Reward-Free Deep-Learning-Based Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Exploration is widely regarded as one of the most challenging aspects of reinforcement learning (RL). We consider the reward-free RL problem, which operates in two phases: an exploration phase, where the agent gathers exploration trajectories over episodes irrespective of any predetermined reward function, and a subsequent planning phase, where a reward function is introduced. The agent then utilizes the episodes from the exploration phase to calculate a near-optimal policy. Existing algorithms and sample complexities for reward-free RL are limited to tabular, linear, or very smooth function approximations, leaving the problem largely open for more general cases. We consider deep-learning-based function approximations, i.e. DQNs, and propose an algorithm based on internal feedback and the agent's own confidence and self-certainty in a graph MDP.

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 in order to investigate whether they hold high rewards. RL policies using complex function approximations have been empirically effective in various fields including reward-free RL. These RL policies must learn the transition model, either directly or indirectly, necessitating efficient exploration.

Sophisticated exploration strategies which deliberately incentivize the agent to visit new states are provably sample-efficient (c.f., Kearns Singh (2002); Brafman Tennenholtz (2002); Azar et al. (2017); Dann et al. (2017); Jin et al. (2018)), with recent work providing a nearly-complete theoretical understanding for maximizing a single prespecified reward function (Dann Brunskill, 2015; Azar et al., 2017; Zanette Brunskill, 2019; Simchowitz Jamieson, 2019). In practice, however, reward functions are often iteratively engineered to encourage desired behavior via trial and error (e.g. in constrained RL formulations (Altman, 1999; Achiam et al., 2017; Tessler et al., 2018; Miryoosefi et al., 2019)). In such cases, repeatedly invoking the same reinforcement learning algorithm with different reward functions can be quite sample inefficient.

One solution to avoid excessive data collection in such settings is to first collect a dataset with good coverage over all possible scenarios in the environment, and then apply a "Batch-RL" algorithm. To methodically study this problem, we concentrate on the reward-free RL framework, which includes an exploration phase and a planning phase. In the exploration phase, the agent interacts with the environment without any pre-determined rewards and gathers empirical trajectories over episodes for the subsequent planning phase. During the planning phase, the agent uses the offline data accumulated in the exploration phase to compute the optimal policy for a given extrinsic reward function r, without further interactions with the environment.

Submitted to the 18th European Workshop on Reinforcement Learning (EWRL 2025). Do not distribute.

The reward-free RL framework has been progressively examined under increasingly complex models —tabular → linear → kernel-based → deep learning based— in several works including (Jin et al., 2020a; Wang et al., 2020; Qiu et al., 2021). The existing literature adequately addresses the tabular and linear settings. It however tends to falter, providing only partial and incomplete results when dealing with the more intricate kernel-based and deep learning based settings. The contribution of this paper is to further the literature by providing order optimal results in the deep-learning-based setting.

In this paper, we aim to develop an end-to-end instantiation of this proposal. To this end we ask:

1. How can we efficiently integrate a reward-free RL framework with deep learning based settings, such as the DQNs algorithms? 2. How can agents efficiently explore the environment without explicit rewards?

Our main objective is designing algorithms for both exploration and planning phases in the reward-free RL framework with deep-learning-based modeling. In particular, by exploring the environment, we aim to gather sufficient information so that we can compute the near-optimal policies for any reward function.

Our Contributions. In this paper, we present the concept of intrinsic signals or self-certainty which characterize the sample complexity of achieving provably sufficient coverage for Batch-RL. We do so by adopting a "reward-free RL" paradigm using a graph MDP and representing every state with a weighted node: During an exploration phase, the agent collects trajectories from an MDP M without a pre-specified reward function but with intrinsic signals. Then, in the planning phase, it is tasked with computing near-optimal policies under the transitions of M for a large collection of given reward functions using the DQN algorithm.

