Sample Complexity of Multi-task Reinforcement Learning

A dream of artificial intelligence is to have lifelong learning agents that learn from prior experience to improve their performance on future tasks. Our interest is in multi-task reinforcement learning (RL), where an agent acts in a sequence of RL tasks. We assume that each task is drawn from a finite set of Markov decision processes with identical state and action spaces, but different reward and/or transition model parameters; however, the MDP parameters are initially unknown and the MDP identity of each new task is also unknown. This model is sufficiently rich to capture important applications like tutoring systems that teach a series of students whose initially unknown learning dynamics can be captured by a small set of types (such as honors, standard and remedial), marketing systems that may characterize a customer into a finite set of types and use that to adaptively provide targeted advertising over time, and medical decision support systems that seek to provide good care to patients suffering from the same condition for whom the best treatment strategy is determined by a hidden latent variable about the patient's physiology.

Although there is encouraging empirical evidence that transferring information can improve RL performance (see Taylor and Stone [2009] for a recent survey), aside from a couple exceptions [Lazaric and Restelli, 2011, Mann and Choe, 2012], there has been very little theoretical work on transfer or mult-task reinforcement learning. In particular, we are aware of no work that seeks to formally analyze and prove bounds on how transferred knowledge can accelerate RL in online multi-task learning settings. Towards addressing this gap, we introduce a new multi-task RL algorithm whose per-task sample complexity can be significantly lower due to leveraging prior experience compared to the single-task RL sample complexity.

We consider an agent learning across across a series of T RL tasks, each run for H steps. We assume each task is sampled from a set $\mathcal{M}$ of C MDPs, which share the same state, action, and discount factor, but have different reward and/or transition dynamics.

Our algorithm involves two phases. Since at the start the agent does not know the model parameters, during the first phase the agent acts in each task, and uses the observed transitions and rewards to estimate the parameters of the set of underlying MDPs. Then in phase 2 the agent uses these learned models to accelerate learning in each new task.

More precisely, in phase 1, T_1 tasks are drawn iid from an unknown distribution. On each task the agent executes the single-task RL algorithm E^3 [Kearns and Singh, 2002], and stores the observed transitions and rewards per task.

After phase 1 finishes, this data is clustered to identify a set of at most $\hat{C}$ MDPs. To do this, the mean transition and reward MDP parameters are estimated for each task, and tasks whose parameters differ by no more than a fixed threshold are clustered together. The transition and reward counts for tasks in the same cluster are then

merged.

At the start of phase 2 the agent now has access to a set of (at most) $\bar{C}$ MDPs which approximate the true set of MDP models from which each new task is sampled. The key insight is that the agent can use these models to identify the model of the current task, and then act according to the policy of identified model, and that this process of identification is generally faster than the standard exploration needed in single-task learning. To do so, we draw upon and extend the noisy union learning algorithm Li et al. [2011]. One critical distinction is that our approach can be used to compare models that themselves do not have perfect estimates of their own parameters, and do so in way that allows us to eliminate models that are sufficiently unlikely to have generated the observed data. A key part of our approach is to eliminate a particular model when there is sufficient evidence that the model is very unlikely to have generated the observed transitions and rewards of the current task. We do this by tracking the difference in the sum of the ℓ_2 error between the current task's observed (s, a, s', r) transitions and the transitions predicted given each of the $\hat{C}$ MDP models obtained at the end of phase 1.

The base algorithm used for each task in phase 2 is very similar to E^3, allowing us to at least preserve the standard single-task PAC RL performance. Consequently, negative transfer, relative to single-task E^3, is avoided[1].

Under a few assumptions, we can prove that the sampling complexity of this algorithm is significantly smaller in phase 2 than standard single-task RL approaches. The most important of these assumptions is that there is a known diameter D, such that for every MDP in $\mathcal{M}$, any state s' is reachable from any state s in at most D steps *on average*. This assumption ensures we can obtain sufficiently good estimates of each task's parameters that it is possible to cluster and learn the parameters of the underlying MDPs.

Theorem 1 *Given any ϵ and δ, run the algorithm for T tasks, each for $H = O\left(\frac{DSA}{\Gamma^2}\log\frac{T}{\delta}\right)$ steps. Then, the algorithm will select an ϵ-optimal policy on all but at most $\tilde{O}\left(\frac{\zeta V_{\max}}{\epsilon(1-\gamma)}\right)$ steps, with probability at least $1-\delta$, where*

$$\zeta = O\left(T_1\zeta_s + \bar{C}\zeta_s + (T - T_1)\frac{NV_{\max}^2\bar{C}}{\epsilon^2(1-\gamma)^2}\right),$$

and $\zeta_s = \tilde{O}\left(\frac{NSAV_{\max}^2}{\epsilon^2(1-\gamma)^2}\right)$.

In particular, on average and asymptotically, in phase 2 our algorithm has a sample complexity that is independent of the size of the state and action spaces, trading this for a dependence on the number of models $\bar{C}$. In contrast, applying single-task learning without transfer in T tasks can lead to an overall sample complexity of $\tilde{O}(T\zeta_s) = O(TNSA)$. Since we expect $\bar{C} \ll SA$, this yields a significant improvement over single-task reinforcement learners, whose sample complexity has at least a linear dependence on the size of the state–action space [Strehl et al., 2006, Szita and Szepesvári, 2010], and some have a polynomial dependence.

Looking forward, one very interesting direction is to explore similar results in very large or continuous state spaces, possibly relying on function approximation or compact model representations. We anticipate this will be important to address real-world multi-task RL applications.

[1] Up to log factors.

References

M. J. Kearns and S. P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2–3):209–232, 2002.

A. Lazaric and M. Restelli. Transfer from Multiple MDPs. In *NIPS*, 2011.

L. Li, M. L. Littman, T. J. Walsh, and A. L. Strehl. Knows what it knows: A framework for self-aware learning. *Machine Learning*, 82(3):399–443, 2011.

T. A. Mann and Y. Choe. Directed exploration in reinforcement learning with transferred knowledge. In *EWRL*, 2012.

A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman. PAC model-free reinforcement learning. In *ICML*, 2006.

I. Szita and C. Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. In *ICML*, 2010.

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