Stable Reinforcement Learning with Unbounded State Space (Extended Abstract)

Devavrat Shah
EECS, MIT
devavrat@mit.edu

Qiaomin Xie
ORIE, Cornell University
qiaomin.xie@cornell.edu

Zhi Xu
EECS, MIT
zhixu@mit.edu

We consider the problem of reinforcement learning (RL) for controlling an unknown dynamical system with an unbounded state space. Such problems are ubiquitous in various application domains, as exemplified by scheduling for networked systems. As a paradigm for learning to control dynamical systems, RL has a rich literature. In particular, algorithms for settings with finite, bounded or compact state spaces have been well studied, with both classical asymptotic results and recent non-asymptotic performance guarantees. However, literature on problems with unbounded state space is scarce, with few exceptions such as linear quadratic regulator, where the structure of the dynamics is *known*. Indeed, the unboundedness of the state space presents with new challenges for algorithm or policy design, as well as analysis of policy in terms of quantifying the "goodness".

Solutions that may not work. In traditional RL approaches, the policy is trained *offline* using finitely many samples for a finite, bounded or compact state space and then it is deployed in the *wild* without further changes. A natural adaption of such an approach is by restricting the RL policy to a finite subset of the state space chosen appropriately or arbitrarily. Examples of such algorithms include model-based methods such as UCRL/PSRL and model-free methods such as TD/Q-Learning/policy gradient. However, even in simple queueing networks, the system will reach a state (i.e., queue lengths) q not contained in the training data with non-zero probability. The estimate for q's transition probabilities and value function will remain at their initial/default values (say 0). With such an uninformative estimate, the corresponding policy will be independent of the state q. And it is likely that the policy may end up serving empty queue with a nonzero probability. This might cause the queues to grow unboundedly with strictly positive probability. Clearly, more sophisticated approaches to truncate systems are not going to help as they will suffer from a similar issue.

An alternative to truncation is to "compactify" the state space by mapping the unbounded space to a bounded set. However, traditional RL approaches may also fall short for the reduced problem: properties of the original problem that allow for efficient learning can be easily destroyed under the mapping. Consider a simple example where the state space is $\mathcal{S} = (-\infty, \infty)$. Suppose that for the original MDP, the optimal value function V^* that we wish to learn is L-Lipschitz. Consider a natural mapping $z = \tanh s$, which "compactifies" the unbounded space $\mathcal{S}$ to $[-1, 1]$. Using chain rule, we have $|\frac{\partial V^*}{\partial z}| = |\frac{\partial V^*}{\partial s} \cdot \frac{\partial s}{\partial z}| \sim L \cdot |\frac{1}{1-z^2}|$. As the original state s approaches infinity, z approaches either 1 or -1, in which case the derivative becomes infinity, implying that the smoothness property is completely lost. Therefore, it may not be possible to learn the function over the bounded set well with *finite* samples. Such issues will be exaggerated in higher dimensions. In general, this kind of state space compactification suffers similar issues as truncation: it necessarily discounts/skews large states, which are exactly the states we care about when studying systems such as queueing networks. Indeed, it is challenging to find a meaningful compactification map that preserves all the nice properties; thus efficient learning in the "compactified" space is far from obvious if not impossible.

Another potential approach is to find "lower dimensional structure" through functional approximation, e.g., by parametrizing the policy π within some function class (such as linear functions or neural networks). For this approach to work, the function class must be expressive enough to contain a stable policy. However, it is not at all clear, *a priori*, which parametric function class has this property. This challenge is only exacerbated in more complicated systems. Although some approximation architectures work well empirically, there is no rigorous performance guarantee in general.

Extended Abstract. Full Version: https://arxiv.org/abs/2006.04353.

Challenges. To sum up, the traditional RL approaches for finite, bounded or compact state spaces are not well suited for systems with unbounded state spaces. Approaches that rely on *offline* training only are bound to fail as system will reach a state that is not observed in finitely many samples during offline training and hence, there is no meaningful guidance from the policy. Therefore, to learn a reasonable policy with an unbounded state space, the policy ought to be updated whenever a new scenario is encountered. That is, unlike traditional RL, we need to consider *online* policies, i.e., one that is continually updated upon incurring new scenarios.

