Information-theoretic Algorithms for Model-free Reinforcement Learning

by

Farrell Eldrian S. Wu

S.B. Computer Science and Engineering, Business Analytics, Massachusetts Institute of Technology, 2021

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2023

Authored by: Farrell Eldrian S. Wu
Department of Electrical Engineering and Computer Science
August 25, 2023

Certified by: Vivek F. Farias
Professor of Operations Management, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Information-theoretic Algorithms for Model-free Reinforcement Learning

by

Farrell Eldrian S. Wu

Submitted to the Department of Electrical Engineering and Computer Science
on August 25, 2023 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

In this work, we propose a model-free reinforcement learning algorithm for infinte-horizon, average-reward decision processes where the transition function has a finite yet unknown dependence on history, and where the induced Markov Decision Process is assumed to be weakly communicating. This algorithm combines the Lempel-Ziv (LZ) parsing tree structure for states introduced in [4] together with the optimistic Q-learning approach in [9]. We mathematically analyze the algorithm towards showing sublinear regret, providing major steps towards the proof of such. In doing so, we reduce the proof to showing sub-linearity of a key quantity related to the sum of an uncertainty metric at each step. Simulations of the algorithm will be done in a later work.

Thesis supervisor: Vivek F. Farias
Title: Professor of Operations Management

Acknowledgments

I would like to thank Vivek Farias, my thesis advisor, for his guidance over the past year on this interesting project. I learned a lot about the research process in this field which will be invaluable for the coming years in the PhD and which makes me excited about the prospect of continuing this line of research.

I would like to thank Yang Liu and Mark Sellke for the chats on the combinatorial aspects of this project. Thank you as well for being inspiring mentors ever since my IMO days.

Finally, thanks to all the wonderful faculty at MIT — especially Guy Bresler, Leslie Kaelbling, James Orlin, Patrick Jaillet, and Ronitt Rubinfeld — whose classes sparked my passion in the subjects of machine learning, probability, statistics, and optimization.

Contents

Chapter 1

Introduction

This thesis is on designing model-free reinforcement learning (RL) algorithms for infinite-horizon decision processes in stationary environments where the transition function has a finite yet unknown dependence on history. The objective is to design general RL agents with *sublinear regret* under this setting. Our approach will center on combining ideas from optimistic Q-learning and from data compression algorithms such as Lempel-Ziv to efficiently summarize information from states.

1.1 Mathematical Formulation and Assumptions

We use the standard RL setup, similar to [3] and [4], where an agent interacts with an environment over time steps $t \in \mathbb{Z}^+$. At time t, the agent selects action A_t from an alphabet $\mathbb{A}$ of actions and then observes a response X_t from alphabet $\mathbb{X}$ of states, with both sets $\mathbb{A}$ and $\mathbb{X}$ being fixed over time. (To avoid trivial models, assume that both alphabets have size at least 2.) [1] The interaction produces a stream of actions and states $(A_1, X_1, A_2, X_2, \ldots)$. Denote the history $H_t = (A_1, X_1, \ldots, A_t, X_t)$ as the actions and responses through time t, and $H_i^j = (A_i, X_i, \ldots, A_j, X_j)$ as the history from $t = i$ to $t = j$, inclusive.

To represent a *finite yet unknown dependence on history*, environment dynamics involve a parameter k which specifies the maximum dependence on history. Then, it is assumed that there exists a transition kernel $\mathbb{P} = P(X_t|H_{t-k}^{t-1}, A_t)$, fixed across time, that specifies the probabilities for the response X_t over $\mathbb{X}$ given the most recent action as well as the last k (action, response) pairs in history. All draws from $P(X_t|H_{t-k}^{t-1}, A_t)$ are assumed to be mutually independent. The environment also generates a reward $R_t = R(H_{t-k}^{t-1}, A_t) \in [0, 1]$

[1] The choice of using "response" rather than "state" here is made to prevent confusion with the "state" in the associated Markov Decision Process (MDP) formulation.

that is a deterministic function of the current action and the previous k such pairs in history.[2]

The design of an agent involves formulating a policy $\pi(H_t)$ that selects an action $a \in \mathcal{A}$ based on the history, potentially from the very start of the process. However, an agent that efficiently learns will likely involve learning the parameter k, therefore not needing the whole history. We aim to minimize the infinite-horizon average cost, i.e.

$$\lambda_\pi = \liminf_{T\to\infty} \mathbb{E}_\pi \left[\frac{1}{T} \sum_{t=1}^{T-1} R_t \right].$$

Writing λ^* to be the maximum λ_π over all policies π, we aim to the regret R_π with respect to an optimal policy. In the finite (episodic) case, as a policy's design may depend on the number of time steps the interaction will run for, we allow π to be parameterized by T, as π^T. For example, a larger total duration may involve a longer horizon for exploration before exploitation. We thus define regret, which we aim to be sublinear in T:

$$\text{Regret}_{\pi^T}(T) = \mathbb{E}_{\pi^T} \left[\sum_{t=1}^{T} (\lambda^* - R_t) \right].$$

Similar to the setting in [4], the agent has no knowledge of $\mathbb{P}$ or even k. However, in order to aid analysis of a policy in this setting relative to λ^*, we consider the Markov Decision Process (MDP) induced by the given dynamics. Such a MDP can be formulated by specifying the state S_t at time t to be the history H_{t-k}^{t-1}. To confirm that it is indeed a MDP, note that the distribution of $S_{t+1} = H_{t-k+1}^{t}$ is uniquely specified given the previous state $S_t = H_{t-k}^{t-1}$ and the action A_t. Examining the contents of $S_{t+1} = H_{t-k+1}^{t}$, the distribution of X_t is specified by the transition function, while A_t and H_{t-k+1}^{t-1} are already conditioned on. This MDP has $(|\mathbb{A}|\,|\mathbb{X}|)^k$ states and $|\mathbb{A}|$ choices for the action from each state. However, for each (state, action) pair, there are only $|\mathbb{X}|$ possible immediate next states.

