Reward-Free Reinforcement Learning with GNN and Adversarial Linear Mixture MDPs

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Reward-free RL is independently developed in the unconstrained literature, which learns the transition dynamics without using the reward information and is thus naturally capable of addressing RL with multiple objectives under the common dynamics. This paper proposes a new framework for the reward-free RL setting with function approximation i.e. the adversarial linear mixture MDPs. As Jin, et al. (2020). We partition this setting into an exploration phase and a planning phase. During the exploration phase, the agent first collects trajectories from an MDP M without a pre-specified reward function. Using the Graph Neural Networks (GNNs) to store the significant states in dataset D instead of all states, each with a heuristic weight. In the planning phase, it is tasked with computing near-optimal policies under M for a collection of given reward functions. The agent generalizes previously learned information using the linear mixture MDPs that allows it to approximate the policy given an arbitrary reward function.

1 Introduction

In reinforcement learning (RL), an agent repeatedly interacts with an unknown environment with the goal of maximizing its cumulative reward. To do so, the agent must engage in exploration, learning to visit states to investigate whether they hold high rewards.

Exploration is widely regarded as the most significant challenge in RL, because the agent may have to take precise sequences of actions to reach states with high reward. Here, simple randomized exploration strategies provably fail: for example, a random walk can take exponential time to reach the corner of the environment where the agent can accumulate high rewards (Li, 2012). While reinforcement learning has seen a tremendous surge of recent research activity, essentially all of the standard algorithms deployed in practice employ simple randomization or its variants, and consequently incur extremely high sample complexity.

In this extended abstract paper, we aim to develop an end-to-end instantiation of this proposal. To this end we ask: How can we generalize the concepts of significant states and coverage guarantees? And how can we develop such an agent that can generalize enough?

1.1 Notations

In the reward-free setting, we would like to design algorithms that efficiently explore the state space without the guidance of reward information. Over the course of K episodes, the agent collects a dataset of visited states, actions, and transitions $D = s_h^{(}k), a_h^{(}k)_{(}k, h) \in [k] \times [H]$, which is the outcome of the exploration phase.

Graph Neural Networks Graph neural networks (GNN) are a class of neural networks that operate directly on graph-structured data. A wide variety of graph neural network architectures have been

Submitted to 17th European Workshop on Reinforcement Learning (EWRL 2024). Do not distribute.

proposed. These range from simple graphs, to directed graphs, to graphs that contain information, up to convolutional graphs. The graph $G = (N, E)$ is defined as having nodes $n_i \in N$ and directed edges $e_{ij} \in E$ from node n_j to n_i. Both – the nodes and the edges – contain additional information. The node value is denoted as h_i for the i-th node and the edge value as e_{ij} connecting the i-th with the j-th node. In each layer of the GNN, a dense node neural network layer is applied per node and a dense edge neural network layer per edge. Each GNN layer has three computation steps: First, the next edge values $e_{i_j}^{k+1}$ are computed using the current edge values $e_{i_j}^{k}$, the from-node values h_i^k and the to-node values h_j^k. These values are concatenated and passed into a dense neural network layer $f_x^k(.)$ that is parameterized by X. This can be represented as:

$$e_{i_j}^{k+1} = f_x^k([h_i^k, e_{i_j}^k, h_j^k]) \tag{1}$$

Linear Mixture MDPs. We focus on a special class of MDPs named linear mixture MDPs (Ayoub et al., 2020; Cai et al., 2020; Zhou et al., 2021; He et al., 2022; Li et al., 2023), where the transition kernel is linear in a known feature mapping $\phi : SAS \to R^d$ with the following definition.

Definition 1 (Linear Mixture MDPs). An MDP instance $M = (S, A, H, P_{h\,h=}1^H), l_{k\,k=}1^K)$ is called an inhomogeneous, episodic B-bounded linear mixture MDP if there exists a known feature mapping $\phi(s'|s,a) : SAS \to R^d$ with $\phi(s'|s,a)_2 1$ and unknown vectors $\phi^*_{h\,h=1}{}^H \in R^d$ with $\phi^*_h{}_2 B$ such that for all $(s, a, s') \in SAS$ and $h \in [H]$, it holds that $P_h(s'|s,a) =< \phi(s'|s,a), \phi^*_h >$

2 Approximate MDP Solvers

Approximate MDP solvers aim to find a near-optimal policy when the exact transition matrix P and reward r are known. The simplest way to achieve this is by the Value Iteration (VI) algorithm, which solves the Bellman optimality equation in a dynamical programming fashion. Then the greedy policy induced by the result Q* gives precisely the optimal policy without error. Another popular approach frequently used in practice is the Natural Policy Gradient (NPG) algorithm. In each iteration, the algorithm first evaluates the value of policy $\pi^{(t)}$ using Bellman equation. Then it updates the policy by first scaling it with the exponential of learning times value $Q^{\pi(t)}$, and then performs a normalization. For completeness, we provide its guarantee here, which resembles the infinite horizon analysis in (Agarwal et al., 2019)

References

[1] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In International Conference on Machine Learning, pages 4870–4879. PMLR, 2020.

[2] Li, L. Sample complexity bounds of exploration. In Reinforcement Learning, pp. 175–204. Springer, 2012

[3] Ayoub, A., Jia, Z., Szepesv´ari, C., Wang, M., and Yang, L. (2020). Model-based reinforcement learning with value-targeted regression. In Proceedings of the 37th International Conference on Machine Learning (ICML), pages 463–474.

[4] Cai, Q., Yang, Z., Jin, C., and Wang, Z. (2020). Provably efficient exploration in policy optimization. In Proceedings of the 37th International Conference on Machine Learning (ICML), pages 1283–1294.

[5] Zhou, D., Gu, Q., and Szepesv´ari, C. (2021). Nearly minimax optimal reinforcement learning for linear mixture Markov decision processes. In Proceedings of the 34th Conference on Learning Theory (COLT), pages 4532–4576.

[6] He, J., Zhou, D., and Gu, Q. (2022). Near-optimal policy optimization algorithms for learning adversarial linear mixture MDPs. In Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS), pages 4259–4280.

[7] Li, L.-F., Zhao, P., and Zhou, Z.-H. (2023). Dynamic regret of adversarial linear mixture MDPs. In Advances in Neural Information Processing Systems 36 (NeurIPS), pages 60685–60711.

[8] Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. Optimality and approximation with policy gradient methods in markov decision processes. arXiv preprint arXiv:1908.00261, 2019.