JMLR: Workshop and Conference Proceedings 24:59–75, 2012 10th European Workshop on Reinforcement Learning

Directed Exploration in Reinforcement Learning with Transferred Knowledge

Timothy A. Mann MANN23@TAMU.EDU
Yoonsuck Choe CHOE@TAMU.EDU
Department of Computer Science & Engineering
Texas A&M University

Editor: Marc Peter Deisenroth, Csaba Szepesvári, Jan Peters

Abstract

Experimental results suggest that transfer learning (TL), compared to learning from scratch, can decrease exploration by reinforcement learning (RL) algorithms. Most existing TL algorithms for RL are heuristic and may result in worse performance than learning from scratch (i.e., negative transfer). We introduce a theoretically grounded and flexible approach that transfers action-values via an intertask mapping and, based on those, explores the target task systematically. We characterize positive transfer as (1) decreasing sample complexity in the target task compared to the sample complexity of the base RL algorithm (without transferred action-values) and (2) guaranteeing that the algorithm converges to a near-optimal policy (i.e., negligible optimality loss). The sample complexity of our approach is no worse than the base algorithm's, and our analysis reveals that positive transfer can occur even with highly inaccurate and partial intertask mappings. Finally, we empirically test directed exploration with transfer in a multijoint reaching task, which highlights the value of our analysis and the robustness of our approach under imperfect conditions.

Keywords: Reinforcement Learning, Transfer Learning, Directed Exploration, Sample Complexity

1. Introduction

Transfer learning (TL) applied to reinforcement learning (RL) exploits knowledge gained while interacting with source tasks to learn faster in a target task [Lazaric, 2008; Taylor and Stone, 2009]. When TL is successful it is referred to as positive transfer, and when TL fails it is referred to as negative transfer. Previous research applying TL to RL has demonstrated its effectiveness in decreasing learning time [Selfridge et al., 1985; Taylor et al., 2007; Fernández et al., 2010], but there is a lack of theoretical understanding about when TL applied to RL will succeed or fail [Taylor and Stone, 2011]. Most TL+RL algorithms are heuristic in nature. This is in contrast with provably efficient single task RL algorithms such as R-MAX [Brafman and Tennenholtz, 2002; Kakade, 2003] and Delayed Q-learning [Strehl et al., 2006]. Lazaric and Restelli [2011] – a notable exception – has analyzed the sample complexity of TL from a batch RL perspective, but their work does not address exploration in the target task.

To learn efficiently, RL algorithms must balance between exploitation (executing a sequence of actions the algorithm has already learned will provide high long-term rewards) and exploration (trying uncertain actions to gain more information about the environment).

Exploration strategies can be broadly categorized as either undirected exploration or directed exploration [Thrun, 1992]. Undirected exploration is characterized by local, random selection of actions, while directed exploration, by contrast, uses global information to systematically determine which action to try. ϵ-greedy represents the most popular undirected exploration strategy, while "optimism in the face of uncertainty" (OFU) represents the most popular directed exploration strategy. OFU initially assumes that all actions result in higher value than may be true in the environment and greedily selects the action believed to give the highest reward (breaking ties arbitrarily). When the agent tries an action it samples its value distribution. If the sampled values are lower than expected, the algorithm lowers the action's estimated value and switches to another (possibly overestimated) action. Otherwise the algorithm sticks with its current action. In this way, the algorithm eventually settles on a nearly optimal action at every state (which results in a near-optimal policy [Kakade, 2003, Theorem 3.1.1]).

Previous research on TL+RL has almost exclusively studied undirected exploration [Taylor et al., 2007; Fernández et al., 2010] or batch transfer [Lazaric, 2008]. Although the sample complexity of exploration has been examined in the literature, to our knowledge, no previous work has formally analyzed sample complexity of exploration in the considerably more complex case of action-value transfer. Analyzing TL is more complicated because there are multiple learning algorithms and tasks that need to be considered. Our main innovation is to show how TL can be analyzed from a sample complexity perspective. We analyze action-value transfer via intertask mappings [Taylor and Stone, 2011] paired with the provably efficient Delayed Q-learning algorithm [Strehl et al., 2006] that uses the OFU exploration strategy to learn faster in the target task while avoiding optimality loss (i.e., positive transfer). Our approach has several advantages compared to previous TL approaches:

1. We can theoretically analyze our TL approach because we precisely define positive transfer in terms of sample complexity and optimality loss. The sample complexity of our approach can be compared to the sample complexity of single task RL algorithms (with respect to the target task).

2. Our analysis reveals that positive transfer can occur even when the distance between the optimal target task action-values and the transferred action-values is large.

3. Intertask mappings enable transfer between two tasks with different state-action spaces or when only a partial intertask mapping can be derived.

The rest of this paper is organized as follows: In section 2, we provide background on RL and TL. In section 3, we introduce α-weak admissible heuristics, which we use in section 4 to analyze the sample complexity and optimality loss of TL+RL. In section 5, we demonstrate the advantage of applying directed exploration to TL. In section 6 we discuss advantages and limitations of our approach and conclude in section 7.

2. Background

A Markov decision process (MDP) M is defined by a 5-tuple $\langle S, A, T, R, \gamma \rangle$ where S is a set of states, A is a set of actions, T is a set of transition probabilities $\Pr[s'|s,a]$ determining the probability of transitioning to a state $s' \in S$ immediately after selecting action $a \in A$

while in state $s \in S$, $R : S \times A \to \mathbb{R}$ assigns scalar rewards to state-action pairs, and γ is a discount factor that reduces the value of rewards distant in the future [Sutton and Barto, 1998]. We assume the transition probabilities T and the reward function R are unknown. The objective of most RL algorithms is to find a policy $\pi : S \to A$ that maximizes

$$Q^{\pi}_{M}(s_t, a_t) = E\left[R(s_t, a_t) + \sum_{\tau=t+1}^{\infty} \gamma^{\tau-t} R(s_\tau, \pi(s_\tau))\right] , \quad (1)$$

the action-value for (s_t, a_t), which is the discounted, expected sum of future rewards from time t forward [Sutton and Barto, 1998]. The value function $V^{\pi}_{M}(s) = \max_{a \in A} Q^{\pi}_{M}(s, a)$. We denote the optimal value and action-value functions by V^{*}_{M} and Q^{*}_{M} (respectively) and assume that the reward function is bounded to the interval $[0, 1]$. Therefore, $0 \leq V^{*}_{M}(s) \leq \frac{1}{1-\gamma}$ for all states $s \in S$.

Sample complexity enables theoretical comparison between RL algorithms by measuring the number of samples required for an RL algorithm to achieve a learning objective. Given any MDP M, $\epsilon > 0$, and $\delta \in (0, 1]$, the *sample complexity of exploration* of an RL algorithm $\mathcal{A}$ is the number of timesteps t, such that, with probability at least $1 - \delta$,

$$V^{\mathcal{A}_t}_{M}(s_t) < V^{*}_{M}(s_t) - \epsilon \quad (2)$$

where s_t is the state and $V^{\mathcal{A}_t}_{M}$ denotes the value of $\mathcal{A}$'s policy at timestep t [Kakade, 2003]. Sample complexity that is polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, $\frac{1}{1-\gamma}$, $N = |S|$, and $K = |A|$ is considered efficient. Algorithms that are provably efficient with respect to sample complexity of exploration are called PAC-MDP [Strehl et al., 2009].