We assume that this MDP is *weakly communicating*, which by standard MDP theory such as in [1], ensures that λ^* is the same regardless of the starting conditions. This assumption, which is also used in [4] and [9], is reasonable as it is essentially required for any meaningful comparison across policies.

1.1.1 Example and Applications

A prototypical example, which is also exhibited in [4], and is designing an agent that plays Rock-Paper-Scissors. Here, the environment is an opponent that maintains a fixed strategy

[2] This formulation allows us to interpret the "state" as the previous k (action, response) pairs.

consistent with the mathematical formulation above, meaning that it draws from a random distribution determined by a fixed length of recent history. Knowing the opponent's strategy will allow computation of the optimal solution through standard dynamic programming methods for the associated MDP. We need, however, a learning algorithm that allow k being unknown, which the MDP approach does not accommodate.

The design of such a model-free agent may also contribute to the broader understanding of the role of information theory in RL. Algorithms in RL vary in the nature of states stored. For example, most model-based approaches will involve data structures that will only depend on input parameters, and will not substantially vary over vastly different transition functions. Model-free algorithms have the potential to form data structures that properly characterize the information captured in the transition function. This direction may be explored in future work extending from this thesis.

1.2 Related Work

1.2.1 Model-free RL and Optimistic Q-learning

The most well-known form of model-free RL is Q-learning, though it is most extensively analyzed in the discounted-reward rather than the average-reward setting. Wei et. al (2019) [9] proposes a model-free RL algorithm for this setting through an optimistic algorithm. An extended framework of model-free RL that involves transitions on subset of history called the *aleatoric state* is discussed in [7].

Some recent literature ([3], [9]) on model-free RL uses the technique of optimistic Q-learning, which involves both a time-varying bonus function to the temporal difference term in the update, and a non-increasing value function. This use of optimism is a smooth version of the traditional ϵ-greedy approach towards the exploration-exploitation trade-off, and is related to Thompson sampling, which is applied in [8] to a similar problem. Another related technique is that of the Upper Confidence Bound (UCB), the efficiency of which in Q-learning is examined in [5].

1.2.2 Data compression algorithms and information theory in RL

A 2007 paper [4] proposes a model-based RL algorithm that constructs the Lempel-Ziv (LZ) parsing tree by simulating the LZ algorithm on (state, action) pairs. The algorithm maintains estimates of probabilities and value functions at each node of the parsing tree. On the information theory side, Lu and Van Roy (2019) [6], as well as Section 4 in a related paper

[7], analyze the role of information gain towards per-period performance, characterizing the exploration-exploitation tradeoff.

1.2.3 Undetermined and Growing Horizons

The planning horizon H is central in the analysis of a learning algorithm, as H can be adjusted depending on the total running time T. For example, [9] and [3] decompose the regret into separate terms that increase and decrease in H, then solving for the value of H that gives the best asymptotic regret in T. A related technique to handle the situation where k is variable and unknown is *growing the horizon*, which is covered in [3].

Chapter 2

LZ-augmented Optimistic Q-learning

In this chapter, we present the *LZ-augmented Optimistic Q-learning* algorithm, the central idea of which involves combining the Lempel-Ziv (LZ) parsing tree structure in [4] (where the $|\mathbb{AX}|$ possible (action, response) pairs comprise the alphabet), with the optimistic Q-learning value functions and corresponding updates presented in [9] (where the value functions are computed at each context generated by the LZ parsing tree). The algorithm is presented in *Algorithm 1* at the following page, and the two main ideas are expounded in this chapter.

2.1 Lempel-Ziv Parsing Tree

Algorithm 1 builds a LZ parsing tree very similar to that in [4]. The contexts in the tree are subsequences of (action, response) pairs in the history. A count $n_t(c, a)$ is maintained at each (context, action) pair and incremented after each phrase of the LZ parsing. The algorithm starts with an empty context, which is updated by including the (action, response) pairs in the current phrase (line 7). An action is then picked (line 8) based on the value functions at the context, to be described in the next section, after which a response is observed (line 9).

The algorithm then checks whether the context has been seen before (line 10). If it is in a context that has not been seen before, then similar to the LZ algorithm, the phrase is ended (line 19). In *Algorithm 1*, however, the value functions along the phrase are updated in addition. The timing and duration of the phrases are tracked by the variables u and start_u, which refer to the index of the current phrase and the start time of the uth phrase, respectively. The process is continued until the Tth time step.

A context that contains a history of length at least k is called a *valid context*; otherwise, it is called an *invalid context*. It is only at *valid contexts* where the distribution of the response, given a specific action at this context, is known, as otherwise, the history of length k for which the distribution depends on will contain history from a previous phrase.

