Adaptive Discretization for Episodic Reinforcement Learning in Metric Spaces

Sean R. Sinclair
Cornell University
srs429@cornell.edu

Siddhartha Banerjee
Cornell University
sbanerjee@cornell.edu

Christina Lee Yu
Cornell University
cleeyu@cornell.edu

Introduction

Reinforcement learning (RL) is a natural model for systems involving real-time sequential decision making [5]. In these models, an agent interacts with a system having stochastic transitions and rewards, and aims to learn to control the system by exploring available actions using real-time feedback. This requires the agent to navigate the *exploration-exploitation trade-off*, between exploring unseen parts of the environment and exploiting historical high-reward actions. A popular approach is Q-Learning algorithms which focus on learning good state-action mappings [1, 4, 6]. In continuous or large finite state-action spaces, RL algorithms require embedding the state-action space in a metric space and using an appropriate discretization of the space. Under this embedding, many problems exhibit structure where nearby actions result in a similar behavior of the system. This motivates our central question: *Can we modify Q-learning algorithms to learn a near-optimal policy while limiting the size of the discretization?*[1]

Current approaches to this problem consider uniform discretization policies, which are either fixed based on problem primitives, or updated via a fixed schedule (for example, via a 'doubling trick') [4]. However, a more natural approach is to adapt the discretization over space and time in a data-driven manner. This allows the algorithm to learn policies which are not uniformly smooth, but adapt to the geometry of the underlying space. Moreover, the agent would then be able to explore more efficiently by only sampling important regions.

Model and Algorithm

We consider an agent interacting with an underlying finite-horizon Markov Decision Process (MDP) over K sequential episodes. The underlying MDP is given by a five-tuple $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ where $\mathcal{S}$ denotes the set of states, $\mathcal{A}$ the set of actions, and horizon H is the number of steps in each episode. We further assume that $\mathcal{S} \times \mathcal{A}$ is a compact metric space with a given metric $\mathcal{D} : (\mathcal{S} \times \mathcal{A})^2 \to \mathbb{R}$. A policy π gives a sequence of mappings $\pi_h : \mathcal{S} \to \mathcal{A}$ dictating the action taken in a given state. From this we can define the Value function and Q function which gives the expected future rewards starting from a given state:

$$V_h^\pi(x) := \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'}(x_{h'}, \pi_{h'}(x_{h'})) \mid x_h = x\right]$$

$$Q_h^\pi(x,a) := r_h(x,a) + \mathbb{E}\left[\sum_{h'=h+1}^{H} r_{h'}(x_{h'}, \pi_{h'}(x_{h'})) \mid x_h = x, a_h = a\right].$$

Clearly, the optimal policy $\pi^\star$ is the one satisfying $V_h^{\pi^\star}(x) = \sup_\pi V_h^\pi(x)$. The agent plays the game over K episodes $k = 1, \dots, K$ where for each episode k the agent selects a policy π^k which is executed over the H steps. Their goal is to minimize the *regret*, the expected loss the agent experiences by exercising their policy instead of an optimal policy in every episode, defined as $R(K) = \sum_{k=1}^{K} V_1^{\pi^\star}(x) - V_1^{\pi^k}(x)$.

[1] The full paper will be published in PACM Measurement and Analysis of Computing Systems (POMACS) but a draft is available at `https://arxiv.org/abs/1910.08151`.

Optimization Foundations for Reinforcement Learning Workshop at NeurIPS 2019, Vancouver, Canada.