A key component of algorithms with polynomial sample complexity is directed exploration. Whitehead [1991] demonstrated conditions where undirected exploration leads to exponential sample complexity, with respect to the number of states. Provably efficient algorithms such as R-MAX [Brafman and Tennenholtz, 2002; Kakade, 2003] and Delayed Q-learning [Strehl et al., 2006] use the OFU directed exploration strategy. Although the analysis in this paper can be extended to other provably efficient single task RL algorithms, we focus on Delayed Q-learning (DQL) for clarity. DQL has two parameters m and ϵ_1, where m controls the number of samples used during each update of a state-action pair and ϵ_1 controls how close to optimal the learned policy should be.

Previous research on TL+RL has primarily considered undirected exploration strategies [Taylor et al., 2007; Fernández et al., 2010] or batch RL, which does not consider the problem of exploration [Lazaric, 2008; Lazaric and Restelli, 2011]. If the transferred knowledge is not similar enough to any near-optimal policy, undirected exploration can lead to exponential sample complexity due to the same reasons it fails in single task RL. In this paper, we consider the transfer of action-values, which has been demonstrated to be effective in experiments [Selfridge et al., 1985; Taylor et al., 2007] with directed exploration. Before explaining our TL setup, we will find it useful to define a new structure called a weak admissible heuristic.

3. Action-value Initialization with Weak Admissible Heuristics

A good guess of the initial action-values can be useful in speeding up RL. A function $U : S \times A \to \mathbb{R}$ is an admissible heuristic if $Q^{*}(s, a) \leq U(s, a) \leq \frac{1}{1-\gamma}$ for all $(s, a) \in S \times A$.

Strehl et al. [2009] demonstrated that if the action-values are smaller than $\frac{1}{1-\gamma}$, then this initialization can decrease the sample complexity of exploration while maintaining PAC-MDP guarantees. Admissible heuristics provide valuable prior knowledge to PAC-MDP RL algorithms, but the specified prior knowledge does not need to be exact. This property will be useful in a TL setting for two reasons: (1) knowledge is estimated from a source task and (2) there is rarely an exact relationship between source and target tasks.

The admissible heuristic used by Strehl et al. [2009] is more restrictive than necessary. For example, Figure 3.1 shows an example where some of the initial action-values' estimates are below their corresponding optimal values, yet, following a simple OFU exploration strategy will converge to the near-optimal action b_2. Consider what would happen if the OFU exploration strategy is run on the example in Figure 3.1. First, the algorithm's initial policy would select action b_6 because the heuristic value (dotted box) is the highest. After selecting b_6 several times an update would occur decreasing the value associated with b_6 because its true value is much lower than the estimated value. Now action b_2 would be selected because it has the second highest estimated value and this estimate is very unlikely to drop below the value of any other action. So the algorithm would converge on the near-optimal action b_2. For our analysis of TL+RL, we would like to derive an extremely weak structure that will help characterize when TL will succeed and when it will fail. To help with this, we define the concept of a weak admissible heuristic.

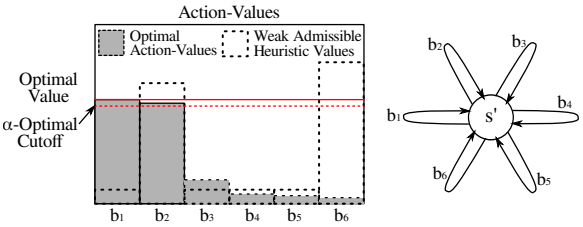


Figure 3.1: Weak admissible heuristic applied to a one-state task with six actions. The weak admissible heuristic only needs to optimistically initialize the action-value for a single nearly optimal action.

Definition 1 *A function $W : S \times A \to \mathbb{R}$ is an* **α-weak admissible heuristic** *(or just weak admissible heuristic) for MDP $M = \langle S, A, T, R, \gamma \rangle$, if for each $s \in S$, there exists $\tilde{a} \in A$ such that*

$$V^*(s) - \alpha \leq Q^*(s, \tilde{a}) \leq W(s, \tilde{a}) \leq \frac{1}{1-\gamma} \tag{3}$$

where α is the smallest non-negative value satisfying this inequality.

In other words, W is a weak admissible heuristic if at every state $s \in S$ at least one α-optimal action is mapped to an optimistic value, even though other action-values (including the optimal action) may be severely pessimistic. This is consistent with the situation in Figure 3.1, where an OFU exploration strategy converges to the near-optimal action b_2, with high probability, and not the optimal action b_1. In this case, the weak admissible heuristic implicitly eliminated several actions (including the optimal action b_1). This helped

reduce sample complexity because only one suboptimal action (b_6) was ever explored, and the algorithm still converges to a near-optimal policy.

Although weak admissible heuristics are considerably less constrained than the notion of admissible heuristic employed by Strehl et al. [2009], initializing DQL with a weak admissible heuristic (with small enough α) is sufficient to maintain PAC-MDP guarantees. For our analysis, the critical aspect of weak admissible heuristics is that they implicitly eliminate some state-action pairs from consideration (Lemma 1 below). This improves the efficiency of directed exploration because there are fewer state-action pairs to explore.

Lemma 1 *(State-action Pair Elimination) Let $\eta \geq 0$, W be an α-weak admissible heuristic for an MDP M, and $\mathcal{A}$ is a value-based RL algorithm with initial action-value estimates $\hat{Q}_0 = W$ such that for all timesteps $t \geq 1$, (1) $\mathcal{A}$ follows a greedy policy ($\mathcal{A}_t(s) = \arg\max_{a \in A} \hat{Q}_t(s,a)$), (2) Updates to $\hat{Q}_t(s,a)$ can only occur if (s,a) has been visited, and (3) If $W(s,a) \geq Q^*(s,a)$, then $\hat{Q}_t(s,a) \geq Q^*(s,a) - \eta$, then for all $(s,a) \in S \times A$ where $W(s,a) < V^*(s) - (\alpha + \eta)$, $\mathcal{A}$ will never explore (s,a) ($\mathcal{A}_t(s) \neq a$ at any timestep $t \geq 1$, where $\mathcal{A}_t$ denotes the policy of $\mathcal{A}$ at timestep t).*

Proof Since $\mathcal{A}$ follows a greedy policy with respect to $\hat{Q}_t$, a state-action pair (s,a) will only be selected if $\hat{Q}_t(s,a) = \max_{a' \in A} \hat{Q}_t(s,a')$. The proof is by induction on the timestep t. Suppose, without loss of generality, that $W(s,a) < V^*(s) - (\alpha + \eta)$.

Base Case ($t = 0$): By the definition of W there exists $\tilde{a}$ such that $W(s,\tilde{a}) \geq Q^*(s,\tilde{a}) \geq V^*(s) - \alpha > W(s,a)$. Thus $a \neq \arg\max_{a' \in A} W(s,a') = \arg\max_{a' \in A} \hat{Q}_0(s,a')$.

Induction Step: By assumption 3 the action-value estimate for $\hat{Q}_t(s,\tilde{a}) \geq Q^*(s,\tilde{a}) - \eta \geq (V^*(s) - (\alpha + \eta)) > W(s,a)$. Since (s,a) has not been tried yet no update to its action-value could have occurred (assumption 2), and will not be executed on timestep $t+1$ because $\hat{Q}_t(s,a) = W(s,a) < \hat{Q}_t(s,\tilde{a})$.

Therefore by induction, (s,a) will never be executed. ■

If W is a weak admissible heuristic, then Lemma 1 says that low valued state-action pairs will never be explored by the algorithm's policy. Condition 3 guarantees that the algorithm will never underestimate action-values (by more than η) that are initially optimistic. This combined with the weak admissible heuristic assumption allows us to guarantee that an α-optimal action will eventually be the action selected by the algorithm's greedy policy. DQL satisfies the algorithmic requirements of Lemma 1, with high probability.