Algorithm 1 LZ-augmented Optimistic Q-learning Algorithm

Parameters: horizon $H \geq 2$, confidence level $\delta \in (0,1)$

Define: $\forall \tau : \alpha_\tau = \dfrac{H+1}{H+\tau}$, $\beta_\tau = 4\mathrm{sp}(v^*)\sqrt{\dfrac{H}{\tau}\ln\dfrac{2T}{\delta}}$

$\gamma \leftarrow 1 - H^{-1}$

$\hat{V}_1(\cdot) \leftarrow H$, $Q_1(\cdot,\cdot) \leftarrow H$, $\hat{Q}_1(\cdot,\cdot) \leftarrow H$ {initialize value functions at each context}

$n_1(\cdot,\cdot) \leftarrow 0$ {initialize context counts}

$u_1 \leftarrow 1$ {index of the current phrase}

$\mathrm{start}_u \leftarrow 1$ {start time of the uth phrase}

for $t = 1, \ldots, T$ **do**

  $c_t \leftarrow H_{\mathrm{start}_u}^{t-1}$ {context at time t; may be empty when $t = \mathrm{start}_u$ }

  Pick action a_t uniformly at random over the set $\mathrm{argmax}_{a\in\mathcal{A}}\hat{Q}_t(c_t,a)$.

  Observe response x_t

  if $\sum_{a\in\mathcal{A}} n_t(c_t,a) = 0$ **then** {we are in a context not seen before}

    for t' with $\mathrm{start}_u \leq t' \leq t$ in decreasing order **do**

      {traverse backward through the current phrase to perform the Optimistic Q-learning update steps at each context}

      $n_{t+1}(c_{t'},a_{t'}) \leftarrow n_t(c_{t'},a_{t'}) + 1$

      $\tau \leftarrow n_{t+1}(c_{t'},a_{t'})$

      $c'_{t'} = H_{\mathrm{start}_u}^{t'}$ {context at which to get value estimate for Q-learning}

      $Q_{t+1}(c_{t'},a_{t'}) \leftarrow (1-\alpha_\tau)Q_t(c_{t'},a_{t'}) + \alpha_\tau\left[r(a_t,x_t) + \gamma\hat{V}_t(c'_{t'}) + b_\tau\right]$

      $\hat{Q}_{t+1}(c_{t'},a_{t'}) \leftarrow \min\{\hat{Q}_t(c_{t'},a_{t'}), Q_{t+1}(c_{t'},a_{t'})\}$

      $\hat{V}_{t+1}(c_{t'}) \leftarrow \max_{a\in\mathcal{A}}\hat{Q}_{t+1}(c_{t'},a)$.

      $u_{t+1} \leftarrow u_t + 1$, $\mathrm{start}_u \leftarrow t+1$ {start the next phrase}

    end for

  else

    $u_{t+1} \leftarrow u_t$

  end if

  {All function values not updated at time t remain the same between indices t and $t+1$.}

end for

2.2 Q-learning value functions

Algorithm 1 uses similar setup as in [9] for the Q-learning process. It takes in a horizon H and confidence level δ as parameters, which parameterize the *update weight* α_τ (used in determining the weights of the previous Q value and the value estimate) and the *bonus term* β_τ (added to the value estimate). The Q-learning process involves the same three functions as in [9] — $\hat{V}(c)$, $Q(c, a)$, and $\hat{Q}(c, a)$ — which are updated in a similar manner. The function $\hat{Q}(c, a)$ is used (line 8) to decide on the action. The key difference is that the algorithm has a value function at each *context* rather than each *state*; this distinction is important as the agent is not supposed to know the maximum history length k, which determines the corresponding set of states.

The other difference is in computing the value estimate (line 16) for the Q-learning update, as the value functions are updated at the end of each phrase, and the update is done by traversing backward through the current context in order to perform the Q-learning update steps, as is done in [4]. When the algorithm is in a context not seen before, instead of using the value estimate at the next context, which will be the empty context, it uses the un-updated value estimate (which has the value H) of the context formed by appending the last (action, response) pair of the phrase (line 15). This is to stay consistent with the tree structure induced by the LZ algorithm, as well as to avoid a meaningless value from the empty context, which is an *invalid* context.

In *Algorithm 1*, the functions are indexed by the time t, mostly as a notational convenience for the subsequent analysis, so that we can keep track of the value of the functions at different times. The functions are also defined in theory for all possible contexts c, even those never visited, and for all times t. In a practical implementation, however, only the latest value needs to be stored. Similarly, only values of the functions at contexts that have been visited need to be stored, as otherwise, the value will simply be H.

Chapter 3

Theoretical Analysis of the Algorithm

In this chapter, we present steps towards proving that *Algorithm 1* obtains sublinear regret. We follow a similar approach as Section 4 in [9], where the analysis is done for a particular setting of H and δ. The major differences stem from the distinction between the state s_t, context c_t, and response x_t at the time t.

3.1 Preliminaries

Unlike the previous chapter, the analysis in this chapter may explicitly consider the states s_t, as there is an underlying MDP that was only unknown to the algorithm.

3.1.1 Definitions

We define $\mathcal{C}$ to be the set of valid contexts, and $\mathcal{T}$ to be the set of times $1 \leq t \leq T$ where $c_t \in \mathcal{C}$. A consequence is that for a valid context $c \in \mathcal{C}$, there is a unique corresponding state given by the last k (state, action) pairs in c; we denote this state as $s(c)$.

We consider the MDP over $(\mathbb{A} \times \mathbb{X})^k$ as discussed in Section 1.1. Under the weakly communicating assumption, the discounted (with horizon H and corresponding decay $\gamma = 1 - H^{-1}$) and undiscounted value functions $V^*(s)$ and $v^*(s)$, respectively.

Similar to Section A.2 in [9], define $n_t = n_{t+1}(c_t, a_t)$, the number of visits to the *context*-action pair (c_t, a_t) in the first t times. As will be seen later, a key quantity of interest is the measure of uncertainty $U = \sum_{t \in \mathcal{T}} n_t^{-1/2}$.