2 Related Work

The reward-free RL framework under the episodic setting has been studied with tabular model in Jin et al. (2020a); Zhang et al. (2020); Menard et al.(2021); Kaufmann et al. (2021), and with linear model in Wang et al. (2020); Zanette et al. (2020c); Wagenmaker et al. (2022). The problem has also been studied under the linear mixture model in Zhang et al. (2021); Chen et al. (2021); Zhang et al. (2023). The sample complexity of the RL problem on a discounted MDP setting with an infinite horizon has been considered under various tabular, linear, and kernel-based settings in (Kearns Singh, 1998; Azar et al., 2013; Sidford et al., 2018; Agarwal et al., 2020; Yang Wang, 2019; Yeh et al., 2023). These works however assume the existence of a generative oracle (Kakade, 2003), which provides sample transitions from any state-action pair of the algorithm's choice. This assumption simplifies the problem compared to the reward-free RL framework considered in this work, where the agent must follow the MDP trajectory within each episode and cannot arbitrarily inquire transitions from state-action pairs.

Specifically, we design an exploration algorithm based on intrinsic signals obtained from the agent itself that add significant challenges to the analysis. Our algorithm design is inspired by the RLIF technique used in Zhao et al (2025). In comparison, Zhao et al (2025) considered reasoning in LLMs where they replace external rewards in Group Relative Policy Optimization (GRPO) with self-certainty scores, enabling fully unsupervised learning. That is different from the reward-free RL framework considered in this work and their results do not apply here.

There is extensive literature on the analysis of RL policies which does not rely on a generative model or an exploratory behavioral policy. The literature has primarily focused on the tabular setting (Jin et al., 2018; Auer et al., 2008; Bartlett Tewari, 2012). Recent literature has placed a notable emphasis on employing function approximation in RL, particularly within the context of generalized linear settings. This approach involves representing the value function or transition model through a linear transformation to a well-defined feature mapping. Important contributions include the work of Jin et al. (2020b); Yao et al. (2014), as well as subsequent studies by Russo (2019); Neu Pike-Burke (2020); Yang Wang (2020). Furthermore, there have been several efforts to extend these techniques to a kernelized setting, as explored in Yang et al. (2020a); Yang Wang (2020); Chowdhury Gopalan (2019); Yang et al. (2020b); Domingues et al. (2021).

3 Problem Formulation

In this section, we present the episodic graph MDP setting, the reward-free RL framework, and background on DQNs method.

3.1 Graph-based MDPs

We assume that the full state x can be represented as a collection of state variables xi, so that X is a Cartesian product of the domains of the $xi : X = X_1 X_2 X_N$, and similarly for $d : D = D1D2DN$. We consider the following particular factored form for MDPs: for each variable i, there exist neighborhood sets γ_i (including i) such that the value of $X_i^t + 1$ depends only on the variable i's neighborhood, $x^t[\gamma_i]$, and the ith decision d_i^t. Then, we can write the transition function in a factored form:

$$T(y|x,d) = \prod_{i=1}^{N} T_i(y_i|x[\gamma_i], d_i \tag{1}$$

where each factor is a local-scope function $T_i : X[\gamma_i]D_iX_i \to [0,1], \forall i \in 1,2,...,N$. We also assume that the reward function is the sum of N local-scope rewards:

$$R(x,d) = \sum_{i=1}^{N} R_i(x_i, d_i) \tag{2}$$

with local-scope functions $R_i : X_iD_ißR, \forall i \in 1,2,...,N.$ To summarize, a graph-based Markov decision process is characterized by the following parameters: $(X_i : 1iN; D_i : 1iN; R_i : 1iN; \gamma_i : 1iN; T_i : 1iN)$. These assumptions for graph-based MDPs can be easily generalized, for example to include T_i and R_i that depend on arbitrary sets of variables and decisions, using some additional notation.

The optimal policy $\pi(x)$ cannot be explicitly represented for large graph-based MDPs, since the number of states grows exponentially with the number of variables. To reduce complexity, we consider a particular class of local policies: a policy $\pi(x) : X \to D$ is said to be local if decision d_i is made using only the neighborhood γ_i, so that $\pi(x) = (\pi 1(x[\gamma_1]), \pi_2(x[\gamma_2]), ..., \pi_N(x[\gamma_N]))$ where $\pi_i(x[\gamma_i]) : X[\gamma_i] \to D_i$. The main advantage of local policies is that they can be concisely expressed when the neighborhood sizes $|\gamma_i|$ are small.