Theorem 2 *Let $\alpha \geq 0$, $\epsilon > 0$, $\delta \in (0,1]$, and W be an α-weak admissible heuristic with respect to $M = \langle S, A, T, R, \gamma \rangle$. There exists $\epsilon_1 = O(\epsilon(1-\gamma))$ and $m = O\left(\frac{\ln(NK/\delta)}{\epsilon_1^2(1-\gamma)^2}\right)$, such that if $\mathcal{A}$ is an instance of the DQL algorithm initialized with parameters ϵ_1 and m, then $V^{\mathcal{A}_t}(s_t) \geq V^*(s_t) - (\epsilon + \frac{\alpha}{1-\gamma})$ on all but*

$$O\left(\frac{NK - X}{\epsilon^4(1-\gamma)^8} \ln\frac{1}{\delta} \ln\frac{1}{\epsilon(1-\gamma)} \ln\frac{NK}{\delta\epsilon(1-\gamma)}\right) \tag{4}$$

timesteps t, with probability at least $1-\delta$, where $X = \left|\left\{(s,a) \in S \times A \mid W(s,a) < V^*(s) - \alpha - \frac{\alpha}{1-\gamma}\right\}\right|$.

The proof of this theorem is similar to [Strehl et al., 2006, Theorem 1], which provides the following sample complexity bound for DQL without transferred knowledge:

$$O\left(\frac{NK}{\epsilon^4(1-\gamma)^8}\ln\frac{1}{\delta}\ln\frac{1}{\epsilon(1-\gamma)}\ln\frac{NK}{\delta\epsilon(1-\gamma)}\right)$$

Our proof relies on Lemma 1 to demonstrate its dependence on $NK - X \leq NK$. See the Appendix for a proof of Theorem 2.

If $X > 0$ and α is small, then the sample complexity bound depends on $\tilde{O}(NK - X)$, which is smaller than the lower bound for any single task RL algorithm $\Omega(NK)$ with respect to the number of states and actions Strehl et al. [2009]. If $X = 0$ and $\alpha = 0$, then we restore the upper bound from [Strehl et al., 2006, Theorem 1]. Therefore, a weak admissible heuristic can help to decrease the sample complexity of exploration.

4. Transferring a Weak Admissible Heuristic

We use the concept of a weak admissible heuristic to analyze action-value transfer with the objective of learning to act near-optimally in the target task with sample complexity of exploration (Eq. 2) that is smaller than DQL without transferred knowledge. We denote the source task/MDP by M_{src} and the target task/MDP by M_{trg}. Consider the situation in Figure 4.1. First the agent learns action-values for the source task. Next, because the source task and the target task have a different number of actions, a function h called an intertask mapping (defined below) is used to relate action-values from the source task to the target task. Finally, notice that in Figure 4.1 the transferred action-values satisfy a weak admissible heuristic. In this section, we explore assumptions about the intertask mapping needed to ensure that the transferred action-values satisfy a weak admissible heuristic and how transfer influences sample complexity of exploration in the target task.

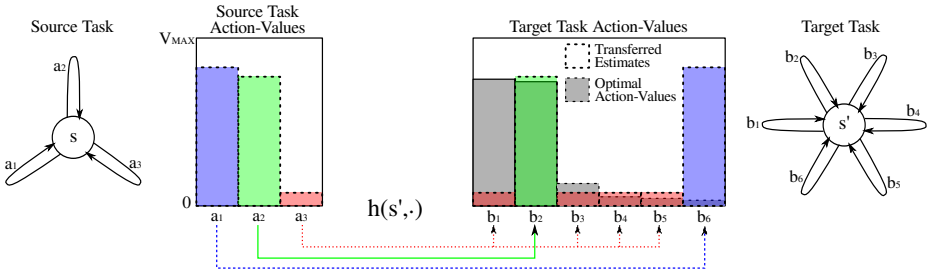


Figure 4.1: Transfer from a one-state source task with three actions to a one-state target task with six actions. Despite the transferred action-values severely underestimating the optimal action b_1 and severely overestimating the lowest valued action b_6, an OFU exploration strategy can still converge to a near-optimal policy (i.e., b_2).

There are two factors that affect positive transfer: (1) sample complexity and (2) optimality loss. Positive transfer occurs when the sample complexity of exploration in the target task is lower than the sample complexity of the base RL algorithm and no optimality loss

has occurred. Optimality loss occurs when transferred knowledge causes an RL algorithm to converge to a suboptimal policy along its current trajectory.

Typically, access to samples of the source task is less "expensive" than access to samples from the target task. For the purposes of this paper, we assume unrestricted access to a generative model for M_{src}. Using, for example, the PhasedValueIteration algorithm [Kakade, 2003] it is possible to learn arbitrarily accurate source task action-values with arbitrarily high confidence with polynomially many samples. Therefore, we will assume that the estimated source task action-values $\hat{Q}_{\text{src}}$ are ϵ_{src}-accurate.

If M_{src} and M_{trg} have different state-action spaces, then an intertask mapping $h : D \to S_{\text{src}} \times A_{\text{src}}$ is needed, where $D \subseteq S_{\text{trg}} \times A_{\text{trg}}$, to relate a subset of state-action pairs from the target task to state-action pairs in the source task. We assume that if $(s, a) \in D$, either there exists $(s, \tilde{a}) \in D$ such that

$$V^*_{\text{trg}}(s) - \alpha \leq Q^*_{\text{trg}}(s, \tilde{a}) \leq Q^*_{\text{src}}(h(s, \tilde{a})) \ , \tag{5}$$

which is analogous to (3) with $W(s, a) = Q^*_{\text{src}}(h(s, a))$ or there exists $(s, \tilde{a}) \notin D$ such that

$$V^*_{\text{trg}}(s) - \alpha \leq Q^*_{\text{trg}}(s, \tilde{a}) \tag{6}$$

in which case we can assign the value $W(s, \tilde{a}) = \frac{1}{1-\gamma}$. To transfer action-values we use

$$W(s, a) = \begin{cases} \min\left(\hat{Q}_{\text{src}}(h(s, a)) + \epsilon_{\text{src}}, \frac{1}{1-\gamma}\right) & \text{if } (s, a) \in D \\ \frac{1}{1-\gamma} & \text{otherwise} \end{cases} \tag{7}$$

to set initial action-value estimates given an intertask mapping h, and ϵ_{src}-accurate source task action-value estimates $\hat{Q}_{\text{src}}$. At every state at least one nearly optimal action is mapped to an action-value which overestimates the true action-value or not mapped at all. If a state-action pair is not in the domain D, then we assign the maximum possible value to ensure it is optimistically initialized. Under these assumptions the transferred action-values are an α-weak admissible heuristic.