We also adapt the definitions $\alpha_\tau^0 = \prod_{j=1}^{\tau}(1 - \alpha_j)$ and $\alpha_\tau^i = \alpha_i \prod_{j=i+1}^{\tau}(1 - \alpha_j)$ from [9], where by convention $\alpha_0^0 = 1$. α_τ^i represents the weight of the ith value estimate (line 16 in *Algorithm 1*) towards the value of $Q(c_t, a_t)$ after the τth visit to this context-action pair.

3.1.2 Lemmas from [9]

We make use of the following lemmas from Appendix A of [9], copied below for reference.

Lemma 1 *The following properties hold for α_τ^i:*

1. $\frac{1}{\sqrt{\tau}} \leq \sum_{i=1}^{\tau} \frac{\alpha_\tau^i}{\sqrt{i}} \leq \frac{2}{\sqrt{\tau}}$ *for every* $\tau \geq 1$.
2. $\sum_{i=1}^{\tau} (\alpha_\tau^i)^2 \leq \frac{2H}{\tau}$ *for every* $\tau \geq 1$.
3. $\sum_{i=1}^{\tau} \alpha_\tau^i = 1$ *for every* $\tau \geq 1$ *and* $\sum_{\tau=i}^{\infty} \alpha_\tau^i = 1 + \frac{1}{H}$ *for every* $i \geq 1$.

Lemma 2 (Azuma's inequality) *Let $X_1, X_2, \cdots$ be a martingale difference sequence with $|X_i| \leq c_i$ for all i. Define $\bar{c}_T^2 = \sum_{i=1}^{T} c_i^2$. Then, for $\delta \in (0,1)$, $\mathbb{P}\left(\sum_{i=1}^{T} X_i \geq \sqrt{2\bar{c}_T^2 \log \frac{1}{\delta}}\right) \leq \delta$.*

We can also adapt the following lemmas in Section 4.1 of [9], which is valid as these lemmas are about the value functions of MDPs, and we indeed have a MDP where the states are s_t and actions are a_t.

Lemma 3 *Let $sp(v^*)$ and $sp(V^*)$ be the spans of the undiscounted and discounted MDPs, respectively. Then,*

1. $|J^* - (1-\gamma)V^*(s)| \leq (1-\gamma) sp(v^*)$, $\forall s \in (\mathbb{A} \times \mathbb{X})^k$
2. $sp(V^*) \leq 2 sp(v^*)$

Lemma 4 *With probability at least $1-\delta$,*

$$\sum_{t=1}^{T} (Q^*(s_t, a_t) - \gamma V^*(s_t) - r(s_t, a_t) \leq 2 sp(v^*) \sqrt{2T \log \frac{1}{\delta}} + 2 sp(v^*).$$

3.2 Reduction to bounding $U = \sum_{t \in \mathcal{T}} n_t^{-1/2}$

We focus on the dependence on the parameters H and T dropping dependence on all other parameters and quantities for simplicity of analysis. In our initial attempts towards showing sublinear regret, we simply need to find an appropriate H for each T.

In this section, take note that the $\hat{Q}$, Q, and $\hat{V}$ functions are on *context-action* pairs, as they are functions in *Algorithm 1*, while the Q^* and V^* functions are on *state-action* pairs, as they are value functions in the associated MDP.

3.2.1 Regret Decomposition

We use a similar regret decomposition in Section 4.1 of [9]. However, considering that for $t \notin \mathcal{T}$, the actions a_t are likely to stay suboptimal as there is no correct transition pmf to learn at the context c_t, we consider these times separately for the $\sum_{t=1}^{T}(V^*(s_t) - Q^*(s_t, a_t)$ term. Hence, we get:

$$R_T = \sum_{t=1}^{T}(J^* - r(s_t, a_t)) \tag{3.1}$$

$$= \sum_{t=1}^{T}(J^* - (1-\gamma)V^*(s_t)) + \sum_{t=1}^{T}(Q^*(s_t, a_t) - \gamma V^*(s_t) - r(s_t, a_t)) \tag{3.2}$$

$$+ \sum_{t\in\mathcal{T}}(V^*(s_t) - Q^*(s_t, a_t)) + \sum_{t\notin\mathcal{T}, 1\le t\le T}(V^*(s_t) - Q^*(s_t, a_t)) \tag{3.3}$$

The first term in (3.2) is $O\left(\frac{T}{H}\right)$ by Lemma 3, and the second term in (3.2) is $O(\sqrt{T})$ by Lemma 4.

We now bound the second term of (3.3). For each t, $0 \le V^*(s_t) - Q^*(s_t, a_t) \le V^*(s_t) \le H$, so the second term of (3.3) is bounded above by H times the size of the set $\{t : t \in \mathcal{T}, 1 \le t \le T\}$, the latter of which is bounded by K times c_T, the number of LZ phrases involved in the parsing. By a similar reasoning as Lemma 13.5.3 in [2] (extending from a binary sequence to an alphabet with size $|\mathbb{A}||\mathbb{X}|$), $c_T = O\left(\frac{T}{\log T}\right)$. Combining,

$$\sum_{t\notin\mathcal{T}, 1\le t\le T}(V^*(s_t) - Q^*(s_t, a_t)) = O\left(H\frac{T}{\log T}\right).$$

It remains to bound $\sum_{t\in\mathcal{T}}(V^*(s_t) - Q^*(s_t, a_t))$. In the remainder of this section, we show that, where $U = \sum_{t\in\mathcal{T}} n_t^{-1/2}$,

$$\sum_{t\in\mathcal{T}}(V^*(s_t) - Q^*(s_t, a_t)) = O\left(H\frac{T}{\log T}\right) + O(\sqrt{H\log T})\cdot U.$$

Collecting the bounds above gives $R_T = O\left(\frac{T}{H} + H\frac{T}{\log T}\right) + O(\sqrt{H\log T})\cdot U$.