Another challenge is in analyzing or quantifying "goodness" of such a policy. Traditionally, the "goodness" of an RL policy is measured in terms of the error induced in approximating, for example, the optimal value function over the entire state space; usually measured through $\|\cdot\|_\infty$ norm error bound. Since the state space is unbounded, expecting a good approximation of the optimal value function over the entire state space is not a meaningful measure. Therefore, we need an alternative to quantify the "goodness" of a policy.

Questions of interest. In this work, we are interested in the following questions: (a) What is the appropriate "goodness" of performance for a RL policy for unbounded state space? (b) Is there an online, data-driven RL policy that achieves such "goodness"? and if so, (c) How does the number of samples required per time-step scale?

Our contributions. Motivated by the above considerations, we consider discounted Markov decision processes with an unbounded state space and a finite action space, under a generative model which allows one to sample state transitions given any state-action pair.

Notion of Stability. As the main contribution, we introduce a notion of *stability* to quantify "goodness" of a RL policy for unbounded state space inspired by the literature in queueing systems and control theory. Informally, a RL policy is stable if the system dynamics under the policy returns to a finite, bounded or compact subset of the system infinitely often — in the context of queueing networks, it would imply that queues remain finite with probability 1. For applications where instability implies unbounded cost, the notion of stability provides a meaningful notion of first-order optimality. Indeed, further refined notions of performance beyond stability, such as diffusion-approximation or heavy traffic analysis as typically considered in queueing systems would be natural next steps to consider.

Stable RL Policy. As a proof of concept, we present a simple RL policy using a Sparse Sampling Monte Carlo Oracle that is stable for any MDP, as long as the optimal policy respects a Lyapunov function with *drift* condition. Our policy *does not* require knowledge of or access to such a Lyapunov function. It recommends an action at each time using finitely many *simulations* of the MDP through the oracle. That is, the policy is *online* and guarantees stability for each trajectory starting *without* any prior training. The number of samples required at each time step scales as $O\Big(\big(\frac{1}{\alpha^4}\log^2\frac{1}{\alpha}\big)^{O(\log\frac{1}{\alpha})}\Big)$, where $-\alpha<0$ is the drift in Lyapunov function. To the best of our knowledge, this is the first online RL policy that is *stable* for generic MDP with unbounded state space.

Sample Efficient Stable RL Policy. To further improve the sample efficiency, for MDPs with Lipschitz optimal value function, we propose a modified Sparse Sampling Monte Carlo Oracle for which the number of samples required at each time step scales as $O\Big(\frac{1}{\alpha^{2d+4}}\log^{d+1}\frac{1}{\alpha}\Big)$, where d is the dimension of the state space. That is, the sample complexity becomes polynomial in $1/\alpha$ from being super-polynomial with the vanilla oracle. The efficient oracle utilizes the minimal structure of smoothness in the optimal value function and should be of interest in its own right, as it provides sample complexity improvement for *all* policies in literature where such an oracle plays a key role.

Adaptive Algorithm Based on a Statistical Test. While the algorithm does not require knowing the Lyapunov function itself, it does have a parameter whose optimal value depends on the drift parameter of the Lyapunov function. Therefore, we further develop an adaptive, agnostic version of our algorithm that automatically searches for an appropriate tuning parameter. We establish that either this algorithm discovers the right value and hence ensures stability, or the system is near-stable in the sense that $||s_t||/\log^2 t = O(1)$ as $t\to\infty$. The near-stability is a form of sub-linear regret. For example, in the context of a queueing system, this would correspond to queues growing as $O(\log^2 t)$ with time in contrast to $O(1)$ queues for stable (or optimal) policy. Further, the sub-linear growth of queue lengths implies "rate" stability — to the best of our knowledge, this is first such general RL policy for generic queueing systems with such a property.