Theorem 3 *Let $\epsilon > 0$, $\epsilon_{\text{src}} > 0$, $\delta \in (0, 1]$, $h : D \to S_{\text{src}} \times A_{\text{src}}$ be an intertask mapping from a subset of state-action pairs in M_{trg} to M_{src} satisfying (5) and (6), and $\hat{Q}_{\text{src}}$ are ϵ_{src}-accurate action-value estimates for M_{src}. There exists $\epsilon_1 = O(\epsilon(1-\gamma))$ and $m = O\left(\frac{\ln(NK/\delta)}{\epsilon_1^2(1-\gamma)^2}\right)$ such that if an instance $\mathcal{A}$ of the DQL algorithm with ϵ_1, m, and action-value estimates initialized by Eq. (7) is executed on M_{trg}, then $V^{\mathcal{A}_t}_{\text{trg}}(s) < V^*_{\text{trg}}(s) - (\epsilon + \frac{\alpha}{1-\gamma})$ occurs on at most*

$$O\left(\frac{NK - Y}{\epsilon^4(1-\gamma)^8} \ln\frac{1}{\delta} \ln\frac{1}{\epsilon(1-\gamma)} \ln\frac{NK}{\delta\epsilon(1-\gamma)}\right)$$

timesteps t, with probability at least $1 - \delta$, where $N = |S|$, $K = |A|$, and

$$Y = \left|\left\{(s, a) \in D \mid \hat{Q}_{\text{src}}(h(s, a)) < V^*_{\text{trg}}(s) - (\alpha + \frac{\alpha}{1-\gamma} + \epsilon_{\text{src}})\right\}\right|$$

is the number of state-action pairs that will never be explored.

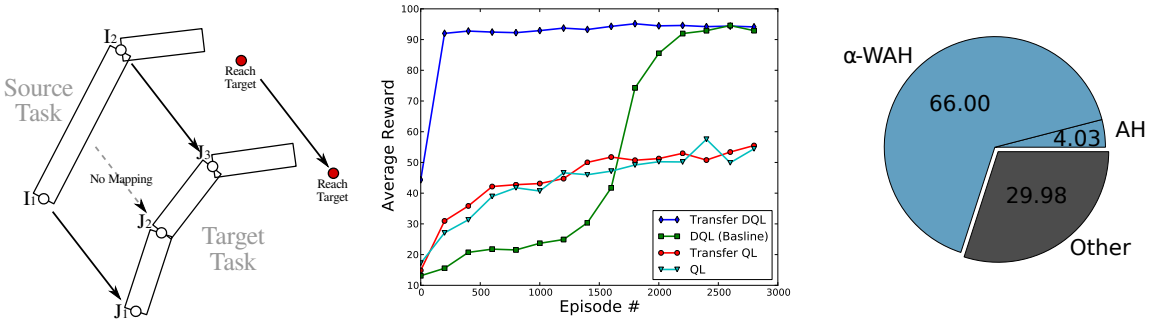
See Appendix for a proof of Theorem 3. The main importance of Theorem 3 is that we have reduced the analysis of action-value transfer to the analysis of learning with an α-weak admissible heuristic. Here, α controls optimality loss, and we can think of α (or $\alpha/(1-\gamma)$) as the error introduced by the intertask mapping h. If $\alpha \approx 0$ compared to ϵ, then there is little or no optimality loss compared to learning from scratch. In many cases, $\alpha = 0$ can be achieved, for example, if W turns out to be an admissible heuristic [Strehl et al., 2009]. However, if α is large, then the result of TL is likely to be poor in the worst case. Similar to X in Theorem 2, when Y is large the sample complexity of exploration in the target task decreases significantly compared to learning from scratch with DQL. Notice, however, that the sample complexity is never worse than learning from scratch. The main consideration should be identifying an intertask mapping that does not introduce much optimality loss. Thus, TL is characterized by optimality loss and sample complexity, and Theorem 3 helps to clarify this relationship.

5. Experiments and Results

Our experimental tasks were inverse kinematics problems (Figure 5.1a). In the source task M_{src}, the agent controlled a two-joint mechanical arm guiding its end-effector to one of four reach target locations. In the target task M_{trg}, the agent controlled a three-joint mechanical arm with noisy actuators by moving its end-effector to one of four reach target locations. In both tasks the state was represented by the target index and the joint locations. The actions encode whether to rotate each joint and in which direction. Actions only perturbed joints by a small amount. So a sequence of actions was needed to complete each task. In the target task there was a small probability (0.2) that the action applied to a joint would either leave the joint unchanged or overshoot the desired location. Because of the different number of joints, there is no one-to-one mapping between the state-action space of the source task and the state-action space of the target task. We defined an intertask mapping that relates the reach target index, and joints J_1 to I_1 and J_3 to I_2, leaving out joint J_2.

We compared DQL, which uses directed exploration, with and without transferred knowledge, and Q-learning (QL) using an ε-greedy exploration strategy with and without transferred knowledge. Figure 5.1b demonstrates that the transferred action-values enable DQL to quickly converge to a near-optimal solution, while DQL (with the same parameters) learning from scratch takes much longer to learn a good policy. Both QL conditions perform worse than the DQL conditions because local exploration is not efficient in the target task's large state-action space. The negligible difference between QL and Transfer QL is due to the fact that the transferred action-values are a very poor approximation to the optimal action-values.

DQL and Transfer DQL spend most of their timesteps in a small number of states (less than 50 out of 1,372). Figure 5.1c shows the average proportion of highly visited states where the transferred action-values satisfy the admissible heuristic (AH) and α-weak admissible heuristic criteria with $\alpha = 0.1$. Most of the transferred action-values at these highly visited states fall under α-WAH and AH conditions. Note that AH $\subseteq$ α-WAH. A small proportion do not satisfy α-WAH (i.e., Other). For the states satisfying α-WAH, we found that the policy learned by Transfer DQL selected good action choices (i.e., $Q^*_{\text{trg}}(s, \pi(s)) \geq V^*_{\text{trg}}(s) - \alpha$) almost 100% of the time. Interestingly, Transfer DQL performs well despite the fact that



(a) (b) (c)

Figure 5.1: (a) Transfer between inverse kinematic tasks where the source task controls a two-joint arm, and the target task controls a three-joint arm. (b) Average reward over 3,000 episodes in the target task for Delayed Q-learning without transfer (DQL (Baseline)), Delayed Q-learning with transferred action-values (Transfer DQL), Q-learning with transferred action-values (Transfer QL), and Q-learning without transfer (QL). Transfer DQL learns much more quickly than the compared algorithms. (c) Percentage of transferred action-values (out of highly visited states) satisfying the admissible heuristic (AH) and α-weak admissible heuristic (α-WAH) criteria. Note: AH $\subseteq$ α-WAH.

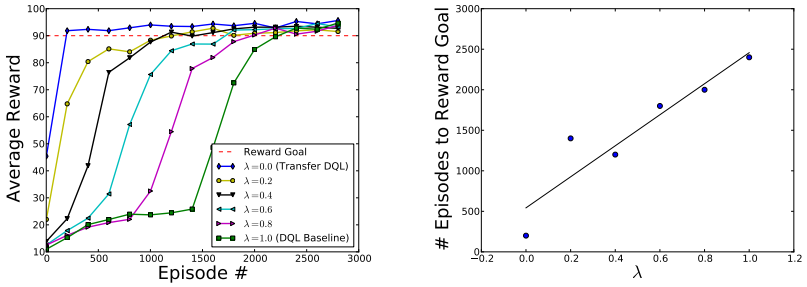
some of the highly visited states do not satisfy the α-WAH condition. This demonstrates the robustness of our TL approach in practice and is an important observation since we would not expect in practice to be able to rigerously guarantee (5) and (6) for all states. These result suggest that the weak admissible heuristic concept is important for understanding when transfer learning succeeds. On the other hand, in practice, guaranteeing that the transferred action-values satisfy the weak admissible heuristic criterion at every state is not necessary to achieve positive transfer.

We considered the impact of TL with partial intertask mappings by modifying the intertask mapping from the previous experiment. State-action pairs from its domain were randomly removed with probability λ. Figure 5.2 shows the average reward curve for Transfer DQL as λ is varied from 0.0 to 1.0. The training time has an approximately linear relationship with λ, increasing as λ increases.