3.2.2 Bounding $\sum_{t\in\mathcal{T}}(V^*(s_t) - Q^*(s_t, a_t))$

We show a bound of $\sum_{t\in\mathcal{T}}(V^*(s_t) - Q^*(s_t, a_t)) = O\left(H\frac{T}{\log T}\right) + O(\sqrt{H\log T})\cdot U$. The proof will follow Lemma 3 in [9], though it will deviate considerably as the telescoping argument is more complicated. We show this bound by restating Lemma 12 in [9], and then adapting the telescoping argument given the unbounded number of possible contexts.

Restating Lemma 12 in [9]

Given our formulation with contexts and states, we provide a restated version of Lemma 12 in [9]. Here, c'_{t_i} is the context at time t_i, with the (action, response) pair at time t_i, as specified in line 15 of *Algorithm 1*.

Lemma 5 *With probability at least $1-\delta$, for any time $1 \le t \le T$ and context-action pair (c, a), and where $t_1, \ldots, t_\tau \le t$ are the times where (c, a) is visited,*

$$0 \le \hat{Q}_{t+1}(c,a) - Q^*(s(c),a) \le H\alpha_\tau^0 + \gamma\sum_{i=1}^{\tau}\alpha_\tau^i\left[\hat{V}_{t_i}(c'_{t_i}) - V^*(s(c'_{t_i}))\right] + 12sp(v^*)\sqrt{\frac{H}{\tau}\log\frac{2T}{\delta}}.$$

The proof is very similar as that of Lemmas 12 in [9], though some lines have to be restated given the distinction between states and contexts.

As in [9], we compute $Q_t(c, a)$ and $Q^*(s(c), a)$, then subtract the latter from the former.

$Q_{t+1}(c, a)$ is computed by expanding the τ applications of line 16 in *Algorithm 1*:

$$Q_{t+1}(c,a) = H\alpha_\tau^0 + \sum_{i=1}^{\tau}\alpha_\tau^i\left[r(s(c),a) + \gamma\hat{V}_{t_i}(c'_{t_i})\right] + \sum_{i=1}^{\tau}\alpha_\tau^i b_i.$$

Meanwhile, $Q^*(s(c), a)$ can be written in a similar form using the Bellman Equation of the MDP, $Q^*(s(c), a) = r(s(c), a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s(c),a)}V^*(s')$:

$$Q^*(s(c),a) = \alpha_\tau^0 Q^*(s(c),a) + \sum_{i=1}^{\tau}\alpha_\tau^i\left[r(s(c),a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s(c),a)}V^*(s')\right].$$

Taking the difference,

$$Q_{t+1}(c,a) - Q^*(s(c),a) = (H - Q^*(s(c),a))\alpha_\tau^0 + \gamma\sum_{i=1}^{\tau}\alpha_\tau^i\left[\hat{V}_{t_i}(c'_{t_i}) - \mathbb{E}_{s'\sim p(\cdot|s(c),a)}V^*(s')\right] + \sum_{i=1}^{\tau}\alpha_\tau^i b_i.$$

Finally, we can split

$$\left[\hat{V}_{t_i}(c'_{t_i}) - \mathbb{E}_{s'\sim p(\cdot|s(c),a)}V^*(s')\right] = \left[\hat{V}_{t_i}(c'_{t_i}) - V^*(s(c'_{t_i}))\right] + \left[V^*(s(c'_{t_i})) - \mathbb{E}_{s'\sim p(\cdot|s(c),a)}V^*(s')\right],$$

after which the rest of the proof can proceed similarly as in [9]. The union bound can proceed similarly as the second term in the above split only takes T different forms.

Completing the Telescoping Argument

For $t \in \mathcal{T}$, define $\Delta_V(t) = \hat{V}_t(c_t) - V_t^*(s(c_t))$, and analogously, $\Delta'_V(t) = \hat{V}_t(c'_t) - V_t^*(s(c'_t))$. $\Delta_V(t)$ represents the difference between the time-t estimate of the value function c_t, with the true value function of the associated state in the MDP. $\Delta'_V(t)$ represents the same, though for the context c'_t. Note that this is not necessarily the next context, but rather, c_t appended with the (action, response) pair at time t. Now, we compute:

$$\sum_{t\in\mathcal{T}}(V^*(s_t) - Q^*(s_t, a_t)) \tag{3.4}$$
$$= \sum_{t\in\mathcal{T}}(V^*(s(c_t)) - Q^*(s(c_t), a_t)) \tag{3.5}$$
$$= \sum_{t\in\mathcal{T}}(\hat{V}_t(c_t) - Q^*(s(c_t), a_t)) - \sum_{t\in\mathcal{T}}(\hat{V}_t(c_t) - V^*(s(c_t)) \tag{3.6}$$
$$= \sum_{t\in\mathcal{T}}(\hat{V}_t(c_t) - Q^*(s(c_t), a_t)) - \sum_{t\in\mathcal{T}}\Delta_v(t), \tag{3.7}$$

where (3.5) is from c_t being a valid context, (3.6) is from adding then subtracting $\hat{V}_t(c_t)$ inside the summation, and (3.7) is from the definition of $\Delta_v(t)$.

