Open Problem: Order Optimal Regret Bounds for Non-Markovian Rewards

Aya Shabbar*
Department of Mechatronics Engineering
Tishreen University
aya.shabar@tishreen.edu.sy

Abstract

The standard RL world model is that of a Markov Decision Process (MDP) that assumes Markovian transitions and rewards. Yet, many real-world rewards are non-Markovian. A basic premise of MDPs is that the rewards depend on the last state and action only. Some problem settings involve "double-state" or non-Markovian reward functions where the reward depends on the trajectory. Past work considered the problem of modeling and solving MDPs with non-Markovian rewards (NMR), but we know of no principled approaches for RL with NMR. This approach is particularly interesting as it naturally extends the MDP structure. Thus, we will address the problem of policy learning from experience with such rewards. This exacerbates the misalignment between theoretical researchers and practitioners. An open problem is to develop algorithms that can efficiently solve such problems and provide provable regret bounds, even with knowledge of the transition model. We will highlight this open problem and discuss related challenges.

1 Introduction

In MDPs, rewards depend on the last state and action only. However, many real-world settings are non-Markovian. An RL agent that attempts to learn in such domains without realizing that the rewards are non-Markovian will display sub-optimal behavior. We can address non-Markovian rewards (NMRs) by augmenting the state with information that makes the new model Markovian. Except for pathological cases, this is always possible. Indeed, early work on this topic described various methods for efficiently augmenting the state to handle NMRs [2], [19]. Another work that leveraged our better understanding of the relationship between NMRs specified using temporal and dynamic logics and automata to provide even better methods [6], [4]. Yet, such model transformations are performed offline with complete knowledge of the model and the reward structure, and are not useful for a reinforcement learning agent equipped with some fixed state model and state variables. For a sharp and clear presentation of our open problem, we focus on Non Markovian Reward Decision Processes (NMRDPs) under the regret performance measure. However, similar problems can be raised in other settings, including infinite horizon discounted or undiscounted MDPs within an online or offline framework. We specifically ask: *Is it possible to design order-optimal or, at the very least, no-regret learning algorithms under reasonable assumptions on Markovian transitions and rewards structure that can solve such problems and provide regret bounds, even with the knowledge of the transition model?* In this paper, we will formally present this open problem, provide an overview of existing work, and discuss some of the challenges involved.

*Decision Making and Reinforcement Learning Lab

2nd Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2025).

2 MDPs, RL, and NMR

RL is usually formulated in the context of a Markov Decision Process (MDP) $M = <S, A, Tr, R>$. S is the set of states, A is the set of actions, $Tr\colon S \times A \to \pi(S)$ is the transition function that returns for every state s and action a a distribution over the next state. $R\colon S \times A \to R$ is the reward function that specifies the real-valued reward received by the agent when applying action a in state s. A solution to an MDP is a policy that assigns an action to each state, possibly conditioned on past states and actions. The value of policy ρ at s, $v^{\rho}(s)$, is the expected sum of (possibly discounted) rewards when starting at s and selecting actions based on ρ . Focusing on the infinite horizon, discounted reward case, we know that every MDP has an optimal policy, ρ^* that maximizes the expected discounted sum of rewards for every starting state $s \in S$, and that an optimal policy that is stationary and deterministic $\rho^*\colon S \to A$ exists [18]. The value function obeys Bellman's optimality condition [3]

$$v_h^{\pi}(s) = \mathbb{E}[\Sigma_{h'=h}^{H} r'_h(s'_h, a'_h) | s_h = s] \forall s \in S, h \in [H] \tag{1}$$

Non-Markovian Rewards In the canonical RL problem setup of a Markov decision process (MDP), rewards depend only on the most recent state-action pair. In a non-Markovian reward decision process (NMRDP) [2], rewards depend on the full preceding trajectory [2]. NMRDPs can be expanded into MDPs (and thus solved by RL) by augmenting the state with a hidden state that captures all reward relevant historical information, but this is typically not known a priori. Data-driven approaches to learning NMRDP expansions often make use of domain-specific propositions and temporal logic operators [19]. Outside of the RM context, recurrent architectures such as LSTMs have been used in NMRDPs to reduce reliance on pre-specified propositions [10]. They also have a long history of use in partially observable MDPs, where dynamics are also non-Markovian [21].