6. Discussion

The main advantages of our approach are that it can be theoretically analyzed and that it works with several existing provably efficient algorithms, such as R-MAX, Modified R-MAX, and DQL. This approach may also work with future RL algorithms. We have shown that the transferred action-values do not need to be accurate (with respect to L_1-norm), and our analysis holds for MDPs with stochastic transitions and rewards.

The main limitation of our approach is that if the action-values do not satisfy a weak admissible heuristic with small α, then the final learned policy may be poor and the algorithm



(a) (b)

Figure 5.2: (*a*) Average reward for Transfer DQL as state-action pairs are removed from the intertask mapping's domain (with probability λ). (*b*) Increase in training time is approximately proportional to the probability that state-action pairs are removed.

will never recover. Another potential objection to our method is that there is no general way to obtain a good intertask mapping between tasks without solving both tasks. However, there are interesting special cases, for example, where an identity intertask mapping is used because the tasks only differ by their dynamics or reward structure.

7. Conclusion

We theoretically analyzed the combination of (1) action-value transfer via an intertask mapping with (2) directed exploration and found that the approach has several provable benefits. Only weak assumptions on the intertask mapping are necessary for positive transfer. Positive transfer can occur even when the distance between the transferred action-values and the optimal target task action-values is large. Although we have analyzed our TL approach using DQL, our analysis can be extended to other efficient algorithms such as R-MAX [Brafman and Tennenholtz, 2002] or Modified R-MAX [Szita and Szepesvári, 2010]. Finally, we demonstrated these advantages empirically on an inverse kinematics task. For states where the intertask mapping induced a weak admissible heuristic, DQL almost always learned to select a good action. Furthermore, our TL approach performed well even though a few highly visited states did not satisfy the weak admissible heuristic criteria, suggesting the approach is robust in practice. Although previous research has mostly paired undirected exploration with TL, we believe that directed exploration is an important mechanism for developing provably efficient TL+RL algorithms.

Acknowledgments

We acknowledge the EWRL reviewers and Matthew Taylor for their helpful feedback. This report is based in part on work supported by Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST) and the Texas A&M University Dissertation Fellowship.

References

Ronen I. Brafman and Moshe Tennenholtz. R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

Fernando Fernández, Javier Garcá, and Manuela Veloso. Probabilistic policy reuse for inter-task transfer learning. *Robotics and Autonomous Systems*, 58:866–871, 2010.

Sham Machandranath Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, March 2003.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232, 2002.

Alessandro Lazaric. *Knowledge Transfer in Reinforcement Learning*. PhD thesis, Politecnico Di Milano, 2008.

Alessandro Lazaric and Marcello Restelli. Transfer from multiple MDPs. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1746–1754, 2011.

Oliver G. Selfridge, Richard Sutton, and Andrew Barto. Training and tracking in robotics. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 670–672, 1985.

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In *Proceedings of the Twenty-third International Conference on Machine Learning (ICML-06)*, 2006.

Alexander L. Strehl, Lihong Li, and Michael Littman. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research*, 10:2413–2444, 2009.

Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

István Szita and Csaba Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

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

Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.

Matthew E. Taylor and Peter Stone. An introduction to inter-task transfer for reinforcement learning. *AI Magazine*, 32(1):15–34, 2011.

Sebastian Thrun. Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Carnegie-Mellon University, January 1992.

Steven D. Whitehead. A study of cooperative mechanisms for faster reinforcement learning. Technical report, University of Rochester, 1991.

Appendix

In this appendix we provide proofs for Theorems 2 and 3. Throughout this appendix we assume that M is an MDP defined by $\langle S, A, T, R, \gamma \rangle$ and W is an α-weak admissible heuristic for some $\alpha > 0$, unless otherwise noted. To prove Theorem 2 we need to introduce several concepts and lemmas.

7.1. PAC-MDP Framework

Analysis of PAC-MDP algorithms typically depends on the notion of an induced MDP [Kakade, 2003; Strehl et al., 2006, 2009; Szita and Szepesvári, 2010]. We introduce a modified version of the induced MDP that depends on how action-values are initialized. We assume that the RL algorithm maintains action-values $\hat{Q}_t$ initialized by $W = \hat{Q}_0$. State-action pairs with optimistically initialized action-values (i.e., $W(s,a) \geq Q^*_M(s,a)$) are treated separately from state-action pairs with pessimistically initialized action-values (i.e., $W(s,a) < Q^*_M(s,a)$). This allows us to essentially ignore pessimistically initialized state-action pairs and focus on optimistically initialized state-action pairs.

Definition 2 *Let $M = \langle S, A, T, R, \gamma \rangle$ be an MDP, $\kappa \subseteq S \times A$, and W be an α-weak admissible heuristic with respect to M. The* **induced MDP** *with respect to M and κ is denoted by $M_\kappa = \langle S \cup \{z_{s,a} \mid (s,a) \notin \kappa\}, A, T_\kappa, R_\kappa, \gamma \rangle$ where*

$$T_\kappa(s'|s,a) = \begin{cases} T(s'|s,a) & \text{if } (s,a) \in \kappa \text{ and } W(s,a) \geq Q^*_M(s,a) \\ 1 & \text{if } ((s,a) \notin \kappa \text{ and } s' = z_{s,a}) \text{ or } (s \in \{z_{s,a} \mid (s,a) \notin \kappa\} \text{ and } s' = s) \\ 0 & \text{otherwise} \end{cases}$$

and

$$R_\kappa(s,a) = \begin{cases} R(s,a) & \text{if } (s,a) \in \kappa \text{ and } W(s,a) \geq Q^*_M(s,a) \\ (1-\gamma)\hat{Q}(s,a) & \text{otherwise} \end{cases}$$

where T_κ defines the transitions probabilities and R_κ defines the reward function of M_κ.

The purpose of this more complex version of the induced MDP is to obtain the following action-value function. Suppose that a value-based RL algorithm is initialized with action-values $W = \hat{Q}_0$, π_t is the policy followed by the RL algorithm at timestep t and $\kappa_t \subseteq S \times A$, then the induced MDP M_{κ_t} has action-values defined by

$$Q^{\pi_t}_{M_{\kappa_t}}(s,a) = \begin{cases} R(s,a) + \gamma \sum_{s' \in S} V^{\pi}_{M_{\kappa_t}}(s') & \text{if } (s,a) \in \kappa_t \text{ and } W(s,a) \geq Q^*_M(s,a) \\ \hat{Q}_t(s,a) & \text{if } (s,a) \notin \kappa_t \text{ or } W(s,a) < Q^*_M(s,a) \end{cases} \tag{8}$$

for all $(s,a) \in S \times A$. The induced MDP represents an idealized version of what the RL algorithm has learned. The set κ represents the number of state-action pairs that have been experienced enough to be considered "well modeled". In the standard analysis of Delayed Q-learning, the induced MDP M_κ becomes equivalent to M, if $\kappa \equiv S \times A$. In our analysis, we show instead that when it is difficult to escape from the set κ, then an ϵ-optimal policy in M_κ is an $\left(\epsilon + \frac{\alpha}{1-\gamma}\right)$-optimal policy in M.

Strehl et al. [2009] introduced a meta-PAC-MDP theorem that has been successfully used to analyze a number of RL algorithms. The next theorem is a minor generalization of [Strehl et al., 2009, Theorem 10].