Then, we express the first summation of (3.7) as

$$\sum_{t\in\mathcal{T}}(\hat{V}_t(c_t), Q^*(s(c_t), a_t)) \tag{3.8}$$
$$= \sum_{t\in\mathcal{T}}(\hat{Q}_t(c_t, a_t) - Q^*(s(c_t)), a_t) \tag{3.9}$$
$$= \sum_{t\in\mathcal{T}}(\hat{Q}_{t+1}(c_t, a_t) - Q^*(s(c_t), a_t)) + \sum_{t\in\mathcal{T}}(\hat{Q}_t(c_t, a_t) - \hat{Q}_{t+1}(c_t, a_t)), \tag{3.10}$$

where (3.9) is from a_t being the chosen action at time t, and (3.10) is from adding and subtracting $\hat{Q}_{t+1}(c_t, a_t)$ inside the summation, then splitting the summation.

The second term of (3.10) can be analyzed by split splitting the summation by (context, action) pair. Within each context-action pair, the terms are a telescoping sequence of differences of value function estimates bounded in $[0, H]$, so the sum of all such differences is

bounded by H. The number of (context, action) pairs is at most $|\mathbb{A}|c_T$, and $c_T = O\left(\frac{T}{\log T}\right)$, so

$$\sum_{t\in\mathcal{T}}(\hat{Q}_t(c_t,a_t) - \hat{Q}_{t+1}(c_t,a_t)) = O\left(H\frac{T}{\log T}\right) \tag{3.11}$$

Meanwhile, the bound in 5 can be used to analyze the first term of (3.10). First, we restate 5 using the definition of $\Delta'_V(t)$ as

$$0 \le \hat{Q}_{t+1}(c,a) - Q^*(s(c),a) \le H\alpha_\tau^0 + \gamma\sum_{i=1}^{\tau}\alpha_\tau^i\Delta'_V(t) + 12\mathrm{sp}(v^*)\sqrt{\frac{H}{\tau}\log\frac{2T}{\delta}}. \tag{3.12}$$

Let $t_i(c,a)$ be the ith time when the (context, action) pair (c,a) is visited. Substituting the right-hand side of (3.12) into the first term of (3.10), and noting that $\tau \ge 1$ always for any t,

$$\sum_{t\in\mathcal{T}}(\hat{Q}_{t+1}(c_t,a_t) - Q^*(s(c_t),a_t)) \tag{3.13}$$

$$\le \sum_{t\in\mathcal{T}}\left(12\mathrm{sp}(v^*)\sqrt{\frac{H}{n_t}\log\frac{2T}{\delta}} \;+\; \gamma\sum_{i=1}^{n_t}\alpha_{n_t}^i\Delta'_V(t_i(c_t,a_t))\right) \tag{3.14}$$

$$= 12\mathrm{sp}(v^*)\sqrt{H\log\frac{2T}{\delta}}\sum_{t\in\mathcal{T}}n_t^{-1/2} + \gamma\sum_{t\in\mathcal{T}}\sum_{i=1}^{n_t}\alpha_{n_t}^i\Delta'_V(t_i(c_t,a_t)) \tag{3.15}$$

$$= O(\sqrt{H\log T})\cdot U + \gamma\sum_{t\in\mathcal{T}}\sum_{i=1}^{n_t}\alpha_{n_t}^i\Delta'_V(t_i(c_t,a_t)). \tag{3.16}$$

The second term in (3.15) term be bounded in a similar way as the third term of Equation (17) in [9], though only valid contexts correspond to those visited at times $t \in \mathcal{T}$. It is also adaptable given 5. This procedure yields a bound of

$$\gamma\sum_{t\in\mathcal{T}}\sum_{i=1}^{n_t}\alpha_{n_t}^i\Delta'_V(t_i(c_t,a_t)) \le \left(1+\frac{1}{H}\right)\gamma\sum_{t\in\mathcal{T}}\Delta'_V(t) \le \sum_{t\in\mathcal{T}}\Delta'_V(t). \tag{3.17}$$

Finally, combining (3.7), (3.10), (3.11), (3.16), and (3.17) gives

$$\sum_{t\in\mathcal{T}}(V^*(s_t) - Q^*(s_t,a_t)) \tag{3.18}$$

$$= \sum_{t\in\mathcal{T}}(\hat{V}_t(c_t) - Q^*(s(c_t),a_t)) - \sum_{t\in\mathcal{T}}\Delta_v(t) \tag{3.19}$$

$$= \sum_{t\in\mathcal{T}} (\hat{Q}_{t+1}(c_t, a_t) - Q^*(s(c_t), a_t)) + \sum_{t\in\mathcal{T}} (\hat{Q}_t(c_t, a_t) - \hat{Q}_{t+1}(c_t, a_t)) - \sum_{t\in\mathcal{T}} \Delta_v(t) \tag{3.20}$$

$$\leq O(\sqrt{H \log T}) \cdot U + \gamma \sum_{t\in\mathcal{T}} \sum_{i=1}^{n_t} \alpha_{n_t}^i \Delta'_V(t_i(c_t, a_t)) + O\left(H \frac{T}{\log T}\right) - \sum_{t\in\mathcal{T}} \Delta_v(t) \tag{3.21}$$

$$\leq O(\sqrt{H \log T}) \cdot U + \sum_{t\in\mathcal{T}} \Delta'_V(t) + O\left(H \frac{T}{\log T}\right) - \sum_{t\in\mathcal{T}} \Delta_v(t) \tag{3.22}$$

$$= O\left(\sqrt{H \log T}\right) \cdot U + O\left(H \frac{T}{\log T}\right) + \left(\sum_{t\in\mathcal{T}} \Delta'_V(t) - \sum_{t\in\mathcal{T}} \Delta_V(t)\right) \tag{3.23}$$