3.2 Reward-Free RL Framework

We aim to learn E-optimal policies using as small as possible number of collected exploration episodes. In particular, we consider the reward-free RL framework that consists of two phases: exploration and planning. In the exploration phase, we collect N exploration episodes $(s_1^n, a_1^n, s_2^n, a_2^n, , s_H^n)_{n=1}N$ without any revealed reward function. Then, in the planning phase, reward r is revealed, and we design a policy for reward r using the trajectories collected in the exploration phase. We refer to N as the sample complexity of designing E-optimal policy. Under certain assumptions, the question is: How many exploration episodes are required to obtain E-optimal policies?

3.3 Deep Q-Learning

We are interested in maximizing the expected total reward in the episode, starting at step h, i.e., the value function, defined as

$$V(s)_h^\pi = E[\Sigma_{h'=h} H r_h'(s_h', a_h')|s_h = s], \forall s, h \in [H], \tag{3}$$

where the expectation is taken with respect to the randomness in the trajectory $(s_h, a_h)_{h=1}H$ obtained by the policy π We also define the state-action value function $Q_h\pi : Z \to [0,H]$ as follows.

$$Q_h^\pi(s,a) = E_\pi[\Sigma_{h'=h} H r_h'(s_h', a_h')|s_h = s, a_h = a] \tag{4}$$

where the expectation is taken with respect to the randomness in the trajectory $(s_h, a_h)_{h=1}H$ obtained by the policy π. The Bellman equation associated with a policy π then is represented as

$Q_h^{\pi}(s,a) = r_h(s,a) + [P_h V_{h+1}^{\pi}](s,a),$
$V_h^{\pi}(s) = max Q_h^{\pi}(s,a), V_{H+1}^{\pi} = 0$

where the expectation is taken with respect to the randomness in the policy π.

4 Algorithm

The two main ideas in our design are (i) the use of a intrinsic signals in the exploration phase and (ii) DQN integration setting in application of deep-learning-based confidence intervals.

Intrinsic Rewards. In the exploration phase, Instead of depending on external evaluation, IR uses the model's own assessment of its outputs or reasoning process as feedback. This offers several advantages: it reduces reliance on supervision infrastructure, provides task-agnostic reward signals, and supports learning in domains where external verification is unavailable, where u(q, o) represents an intrinsic signal derived from the model's internal state or computation, rather than external verification. The key challenge lies in identifying intrinsic signals that correlate with output quality and can effectively guide learning.

4.1 Exploration Phase

Algorithm 1 Exploration Phase

Input: τ, $\beta(\delta)$, K, S, A, H, P
for $n = 1, 2, \ldots$ **do**
for $step = H, 1 \ldots$ **do**
Obtain $Q_h n$
$V_h^n(.) = max_a Q_h(., a)$
end for
for $h = 1, 2, \ldots$ **do**
Take action $a_h^n \leftarrow max_a Q_h^n(s_h^n, a)$
Receive the next state s_{h+1}^n
end for
end for

4.2 Planning Phase

Algorithm 2 Exploration Phase

Input: τ, $\beta(\delta)$, K, S, A, H, P and exploration data $(s_h^n, a_h^n)_{(h,n)} \in [H][N]$
for $all(s; a; s_0; h) \in SAS[H]$ **do**
$\hat{P}_h(s'|s,a) =$N$_h(s,a,s')/N_h(s,a)$
end for
$\pi \leftarrow$ APPROXIMATE-MDP-SOLVER($\hat{P}$; r; e)
Return$Policy \hat{\pi}$

5 Conclusion

In this paper, We considered the reward-free RL framework with deep-learning-based modeling, comprising of two phases. In the exploration phase, the learner first collects trajectories from an MDP M without receiving any reward information. After the exploration phase, the learner is no longer allowed to interact with the MDP and she is instead tasked with computing near-optimal policies under for M for a collection of given reward functions. This framework is particularly suitable when there are many reward functions of interest, or when we are interested in learning the transition operator directly. Finally, we developed algorithms for both exploration and planning phases for with function approximation using deep learning.