Theorem 4 *[Strehl et al., 2009, Theorem 10] Let $\epsilon > 0$, $\delta \in (0,1]$, and $\mathcal{A}(\epsilon, \delta)$ be any value-based greedy learning algorithm such that, for every timestep t, $\mathcal{A}(\epsilon, \delta)$ maintains action-value estimates $\hat{Q}_t \leq \frac{1}{1-\gamma}$ and there exists a set κ_t of state-action pairs that depends on the agent's history up to timestep t. We denote $\max_{a \in A} \hat{Q}_t(s,a)$ by $\hat{V}_t(s)$ and assume that the set κ does not change unless an action-value is updated (i.e., $\kappa_t = \kappa_{t+1}$ unless, $\hat{Q}_t \neq \hat{Q}_{t+1}$) or an escape event occurs. Let M_{κ_t}*

be the known state-action MDP with respect to MDP M and W and π_t be the greedy policy followed by $\mathcal{A}_t$. Suppose that with probability at least $1-\delta$ the following conditions hold for all state-action pairs $s \in S$ and timesteps $t \geq 1$:

Condition 1: $\hat{V}_t(s) \geq V^*_M(s) - \beta$ *(optimism),*

Condition 2: $\hat{V}_t(s) - V^{\mathcal{A}_t}_{M_{\kappa_t}}(s) \leq \epsilon$ *(accuracy), and*

Condition 3: *the total number of updates of action-value estimates plus the number of times the escape event from κ_t can occur is bounded by $\zeta(\epsilon, \delta)$ (learning complexity).*

If $\mathcal{A}(\epsilon, \delta)$ is executed on M it will follow a $(3\epsilon + \beta)$-optimal policy on all but

$$O\left(\frac{\zeta(\epsilon,\delta)}{\epsilon^2(1-\gamma)^2}\ln\frac{1}{\delta}\ln\frac{1}{\epsilon(1-\gamma)}\right)$$

timesteps t with probability at least $1-2\delta$.

The key difference between this version of the theorem and the one reported in Strehl et al. [2009] is that we have separated the error variable β in condition 1 from the error variable ϵ in condition 2, which allows for a tighter optimality bound in our proof of Theorem 2. We omit the complete proof of Theorem 4 because our modification from [Strehl et al., 2009, Theorem 10] requires only minor changes to the proof. However, due to our modified notion of induced MDP, we discuss why the core inequalities hold without modification. The main objective is to argue that the policy followed by the algorithm is either a near-optimal policy in the MDP M or the algorithm's policy is likely to experience a state-action pair that is outside of the set of "well-modeled" experiences. Since the latter can only happen a finite number of times before there are no more "poorly modeled" state-action pairs (because our knowledge of the state-action pair improves with each experience), then eventually the algorithm must follow a near-optimal policy in M. In the proof of Strehl et al. [2009, Theorem 10], the argument starts by comparing the algorithm's value in M over a finite sequence of timesteps H to the same policy π_t in the induced MDP M_{κ_t} over H timesteps minus the probability of Z, an escape event will occur during H timesteps, times $\frac{2}{1-\gamma}$. In our definition of induced MDP the equation

$$V^{\mathcal{A}_t}_M(s,H) \geq V^{\pi_t}_{M_{\kappa_t}}(s,H) - \Pr[Z]\left(\frac{2}{1-\gamma}\right)$$

still holds because π_t always has a smaller value in the induced MDP while in κ_t and outside of κ_t the difference must be bounded by $\left(\frac{2}{1-\gamma}\right)$ since $\frac{1}{1-\gamma}$ is the maximum possible value in both M and M_{κ_t}. Furthermore, the policy π_t only escapes from κ_t with probability $\Pr[Z]$. This enables us to reproduce the core inequalities used by Strehl et al. [2009] to prove their Theorem 10:

$$\begin{aligned}
V^{\mathcal{A}_t}_M(s,H) &\geq V^{\pi_t}_{M_{\kappa_t}}(s,H) - \Pr[Z]\left(\tfrac{2}{1-\gamma}\right) \\
&\geq V^{\pi_t}_{M_{\kappa_t}}(s) - \epsilon - \Pr[Z]\left(\tfrac{2}{1-\gamma}\right) \\
&\geq \hat{V}_t(s) - 2\epsilon - \Pr[Z]\left(\tfrac{2}{1-\gamma}\right) \\
&\geq V^*_M(s) - \beta - 2\epsilon - \Pr[Z]\left(\tfrac{2}{1-\gamma}\right)
\end{aligned}$$

where the second step is obtained by choosing H (see Kearns and Singh [2002, Lemma 2]) large enough so that $V^{\pi_t}_{M_{\kappa_t}}(s) - V^{\pi_t}_{M_{\kappa_t}}(s,H) \leq \epsilon$. The third step is due to Condition 2 ($\hat{V}_t(s) - V^{\mathcal{A}_t}_{M_{\kappa_t}}(s) \leq \epsilon$), and the final step is due to Condition 1 ($\hat{V}_t(s) \geq V^*_M(s) - \beta$).

7.2. Delayed Q-learning Background & Lemmas

Similar to Strehl et al. [2009], our analysis of Delayed Q-learning will apply Theorem 4. The analysis of Delayed Q-learning mostly consists of finding appropriate values for arguments $m > 0$ and $\epsilon_1 > 0$ that depend on $\epsilon > 0$ and $\delta \in (0, 1]$, where increasing the value of m provides more statistical accuracy by averaging over more samples and decreasing ϵ_1 causes the algorithm to learn to act closer to optimal. Our analysis is similar to Strehl et al. [2006, 2009], but there are significant differences due to initializing the action-value estimates $\hat{Q}$ with an α-weak admissible heuristic.

We have assumed that the immediate rewards are bound to the interval $[0, 1]$. In Strehl et al. [2006], the Delayed Q-learning algorithm worked by initializing its action-values to $\frac{1}{1-\gamma}$ (the maximum possible action-value) and decreasing these estimates in a series of updates. Here the main difference is that Delayed Q-learning is initialized by $\hat{Q}_0 = W$. Initializing the action-value estimates with a weak admissible heuristic introduces several technical issues that are not addressed by Strehl et al. [2006] or Strehl et al. [2009]. Delayed Q-learning is called "delayed" because it updates action-value estimates in a series of batches of m samples from a state-action pair (s, a) before attempting to update (s, a)'s action-value estimate. It may collect many batches of m samples from each state-action pair. To know when to stop updating a state-action pair, Delayed Q-learning maintains a Boolean value for each state-action pair. These Boolean values are called the $LEARN$ flags.

Definition 3 *A* **batch** *of m samples*

$$AU(s, a) = \frac{1}{m} \sum_{i=1}^{m} \left(R_{k_i}(s, a) + \gamma \max_{a' \in A_{s'}} \hat{Q}_{k_i}(s', a') \right) \tag{9}$$

occurring at timesteps $k_1 < k_2 < \cdots < k_m$ for a state-action pair (s, a) consists of a sequence of m visits to (s, a) where the first visit occurs at a timestep k_1 corresponding to either the first timestep that (s, a) is visited by the algorithm or during the most recent prior visit to (s, a) at timestep $k' < k_1$, $LEARN_{k'}(s, a) = true$ and $l_{k'}(s, a) = m$ was true. A batch of samples is said to be completed on the first timestep $t > k_1$ such that $LEARN_t(s, a) = true$ and $l_t(s, a) = m$.

Delayed Q-learning has three notions of update:

1. Attempted Updates : An attempted update occurs when a batch of m samples has just been completed.
2. Update (or Successful Updates) : A successful update to a state-action pair (s, a) occurs when a completed batch of m samples at timestep t causes a change to the action-value estimates so that $\hat{Q}_t(s, a) \neq \hat{Q}_{t+1}(s, a)$.
3. Unsuccessful Updates : An unsuccessful update occurs when a batch of m samples completes but no change occurs to the action-values.