Finally, we upper-bound the second quantity in (3.24) by comparing the quantities in the summation. First, note that $0 \leq \Delta_V(t), \Delta'_V(t) \leq H$ for all t, by definition. When t does not mark the end of a phrase, $t+1 \in \mathcal{T}$ and $c'_t = c_{t+1}$, so $\Delta'_V(t) = \Delta_V(t+1)$, so for all such t, the term $\Delta'_V(t)$ is cancelled out by $\Delta_V(t+1)$ in the second summation. Therefore, the second quantity in (3.23) is upper-bounded by H times the number of phrases, u_T, and thus it is $O\left(H \frac{T}{\log T}\right)$. Combining in (3.23) finally gives

$$\sum_{t\in\mathcal{T}} (V^*(s_t) - Q^*(s_t, a_t)) = O\left(\sqrt{H \log T}\right) \cdot U + O\left(H \frac{T}{\log T}\right). \tag{3.24}$$

3.3 Progress towards on bounding U

In this subsection, we discuss progress towards bounding U in order to achieve sublinear regret in T. The previous subsection uses the perspective of guaranteeing a particular bound with a given probability, specified by the parameter. Here, however, we will currently be content at bounding $\mathbb{E}[U]$, though possible approaches towards a similar probabilistic guarantee will be presented towards the end. (Note, however, that given that U is nonnegative, a bound of $\mathbb{E}[U]$ trivially yields a probabilistic bound via Markov's inequality, though this bound has an inferior scaling in the confidence parameter δ.)

The approach in [9] involves setting H for the overall regret R_T to be sub-linear. Considering the bound $R_T = O\left(\frac{T}{H} + H \frac{T}{\log T}\right) + O(\sqrt{H \log T}) \cdot U$, we need U to be sufficiently sublinear in T. Given the definition of U, it is likely that our bounds will depend on T alone and not H. It can be seen that when $U = o\left(\frac{T}{\sqrt{\log T}}\right)$, then there will exist H such that R_T is sublinear, by setting H sufficiently small yet still growing in T.

So far, we have yet to bound U or $\mathbb{E}[U]$, though below we present a simplified version of the setting, which proves to be more tractable to combinatorial techniques. Then, we discuss

how the simplified version may be related to the original setting.

3.3.1 Branching Process Approach

The simplified version involves reinterpreting on the sequence of (context, action) pairs encountered as a branching process over the context tree, ignoring the causality of the learning algorithm's process towards future actions. This formulation enables combinatorial analysis, as will be seen below, though with the difficulty of having some distance to the setting with the learning algorithm.

Formulation of Alternative Model

In the alternative model, replace the learning algorithm with deterministic functions $g_1(u, c)$ and $g_2(u, c, a)$, where g_1 prescribes the action of the agent when at the uth context and at context c, and where g_2 prescribes the PMF of the environment's response at the uth context, at context c, and when the player has done action a. These two functions replace the learning algorithm of the agent and the history-dependent PMF of the environment, respectively.

Given $g_1(u, c)$ and $g_2(u, c, a)$, a stochastic process is defined over the context tree, which we subsequently analyze combinatorially and using recursion on the number of visits to each (context, action) pair. We furthermore assume that for any u, c, a, the PMF given by $g_2(u, c, a)$ places a probability of at least $\epsilon < |\mathbb{X}|^{-1}$ at each of $|\mathbb{X}|$ possible responses. This restriction is easily translated to the original setting, by imposing this on the transition kernel $P(X_t|H_{t-k}^{t-1}, A_t)$. In the analysis below, we consider dependence in ϵ in addition to H and T.

Define u' as the minimum number of phrases needed to guarantee a total time of at least T. Then, run the algorithm for u' time steps, taking a total time of T', which is stochastic. The below bound will be computed in terms of u', though as $u' = O\left(\frac{T}{\log T}\right)$ from the previous discussion, the bound can easily be converted into a bound in T.

Bound Setup

We consider the setting of the alternative model with fixed functions g_1 and g_2. We bound the expectation of $U' = \sum_{i=1}^{T'} n_t^{-1/2}$. We write this as

$$U' = \sum_{i=1}^{T'} n_t^{-1/2} \tag{3.25}$$

$$= \sum_{i=1}^{T'} \sum_{c,a} n_{t+1}^{-1/2}(c,a) \mathbb{1}[c_t = c, a_t = a] \tag{3.26}$$

$$= \sum_{c,a} \sum_{j=1}^{n_{T'+1}(c,a)} j^{-1/2} \tag{3.27}$$

$$= \sum_{i=1}^{\infty} i^{-1/2} \sum_{j=1}^{T'} \mathbb{1}[n_j = i] \tag{3.28}$$

$$= \sum_{i=1}^{\infty} i^{-1/2} \sum_{c,a} \mathbb{1}[n_{T'+1}(c,a) \geq i] \tag{3.29}$$

$$= \sum_{i=1}^{u'|\mathbb{A}|} i^{-1/2} \sum_{c,a} \mathbb{1}[n_{T'+1}(c,a) \geq i], \tag{3.30}$$

where the last equality is due to the number of visited contexts being upper-bounded by u', the number of phrases times the number of actions. Hence,

$$\mathbb{E}[U'] = \sum_{i=1}^{u'|\mathbb{A}|} i^{-1/2} \mathbb{E}\left[f(l)\right], \tag{3.31}$$

where we define $f(l) = \sum_{c,a} \mathbb{1}[n_{T'+1}(c,a) \geq l]$, the number of (context, action) pairs that were visited at least i times.