Non-Markovian Reward Decision Processes (NMRDPs) [2] extend MDPs to allow for rewards that depend on the entire history. It has been observed by much past work that many natural rewards are non-Markovian. A general NMR is a function from $(S \times A)^*$to R. However, this definition is too complex to be of practical use because of its infinite structure. For this reason, past work focused on properties of histories that are finitely describable, and in particular, logical languages that are evaluated w.r.t. finite traces of states and actions, representing the agent's history. The performance of a learning algorithm $\{\pi_t\}_{t\in[T]}$ is measured in terms of the total loss in the value function, referred to as regret, denoted by R(T) in the following definition:

$$R(t) = \Sigma_{t=1}^{T}(V_1^*(s_1, t) - V_1^{\pi_t}(s_1, t)) \tag{2}$$

A learning algorithm with sublinear regret in T is often referred to as a no regret algorithm, since the average regret over T tends toward zero as T increases. This implies that the value of the policy executed by the learning algorithm converges to that of the optimal policy over episodes. For a value function $V\colon S \to R$, and a conditional distribution $P(s|z), s \in S, z \in Z$, we define the notation $[PV](z) = \mathbb{E}_{s\sim P(|z)}[V(s)]$. The state-action value function $Q_h^{\pi}\colon Z \to [0, H]$ is defined as follows: $Q_h^{\pi}(s, a) = \mathbb{E}_{\pi}[\Sigma_{h'=h}^{H} r'_h(s'_h, a'_h) | s_h = s, a_h = a]$, 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: $Q_h^{\pi}(s, a) = r_h(s, a) + [P_h V_{h+1}^{\pi}](s, a)$, $V_h^{\pi}(s) = max_{a\in A} Q_h^{\pi}(s, a), V_{H+1}^{\pi} \equiv 0$

3 Formal Definition of Non-Markovian RM

Consider an agent interacting with an environment with Markovian dynamics. At discrete time t, the current environment state $s_t \in S$ and agent action $a_t \in A$ condition the next environment state s_{t+1} according to the dynamics function $D\colon S \times A \to \Delta S$. A trajectory $\xi \in \Xi$ is a sequence of state-action pairs, $\xi = ((s_0, a_0), \ldots, (s_{T-1}, a_{T-1}))$, and a human's preferences about agent behaviour respect a real-valued return function $G\colon \Xi \to R$. In traditional (Markovian) RM, return is assumed to decompose into a sum of independent rewards over state-action pairs, $G(\xi) = \Sigma_{t=0}^{T-1} R(S_t, a_t)$ and the aim is to reconstruct $R\prime \approx R$ from possibly-noisy sources of preference information. In our generalised non-Markovian model, we consider the human to observe a trajectory sequentially and allow for the possibility of hidden state information that accumulates over time and parameterises R:

$$G(\xi) = \Sigma_{t=0}^{T-1} R(S_t, a_t, h_{t+1}) \tag{3}$$

where $h_{t+1} = \delta(h_t, s_t, a_t)$

δ is a hidden state dynamics function, and h_0 is a fixed value for the initial hidden state. Reconstruction of the human's preferences now requires the estimation of $\delta' \approx \delta$ and $h'_0 \approx h_0$ alongside $R' \approx R$.

The hidden state h may be interpreted as (3) an external feature of the environment that is detectable by the human but excluded from the state, or (4) a psychological feature of the person themselves, through which their response to each new observation is influenced by what they have seen already. The latter framing is more interesting for our purposes, and connects to the psychological literature on human judgment, memory, and biases. In practice, hidden state information may encode the human's preferences about the order in which a sequence of behaviors should be performed, the effect of historic observations on their subjective mood (and in turn on their reward evaluations), or cognitive biases which corrupt the way they aggregate instantaneous rewards into trajectory-level feedback. All of these complications are liable to arise in practical RM applications, but cannot be handled when the Markovian reward assumption is made.

In this work, we focus on one of the simplest and most explicit forms of preference information: direct labelling of returns $G(\xi_i)$ for a dataset of N trajectories $\xi_{i=1}^{N}$. We aim to solve the reconstruction problem by minimizing the squared error in predicted returns:

$$argmin\Sigma_{i=1}^{N}(G(\xi_i) - \Sigma_{t=0}R'(s_{i,t}, a_{i,t}, h'_{i,t+1}))^2 \tag{4}$$

where $h\prime_{i,0} = h\prime_0$ and $h'_{i,t+1} = \delta'(h'_{i,t+1}, s_{i,t}, a_{i,t})\ \forall_i \in 1..N$

We observe that Equation 2 perfectly matches the definition of a MIL problem. Each trajectory ξ_i can be considered as an ordered bag of instances $((s_{i,0}, a_{i,0}), ..., (s_i, T_{i-1}, a_i, T_{i-1}))$ with unobserved instance labels $R(s_{i,t}, a_{i,t}, h_{i,t+1})$, an observed bag label $G(\xi_i) = \Sigma_{t=0}^{T-1}R(s_{i,t}, a_{i,t}, h_{i,t+1})$, and temporal instance interactions via the changing hidden state $h_{i,t}$. This correspondence motivates us to review the space of existing MIL models (specifically those that model temporal dependencies among instances) to provide a starting point for developing our non-Markovian RM approach.