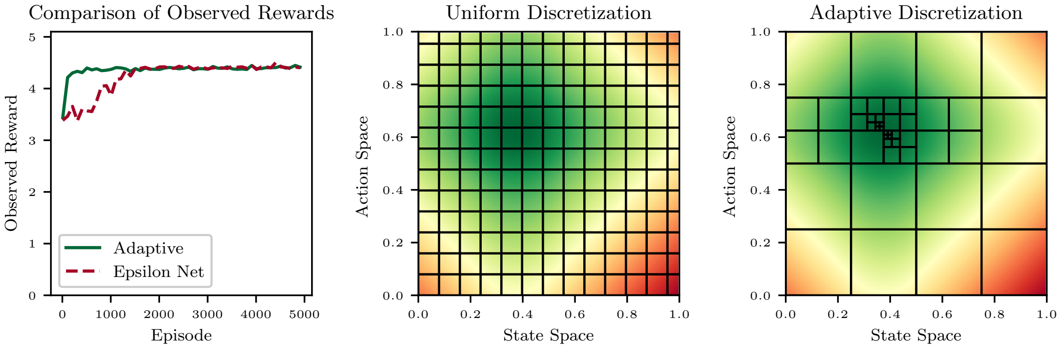

Figure 1: Comparison of the observed rewards and state-action space discretization under the uniform mesh ($\epsilon-$Net) algorithm [4] and our adaptive discretization algorithm. The colors correspond to the relative $Q_h^\star$ value of the given state-action pair, where green corresponds to a higher value for the expected future rewards. The adaptive algorithm converges faster to the optimal policy by keeping a partition whose coarseness corresponds to the quality of that location.

Our algorithm manages the trade-off between exploration and exploitation by careful use of event counters and upper-confidence bounds (UCB), similar to UCB algorithms popular in multi-armed bandits. For each step $h = 1, \dots, H$ it maintains a collection of balls $\mathcal{P}_h^k$ of $\mathcal{S} \times \mathcal{A}$ which is refined over the course of learning. Each element $B \in \mathcal{P}_h^k$ is a ball with radius $r(B)$. For any ball $B \in \mathcal{P}_h^k$ we maintain an *upper confidence value* $\mathbf{Q}_h^k(B)$ for the true $Q_h^\star$ of points in B and $n_h^k(B)$ for the number of times B or its ancestors have been selected by the algorithm at step h in episodes up to k.

The algorithm proceeds by:

- **selection rule**: Select a relevant ball for the current state with maximal value of $\mathbf{Q}_h^k(B)$. Select any action a to play inside of B. This is similar to "greedy upper confidence algorithms" for multi-armed bandits [2, 3].
- **update parameters**: Increment $n_h^k(B)$ by 1, and update the $\mathbf{Q}_h^k(B)$ value for the selected ball given the observed reward r and the state the agent transitions to x_{new}. This is done according to
$$\mathbf{Q}_h^k(B) = (1-\alpha)\mathbf{Q}_h^k(B) + \alpha(r + \mathbf{V}_h^k(x_{new}) + \texttt{BONUS})$$
where α is the learning rate, r is the observed reward, `BONUS` is the UCB term, and $\mathbf{V}_h^k(x_{new})$ is the estimated expected future value from the new state we transitioned to.
- **re-partition the space:** Let B denote the selected ball and $r(B)$ its radius. We split the ball B when $n_h^k(B) \geq (1/r(B))^2$. This forms a bias-variance trade-off between the estimates $\mathbf{Q}_h^k(B)$ of the quality of actions in B relative to its radius.

Main Results

We show that our adaptive discretization policy achieves near-optimal dependence of the regret on the covering dimension of the metric space. In particular, we prove that over K episodes, our algorithm achieves a regret bound

$$R(K) = \tilde{O}\left(H^{5/2}K^{(d+1)/(d+2)}\right)$$

where d is the covering dimension and H is the number of steps in each episode. Moreover, for non-uniform metric spaces where the covering dimension is not tight, we show improved bounds which adapt to the geometry of the space. Existing lower bounds show that this is optimal up to logarithmic terms for K and is off by a linear factor for H.

We compare our algorithm to the net based Q-learning algorithm from [4] on two canonical problems. Our algorithm achieves order-wise better empirical rewards compared to the uniform mesh algorithm, while maintaining a much smaller partition. As an example, in Figure 1 we demonstrate the performance of our algorithm and net based Q-learning for an ambulance routing problem. We see that the adaptive discretization maintains different levels of coarseness across the space, resulting in a faster convergence rate to the optimal policy as compared to the uniform mesh algorithm.

References

[1] Jin C, Jordan M.I, Allen-Zhu Z, Bubeck S, and NeurIPS 2018 32nd Conference on Neural Information Processing Systems. Is Q-learning provably efficient? *Adv. neural inf. proces. syst. Advances in Neural Information Processing Systems*, 2018-December:4863–4873, 2018. OCLC: 8096900528.

[2] Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, 2018.

[3] Aleksandrs Slivkins. Contextual Bandits with Similarity Information. *Journal of machine learning research : JMLR.*, 15(2):2533–2568, 2015. OCLC: 5973068319.

[4] Zhao Song and Wen Sun. Efficient Model-free Reinforcement Learning in Metric Spaces. *arXiv:1905.00475 [cs, stat]*, May 2019. arXiv: 1905.00475.

[5] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[6] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.