References

References follow the acknowledgments in the camera-ready paper. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

[1] Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. Machine learning, 49(2-3):209–232, 2002.

[2] Brafman, R. I. and Tennenholtz, M. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. Journal of Machine Learning Research, 3(Oct):213–231, 2002..

[3] Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In Proceedings of the 34th International Conference on Machine LearningVolume 70, pp. 263–272. JMLR. org, 2017.

[4] Dann, C., Lattimore, T., and Brunskill, E. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In Advances in Neural Information Processing Systems, pp. 5713–5723, 2017.

[5] Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In Advances in Neural Information Processing Systems, pp. 4863–4873, 2018.

[6] Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. In Advances in Neural Information Processing Systems, pp. 2818–2826, 2015.

[7] Zanette, A. and Brunskill, E. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In International Conference on Machine Learning, pp. 7304–7312, 2019.

[8] Simchowitz, M. and Jamieson, K. G. Non-asymptotic gapdependent regret bounds for tabular mdps. In Advances in Neural Information Processing Systems, pp. 1151– 1160, 2019.

[9] Altman, E. Constrained Markov decision processes, volume 7. CRC Press, 1999.

[10] Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 22–31. JMLR. org, 2017

[11] Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. In International Conference on Learning Representations, 2018.

[12] Miryoosefi, S., Brantley, K., Daume III, H., Dudik, M., and Schapire, R. E. Reinforcement learning with convex constraints. In Advances in Neural Information Processing Systems, pp. 14093–14102, 2019.

[13] Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. In International Conference on Machine Learning, pp. 4870– 4879. PMLR, 2020a.

[14] Wang, R., Du, S. S., Yang, L., and Salakhutdinov, R. R. On reward-free reinforcement learning with linear function approximation. Advances in neural information processing systems, 33:17816–17826, 2020.

[15] Qiu, S., Ye, J., Wang, Z., and Yang, Z. On reward-free rl with kernel and neural function approximations: Singleagent mdp and markov game. In International Conference on Machine Learning, pp. 8737–8747. PMLR, 2021.

[16] Zhang, Z., Du, S. S., and Ji, X. Nearly minimax optimal reward-free reinforcement learning. arXiv preprint arXiv:2010.05901, 2020.

[17] Menard, P., Domingues, O. D., Jonsson, A., Kaufmann, ´ E., Leurent, E., and Valko, M. Fast active learning for pure exploration in reinforcement learning. In International Conference on Machine Learning, pp. 7599–7608. PMLR, 2021.

[18] Agarwal, A., Kakade, S., and Yang, L. F. Model-based reinforcement learning with a generative model is minimax optimal. In Conference on Learning Theory, pp. 67–83. PMLR, 2020

[19] Yeh, S.-Y., Chang, F.-C., Yueh, C.-W., Wu, P.-Y., Bernacchia, A., and Vakili, S. Sample complexity of kernelbased q-learning. In International Conference on Artificial Intelligence and Statistics, pp. 453–469. PMLR, 2023.

[20] Chowdhury, S. R. and Gopalan, A. On kernelized multiarmed bandits. In International Conference on Machine Learning, pp. 844–853. PMLR, 2017.

[21] Neu, G. and Pike-Burke, C. A unifying view of optimism in episodic reinforcement learning. In Advances in Neural Information Processing Systems, volume 33, pp. 1392–1403, 2020

[22] Yao, H., Szepesvari, C., Pires, B. A., and Zhang, X. Pseudo- ´ MDPs and factored linear action models. In 2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), pp. 1–9. IEEE, 2014.

[23] Zhao, X., Kang, Z., Feng, A., Levin, S., Song, D. Learning to Reason without External Rewards, 2025.