4 Open Problem

The question we ask is as follows: *Consider the NMRDPs setting described in Section 2. (a) Can a no-regret learning algorithm be designed? (b) What is the minimum regret growth rate with T (and also H)? And, can a learning algorithm be designed to achieve order-optimal (or near-optimal) regret performance, closely aligning with the established lower bound?*

5 Technical Overview

In this section, we give an overview of the technical challenges associated with obtaining the minimax optimal regret bound for RL with trajectory feedback, together with our approaches to tackle these challenges.

Connection with Linear Bandits As observed in prior work on RL with trajectory feedback [8], when the transition model, RL with trajectory feedback can be seen as an instance of linear bandits. More specifically, in each round, suppose the trajectory sampled by the agent is τ, the expected trajectory reward feedback would be ϕ_τ^T R, i.e., a linear function with respect to ϕ_τ. Based on this observation, [8] showed how to build appropriate confidence regions for RL with trajectory feedback by adapting analysis for linear bandits algorithms, and obtained a regret bound of $O(\sqrt{S^2A^2H^4K})$ Although it is plausible to improve their regret bound to $O(\sqrt{S^2AH^3K})$ by a more refined analysis, it is unclear how to improve the order of S in their regret bound. Indeed, in the work of [8], RL with trajectory feedback is naively treated as an instance of linear bandits with feature dimension $d = SAH$, and the best known regret bound for any linear bandits algorithm is $O(d\sqrt{\tau})$ [7], or $O(\sqrt{dTlogK})$ for linear bandits with K arms [5]. Since there are A^{SH} policies for an MDP, and each of them can be seen as an arm in the linear bandits problem instance, improving the order of S in the regret bound of prior work requires fundamentally new ideas.

Tighter Confidence Region Based on Trajectories n order to achieve a minimax optimal regret bound, our first new idea is to build a tighter confidence region by exploiting structures of the linear bandits instance associated with RL with trajectory feedback. Before getting into more details, we first review least squares regression (LSR), an estimator commonly used in linear bandits algorithms

(also in prior work on RL with trajectory feedback [8]) based on the principle of optimism in the face of uncertainty.

Given a set of data points π^t, τ^t, Y^t, where for each $1 \leq t \leq T$, where π^t the policy used in the t-th round, τ^t is the trajectory sampled by executing π^t and the Y^t is the trajectory reward feedback. Clearly, $\mathbf{E}[Y_t] = \phi_{\tau^t} R$,

$$R = argmin \Sigma_{t=1}(Y^t - \phi_{\tau^t} r) + \lambda \left\| R \right\|_2^2 = \Lambda^{-1} \Sigma_{t=1} \phi_{\tau^t} Y^t, \tag{5}$$

where $\Lambda = \lambda \mathbf{I} + \Sigma_{t=1} \phi_{\tau^t} \phi_{\tau^t}$ is the information matrix. Optimism-based linear bandits algorithms typically maintain a set of arms, and eliminate arms outside the confidence region during the execution of the algorithm. For RL with trajectory feedback, each arm in the linear bandits instance corresponds to a deterministic policy in the original MDP. Our construction of the tighter confidence region is based on the following two key observations:

- Although the total number of deterministic policies could be as large as ASH, the number of trajectories is $|T| = (SA)^H$;
- For any deterministic policy π $d_p^\pi = \Sigma_{\tau \in T} Pr_{\pi,p}[T].\phi^\tau$ is a convex combination of $\phi_{\tau \tau \in T}$.

Based on these observations, instead of building confidence region for $|(d_p^\pi)^T(\hat{R} - R)|$ for each deterministic policy π and applying a union bound over all policies which result in suboptimal regret bounds, we consider the following event

$$\epsilon \coloneqq |\phi_\tau^T(\hat{R} - R)| \leq C(min\sqrt{\phi_\tau^T \Lambda^{-1} \phi_\tau \sigma^2 log(2|\tau|/\delta)}, H) \forall \tau \in T \tag{6}$$

where C is some proper constant, and $\sigma^2 \leq H$ is a constant such that $Y^t - {\phi_{\tau^t}^T}_{t=1}^T$ is a group of independent zero-mean σ^2-subgaussian random variables. By standard concentration arguments, ϵ holds with probability at least $1 - \delta$. We assume ϵ holds in the remaining part of the discussion. Note that second observation states that $d_p^\pi = \Sigma_\tau \in T Pr_{\pi,p[T]}.\phi_\tau$ which implies that

$$|(d_p^\pi)^T(\hat{R} - R)| \leq \Sigma_{\tau \in T} Pr_{\pi,p[T]}.\phi_\tau |\phi_\tau^T(\hat{R} - R)|$$

$$\leq O(\Sigma_{\tau \in T} Pr_{\pi,p[T]} min\sqrt{\phi_\tau^T \Lambda^{-1} \phi_\tau \sigma^2 log(2|\tau|/\delta)}, H)$$

$$\leq (H\sqrt{\Sigma_{\tau \in T} Pr_{\pi,p[T]} min \phi_\tau^T \Lambda^{-1} \phi_\tau}, 1) \tag{7}$$

for any policy π, where the last step holds by Cauchy-Schwarz inequality and the fact that $|T| = (SA)^H$.