An attempted update to a state-action pair (s, a) at timestep t is successful if

$$\hat{Q}_t(s, a) - AU_t(s, a) \geq 2\epsilon_1 \tag{10}$$

the completed batch of samples is significantly lower ($< 2\epsilon_1$) than the current action-value estimate, and a successful update assigns

$$\hat{Q}_{t+1}(s, a) = AU_t(s, a) + \epsilon_1 \tag{11}$$

the completed batch of samples plus a small constant to the new action-value estimate. These rules guarantee that whenever a successful update occurs, the action-value estimates decrease by at least ϵ_1.

Lemma 5 *[Strehl et al., 2009, Lemma 19] No more than* $u = NK\left(1 + \frac{NK}{\epsilon_1(1-\gamma)}\right)$ *attempted updates can occurs during an execution of Delayed Q-learning on* M.

Lemma 5 bounds the total number of attempted updates that can possibly occur during the execution of Delayed Q-learning. This lemma is important because it allows us to bound the total probability that a failure event will occur.

The set

$$\kappa_t = \left\{(s,a) \in S \times A \mid \hat{Q}_t(s,a) - \left(R(s,a) - \gamma \sum_{s' \in S} T(s'|s,a)\hat{V}_t(s')\right) \leq 3\epsilon_1\right\} \tag{12}$$

consists of state-action pairs with small Bellman residual. Because the state-action pairs in κ_t have low Bellman error, Szita and Szepesvári [2010] has called Eq. (12) the "nice" set. The set κ_t is somewhat analogous to the known-state MDP used in the analysis of R-MAX. However, unlike R-MAX, the algorithm cannot determine which state-action pairs are actually in this set. It is used strictly for the purposes of analysis.

Let $\mathcal{X}_1$ denote the event that when Delayed Q-learning is executed on M, then every time k_1 when a new batch of samples for some state-action pair (s,a) begins, if $(s,a) \notin \kappa_{k_1}$ and the batch is completed at timestep k_m, then a successful update to (s,a) will occur at timestep k_m.

Lemma 6 *[Strehl et al., 2006, Lemma 1] If Delayed Q-learning is executed on* M *with parameters* m *and* ϵ_1 *where* m *satisfies*

$$m = \frac{1}{2\epsilon_1^2(1-\gamma)^2} \ln \frac{u}{\delta} \tag{13}$$

then event $\mathcal{X}_1$ *will occur, with probability at least* $1 - \delta$.

Lemma 6 establishes a value of m that is large enough to ensure that event $\mathcal{X}$ occurs with high probability.

The next lemma is a modified version of Strehl et al. [2006, Lemma 2]. The reason that this lemma needs to be modified is because not all of the action-value estimates maintained by Delayed Q-learning will be optimistic, since they are initialized according to the α-weak admissible heuristic W. Instead we show that the action-values that were initialized optimistically will remain approximately optimistic. In addition to the original purpose for this lemma, it also shows how Delayed Q-learning can be made to satisfy Condition 3 of Lemma 1, which we will use in our proof of Theorem 2.

Let $\mathcal{X}_2$ be the event that for all timesteps $t \geq 0$ and $(s,a) \in S \times A$, if $W(s,a) = \hat{Q}_0(s,a) \geq Q^*_M(s,a)$, then $\hat{Q}_t(s,a) \geq Q^*_M(s,a) - \frac{\alpha}{1-\gamma}$.

Lemma 7 *If Delayed Q-learning is initialized by the* α*-weak admissible heuristic* W *and is executed with* m *determined by (13), then event* $\mathcal{X}_2$ *occurs with probability at least* $1 - \delta$.

Proof We prove the claim by induction on the timestep t.

Base Case ($t = 0$): By our assumption if $W(s,a) \geq Q^*_M(s,a)$, then $\hat{Q}_0(s,a) = W(s,a) \geq Q^*_M(s,a) > Q^*_M(s,a) - \frac{\alpha}{1-\gamma}$.

Induction Step ($t > 0$): Suppose that for all timesteps $\tau = 1, 2, \ldots, t-1$, if $W(s,a) \geq Q^*_M(s,a)$, then $\hat{Q}_\tau(s,a) \geq Q^*_M(s,a) - \frac{\alpha}{1-\gamma}$. If during timestep $t-1$ no successful update occurs, then the action-value estimates are unchanged implying that $\hat{Q}_t(s,a) = \hat{Q}_{t-1}(s,a) \geq Q^*_M(s,a) - \frac{\alpha}{1-\gamma}$. On the other hand, if a successful update occurs during timestep $t-1$ at some state-action pair (s,a), then by the update rule given by (11)

$$\begin{aligned} \hat{Q}_t(s,a) &= AU(s,a) + \epsilon_1 \\ &= \tfrac{1}{m}\textstyle\sum_{i=1}^{m}\left(R_{k_i}(s,a) + \gamma \hat{V}_{k_i}(s'_{k_i})\right) + \epsilon_1 \end{aligned}$$

where $\hat{V}_{k_i}(s) = \max_{a \in A} \hat{Q}_{k_i}(s,a)$ for $k_1 < k_2 < \cdots < k_m = t-1$. Notice that

$$\hat{V}_{k_i}(s) \geq V^*_M(s) - \alpha - \frac{\alpha}{1-\gamma} \tag{14}$$

for all $s \in S$ and $i = 1, 2, \ldots, m$, because the definition of an α-weak admissible heuristic guarantees that there exists $\tilde{a} \in A$ such that $W(s,\tilde{a}) \geq Q^*_M(s,\tilde{a}) \geq V^*_M(s) - \alpha$. This implies that $\hat{V}_{k_i}(s) = \max_{a \in A} \hat{Q}_{k_i}(s,a) \geq Q^*_M(s,\tilde{a}) - \frac{\alpha}{1-\gamma} \geq V^*_M(s) - \alpha - \frac{\alpha}{1-\gamma}$ for every state $s \in S$. Now we can plug (14) into the previous inequality to obtain

$$\begin{aligned} \hat{Q}_t(s,a) &= \tfrac{1}{m}\textstyle\sum_{i=1}^m \left(R_{k_i}(s,a) + \gamma \hat{V}_{k_i}(s'_{k_i})\right) + \epsilon_1 \\ &\geq \tfrac{1}{m}\textstyle\sum_{i=1}^m \left(R_{k_i}(s,a) + \gamma \left(V^*_M(s'_{k_i}) - \alpha - \tfrac{\alpha}{1-\gamma}\right)\right) + \epsilon_1 \\ &= \tfrac{1}{m}\textstyle\sum_{i=1}^m \left(R_{k_i}(s,a) + \gamma V^*_M(s'_{k_i})\right) - \tfrac{\alpha}{1-\gamma} + \epsilon_1 \end{aligned} \quad .$$

Now we define the random variables $J_i = \left(R_{k_i}(s,a) + \gamma V^*_M(s'_{k_i})\right)$ for $i = 1, 2, \ldots, m$ bound by the interval $\left[0, \frac{1}{1-\gamma}\right]$. By the Hoeffding inequality and our choice of m (13), $\frac{1}{m}\sum_{i=1}^m J_i \geq E[J_1] - \epsilon_1$ with probability at least $1 - \frac{\delta}{u}$. Thus

$$\begin{aligned} \hat{Q}_t(s,a) &\geq E[J_1] - \epsilon_1 + \epsilon_1 - \tfrac{\alpha}{1-\gamma} \\ &= E[J_1] - \tfrac{\alpha}{1-\gamma} \\ &= Q^*_M(s,a) - \tfrac{\alpha}{1-\gamma} \end{aligned}$$

We extend our argument over all u potential attempted updates using the union bound so that $\hat{Q}_t(s,a) \geq Q^*_M(s,a) - \frac{\alpha}{1-\gamma}$ holds over all timesteps $t \geq 0$, with probability at least $1 - u\frac{\delta}{u} = 1 - \delta$. ■

The next lemma bounds the number of timesteps that a state-action pair outside of the "nice" set κ can be experienced. This lemma is a modification of [Strehl et al., 2006, Lemma 4]. Our lemma decreases the number of times that a state-action pair that is not in κ can be experienced from $\frac{2mNK}{\epsilon_1(1-\gamma)}$ to $\frac{2m(NK-X)}{\epsilon_1(1-\gamma)}$, where X is the number of state-action pairs that are eliminated by initializing the action-value estimates with W. If X is large this represents a significant decrease in the number of experiences of state-action pairs outside of κ, which is the key factor in proving Theorem 2.

