetic-algorithm based classifier learner. These subgoal options and the termination and prediction concepts are then transferred to the next learning tasks.

The system then builds a hierarchical BPSMDP system model where the decision-level only utilizes the learned subgoal actions while the evaluation-level model is built for all available actions. On this model, a second task is the learned where the agent is rewarded for retrieving a flag (from a different location than the previous goal) and return it to the home base. During learning, the system augments its decision-level state representation to allow learning of a policy that is within a bound of optimal as shown in Figure 5.

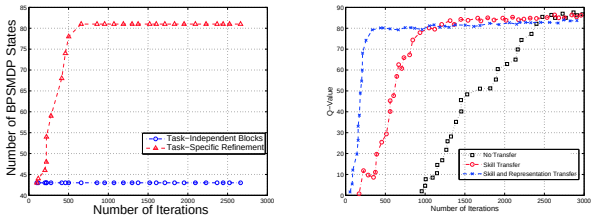

Figure 5. Size of the decision-level state representation (left) and learning performance with and without skill and representation/concept transfer (right).

The left graph here shows that the system starts with an initial state representation containing 43 states. During learning, as value function inconsistencies are found, new actions and state splits are introduced, eventually increasing the decision-level state space to 81 states. On this state space, a bounded optimal policy is learned as indicated in the right graph. This graph compares the learning performance of the system against a learner that only transfers the discovered subgoal options and a learner without any transfer mechanism. These graphs show a transfer ratio[2] of ≈ 2.5 when only subgoal options are transferred, illustrating the utility of the presented subgoal criterion. Including the representation transfer and hierarchical BPSMDP learning approach results in significant further improvement with a transfer ratio of ≈ 5.

4. Conclusion

Most artificial learning agents suffer from the inefficient re-use of acquired control knowledge in artificial. To address this deficiency, the learning approach presented here provides a mechanism which extracts and transfers control knowledge in the form of potentially useful skill and corresponding representation concepts to improve the learning performance on subsequent tasks. The transferred knowledge is used to construct a compact state space hierarchy that captures the important aspects of the environment in the context of the agent's capabilities and thus results in significant improvements in learning performance. Initial experiments in a video game domain have demonstrated the benefit of the presented mechanism.

References

Ando, R. K., & Zhang, T. (2004). *A framework for learning predictive structures from multiple tasks and unlabeled data* (Tech. Rep. RC23462). IBM.

Barto, A., & Mahadevan, S. (2003). Recent Advances in Hierarchical Reinforcement Learning. *Discrete Event Dynamic Systems*, *13*, 341–379.

Gibson, J. (1977). The theory of affordances. In *Perceiving, acting and knowing*. Erlbaum.

Kim, K., & Dean, T. (2003). Solving Factored MDPs using Non-Homogeneous Partitions. *AI 147*, 225.

Marthi, B., Russell, S., Latham, D., & Guestrin, C. (2005). Concurrent hierarchical reinforcement learning. *IJCAI-05*. Edinburgh, Scotland.

Marx, Z., Rosenstein, M. T., & Kaelbling, L. P. (2005). Transfer leraning with an ensemble of background tasks. *NIPS Transfer Learning*. Whistler, Canada.

Sutton, R., Precup, D., & Singh, S. (1999). Between MDPs and Semi-MDPs. *AI 112*, 181–211.

Taylor, M. E., & Stone, P. (2005). Behavior transfer for value-function-based reinforcement learning. *AAMAS 2005* (pp. 53–59).

[2]The transfer ratio is the ratio of the area over the learning curve between the no-transfer and the transfer learner.