6 Related Work

A number of authors have recently emphasized the fact that many desired behaviors are non-Markovian. [15], [14] discussed the need for more elaborate reward specification for RL for robotics. He considered scenarios where RL is used to learn a policy in an unknown world where the reward is not intrinsic to the world, but is specified by the designer, usually in a high-level language. For this, he proposed the use of a temporal logic called GLTL. Similarly, [13] use truncated LTL as a reward specification language. In all above papers, reward specification is part of the input. We are interested in RL when the reward model is unknown. Some of the work on learning in partially observable environments can be viewed as indirectly addressing this problem. For example, classical work on Q-learning with memory [17] maintains and updates an external memory. Thus, it essentially learns an extended state representation. But if we realize that the additional reward variables are essentially states of an automaton, we can apply a more principled approach to learn these automata using state-of-the-art automata learning algorithms. Not surprisingly, Angluin's famous automata learning algorithm [1] and early work on learning in unobservable environments (e.g., [16]) have a similar flavor of identifying states with certain suffixes of observations. Unfortunately, exact learning of a target DFA from an arbitrary set of labeled examples is hard [9], and under standard cryptographic assumptions, it is not PAC-learnable [11]. Thus, provably efficient automata learning is possible

only if additional information is provided, as in Angluin's model that includes a teacher. However, there is much work on practical learning algorithms, and in our experimental evaluation, we use the FlexFringe implementation of the well known EDSM algorithm [12], [20].

References

[1] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987.

[2] Fahiem Bacchus, Craig Boutilier, and Adam Grove. Rewarding behaviors. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1160–1167, 1996.

[3] Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Princeton university press, 2015.

[4] Ronen Brafman, Giuseppe De Giacomo, and Fabio Patrizi. Ltlf/ldlf non-markovian rewards. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[5] Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41–1. JMLR Workshop and Conference Proceedings, 2012.

[6] Alberto Camacho, Oscar Chen, Scott Sanner, and Sheila A McIlraith. Decision-making with non-markovian rewards: From ltl to automata-based reward shaping. In *Proceedings of the Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, pages 279–283, 2017.

[7] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory*, number 101, pages 355–366, 2008.

[8] Yonathan Efroni, Nadav Merlis, and Shie Mannor. Reinforcement learning with trajectory feedback. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7288–7295, 2021.

[9] E Mark Gold. Complexity of automaton identification from given data. *Information and control*, 37(3):302–320, 1978.

[10] Firas Jarboui and Vianney Perchet. Trajectory representation learning for multi-task nmrdp planning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6786–6793. IEEE, 2021.

[11] Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

[12] Kevin J Lang, Barak A Pearlmutter, and Rodney A Price. Results of the abbadingo one dfa learning competition and a new evidence-driven state merging algorithm. In *International Colloquium on Grammatical Inference*, pages 1–12. Springer, 1998.

[13] Xiao Li, Cristian-Ioan Vasile, and Calin Belta. Reinforcement learning with temporal logic rewards. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3834–3839. IEEE, 2017.

[14] Michael L Littman, Ufuk Topcu, Jie Fu, Charles Isbell, Min Wen, and James MacGlashan. Environment-independent task specifications via gltl. *arXiv preprint arXiv:1704.04341*, 2017.

[15] Michael Lederman Littman. Programming agent via rewards. *Invited talk at IJCAI*, 2015.

[16] R Andrew McCallum. Instance-based state identification for reinforcement learning. *Advances in Neural Information Processing Systems*, 7, 1994.

[17] Leonid Peshkin, Nicolas Meuleau, and Leslie Kaelbling. Learning policies with external memory. *arXiv preprint cs/0103003*, 2001.

[18] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[19] Sylvie Thiébaux, Charles Gretton, John Slaney, David Price, and Froduald Kabanza. Decision-theoretic planning with non-markovian rewards. *Journal of Artificial Intelligence Research*, 25:17–74, 2006.

[20] Sicco Verwer and Christian A Hammerschmidt. Flexfringe: a passive automaton learning package. In *2017 IEEE international conference on software maintenance and evolution (ICSME)*, pages 638–642. IEEE, 2017.

[21] Daan Wierstra, Alexander Förster, Jan Peters, and Jürgen Schmidhuber. Recurrent policy gradients. *Logic Journal of IGPL*, 18(5):620–634, 2010.