Lemma 8 *If events $\mathcal{X}_1$ and $\mathcal{X}_2$ occur, then $(s,a) \notin \kappa_t$ can be experienced on at most $\frac{2m(NK-X)}{\epsilon_1(1-\gamma)}$ timesteps t, where $X = \left|\left\{(s,a) \in S \times A \mid W(s,a) < V^*_M(s) - \alpha - \frac{\alpha}{1-\gamma}\right\}\right|$.*

Proof We start by applying Lemma 1. By its definition the Delayed Q-learning algorithm always follows a greedy policy (satisfying Condition 1 of Lemma 1) and cannot change an action-value estimate unless the corresponding state-action pair has been experienced (satisfying Condition 2 of Lemma 1). Since event $\mathcal{X}_2$ occurs, this satisfies Condition 3 of Lemma 1. Therefore by Lemma 1, under the above assumptions, Delayed Q-learning will only visit $NK - X$ state-action pairs.

The result follows from the proof of [Strehl et al., 2006, Lemma 4] over $NK - X$ state-action pairs rather than all NK state-action pairs. ■

7.3. Proof of Theorem 2

Now we are ready to prove Theorem 2.

Proof (of Theorem 2) We proceed by applying Theorem 4. We select $\epsilon_1 = \frac{\epsilon(1-\gamma)}{3}$ and m according to (13) so that event $\mathcal{X}_1$ holds with probability at least $1-\delta$ (by Lemma 6) and event $\mathcal{X}_2$ holds with probability at least $1-\delta$ (by Lemma 7).

Suppose that events $\mathcal{X}_1$ and $\mathcal{X}_2$ occur. Condition 1 ($\hat{V}_t(s) \geq V^*_M(s) - \beta$) of Theorem 4 is satisfied with $\beta = \frac{\alpha}{1-\gamma}$ (by Lemma 7). Now we argue that Condition 2 ($\hat{V}_t(s) - V^{\pi_t}_{M_{\kappa_t}}(s) \leq \epsilon$) of Theorem 4 is also satisfied. Notice that

$$\hat{V}_t(s) - V^{\pi_t}_{M_{\kappa_t}}(s) \leq \begin{cases} \gamma \sum_{s' \in S} T(s'|s,a) \left(\hat{V}_t(s') - V^{\pi_t}_{M_{\kappa_t}}(s') \right) + 3\epsilon_1 & \text{if } (s, \pi_t(s)) \in \kappa_t \\ 0 & \text{if } (s, \pi_t(s)) \notin \kappa_t \end{cases}$$

because $\hat{V}_t(s) - \left(R(s,a) + \gamma \sum_{s' \in S} \hat{V}_t(s') \right) \leq 3\epsilon_1$, whenever $(s, \pi_t(s)) \in \kappa_t$. This implies that $\hat{V}_t(s) - V^{\pi_t}_{M_{\kappa_t}}(s) \leq \frac{3\epsilon_1}{1-\gamma} = \frac{3(\epsilon(1-\gamma)/3)}{1-\gamma} = \epsilon$.

The number of timesteps that a state-action pair that is not in κ_t can be experienced is bounded in Lemma 8, which satisfies Condition 3 of Theorem 4. By our choice of m, events $\mathcal{X}_1$ and $\mathcal{X}_2$ occur with probability at least $1-2\delta$ (applying the union bound). Therefore the conditions of Theorem 4 are met with probability at least $1-2\delta$. By applying Theorem 4 and plugging in the values for the learning complexity, m, and ϵ_1, we obtain the fact that on all but

$$O\left(\frac{NK - X}{\epsilon^4 (1-\gamma)^8} \ln \frac{1}{\delta} \ln \frac{1}{\epsilon(1-\gamma)} \ln \frac{NK}{\delta \epsilon (1-\gamma)} \right)$$

timesteps, Delayed Q-learning follows an $\left(3\epsilon + \frac{\alpha}{1-\gamma}\right)$-optimal policy, with probability at least $1-3\delta$. We obtain our final result by setting $\epsilon \leftarrow \frac{\epsilon}{3}$ and $\delta \leftarrow \frac{\delta}{3}$, which modifies the bound by constant factors only. ■

7.4. Proof of Theorem 3

Proof (of Theorem 3) Let $s \in S_{\text{trg}}$ be a state in the target task. Because h is assumed to satisfy (5) and (6) there exists an action $\tilde{a} \in A_{\text{trg}}$ such that either (1) $(s, \tilde{a}) \in D$ and $V^*_{\text{trg}}(s) - \alpha \leq Q^*_{\text{trg}}(s, \tilde{a}) \leq Q^*_{\text{src}}(h(s, \tilde{a}))$ or (2) $(s, \tilde{a}) \notin D$ and $V^*_{\text{trg}}(s) - \alpha \leq Q^*_{\text{trg}}(s, \tilde{a})$.

In the first case ($(s, \tilde{a}) \in D$), Eq. (7) will assign the value

$$\begin{aligned} W(s, \tilde{a}) &= \min \left(\hat{Q}_{\text{src}}(h(s, \tilde{a})) + \epsilon_{\text{src}}, \tfrac{1}{1-\gamma} \right) \\ &\geq \hat{Q}_{\text{src}}(h(s, \tilde{a})) + \epsilon_{\text{src}} \\ &\geq (Q^*_{\text{src}}(h(s, \tilde{a})) - \epsilon_{\text{src}}) + \epsilon_{\text{src}} \\ &\geq Q^*_{\text{trg}}(s, \tilde{a}) \geq V^*_{\text{trg}}(s) - \alpha \end{aligned}$$

where the initial equality is due to the fact that $(s, \tilde{a}) \in D$ and values are transferred according to Eq. (7). The first step removes the minimum operation. The second step replaces the source task action-value estimates with the true source task action-values by subtracting ϵ_{src}. The final step is due to the fact that the intertask mapping satisfies (5). This inequality satisfies the α-weak admissible heuristic criteria for state s.

In the second case ($(s, \tilde{a}) \notin D$), $W(s, \tilde{a})$ is assigned the maximum value $\frac{1}{1-\gamma}$ which is greater than or equal to $Q^*_{\text{trg}}(s, \tilde{a})$ and $V^*_{\text{trg}}(s) - \alpha$. Therefore in both cases, the α-weak admissible heuristic criteria is satisfied at the state s. Since this argument holds for all states, then the transferred action-values induce an α-weak admissible heuristic with respect to the target task M_{trg}.

The remainder of proof follows from applying Theorem 2 to the target task with the transferred action-values taking the place of the α-weak admissible heuristic. ■