We claim the bound $\mathbb{E}[f(l)] \leq \max\left(u'|\mathbb{A}|, \tilde{f}(u', \epsilon, l)\right)$, where

$$\tilde{f}(n, \epsilon, l) = \begin{cases} 0 & n < l \\ \frac{n - \epsilon(l-1)}{1 + \epsilon(l-1)} & n \geq l \end{cases}.$$

Showing this bound implies that $\mathbb{E}[f(l)] \leq \max\left(u'|\mathbb{A}|, \frac{u'}{\epsilon l}\right)$, as $\frac{u' - \epsilon(l-1)}{1 + \epsilon(l-1)} \leq \frac{u'}{\epsilon l}$; substituting into (3.31) gives a bound of $\mathbb{E}[U'] = O\left(u'\epsilon^{-1}\right)$

Proof of Bound

We now prove the bound $\mathbb{E}[f(l)] \leq \tilde{f}(u', \epsilon, l)$, where

$$\tilde{f}(n, \epsilon, l) = \begin{cases} 0 & n < l \\ \frac{n - \epsilon(l-1)}{1 + \epsilon(l-1)} & n \geq l \end{cases}.$$

It is easy to see that for any n_1 and n_2, $\tilde{f}(n_1+n_2,\epsilon,l) > \tilde{f}(n_1,\epsilon,l)+\tilde{f}(n_2,\epsilon,l)$, and by induction, this extends to an arbitrary number of n_i. We call this property *superadditivity*, for short.

Define $\mathcal{T}_{(c,a)}$ to be the set of times t for which (c_t,a_t) has (c,a) as prefix, when we consider the simple concatenation of the stream of actions and responses. Then, define $V^l_{(c,a)} = \sum_{t\in\mathcal{T}_{(c,a)}} \mathbb{1}[n_t \geq l]$, and define $\mathcal{S}_{(c,a)}$ as the set of indices of the phrases that contain the (context, action) pair (c,a). Extend the notation analogously for (context, action, response) triples (c,a,x).

We now show the following claim: For any (context, action) pair (c,a) and set $\mathcal{S}\subset[u']$ such that $\mathbb{P}(\mathcal{S}_{(c,a)}=\mathcal{S})>0$,

$$\mathbb{E}[V^l_{(c,a)}|\mathcal{S}_{(c,a)}=\mathcal{S}] \leq \tilde{f}(|\mathcal{S}|,\epsilon,l).$$

We show this claim by strong induction on $|\mathcal{S}|$, and simultaneously for all (c,a). The base cases $1\leq|\mathcal{S}|\leq l-1$ are immediate as $V^l_{(c,a)}=0$, due to there being fewer than l visits to (c,a), so none of its descendants will have l or greater visits.

For the inductive step, consider the stochastic process that splits $\mathcal{S}_{(c,a)}=\mathcal{S}$ into the sets $\mathcal{S}_{(c,a,x)}=\mathcal{S}^x$, for $x\in\mathbb{X}$, based on the PMF given by g_2 After this, for each $t\in\mathcal{S}_{(c,a,x)}$, the next action is given deterministically from g_1, so the split from a set $\mathcal{S}_{(c,a,x)}=\mathcal{S}^x$ to the sets $\mathcal{S}_{(c,a,x,a')}=\mathcal{S}^{(x,a')}$, $a\in\mathbb{A}$ is deterministic.

Therefore, we can write:

$$\mathbb{E}[V^l_{(c,a)}|\mathcal{S}_{(c,a)}=\mathcal{S}] = 1+\mathbb{E}\left[\sum_x\sum_{a'}\mathbb{E}\left[V^l_{(c,a,x,a')}|\mathcal{S}_{(c,a,x,a')}=\mathcal{S}^{(x,a')}\right]\right] \tag{3.32}$$

$$\leq 1+\mathbb{E}\left[\sum_x\sum_{a'}\tilde{f}(|\mathcal{S}^{(x,a')}|,\epsilon,l)\right] \tag{3.33}$$

$$\leq 1+\mathbb{E}\left[\sum_x\tilde{f}(|S^x|,\epsilon,l)\right] \tag{3.34}$$

where (3.32) is from the definition of $V^l_{(c,a)}$ and counting the context (c,a) separately from the descendants; (3.33) is from the induction hypothesis, and (3.34) is from super-additivity.

Next, we index the actions as $a_1,\ldots,a_{\mathbb{A}}$ and the responses as $x_1,\ldots,x_{\mathbb{X}}$. Define the indicator random variables $y_i=\mathbb{1}[|S^{x_i}|\leq l-1]$, and make the observation that $\sum_{i=1}^{|\mathbb{X}|} = |\mathcal{S}|-1$.

Continuing,

$$\mathbb{E}[V^l_{(c,a)}|\mathcal{S}_{(c,a)}] \leq 1 + \mathbb{E}\left[\sum_x \tilde{f}(|S^x|, \epsilon, l)\right] \tag{3.35}$$

$$= 1 + \mathbb{E}\left[\sum_{i=1}^{|\mathbb{X}|}(1-y_i)\left(\frac{|\mathcal{S}^{x_i}| - \epsilon(l-1)}{1+\epsilon(l-1)}\right)\right] \tag{3.36}$$

$$= 1 + \mathbb{E}\left[\sum_{i=1}^{|\mathbb{X}|}\left(\frac{|\mathcal{S}^{x_i}| - \epsilon(l-1)}{1+\epsilon(l-1)}\right) - \sum_{i=1}^{|\mathbb{X}|} y_i\left(\frac{|\mathcal{S}^{x_i}| - \epsilon(l-1)}{1+\epsilon(l-1)}\right)\right] \tag{3.37}$$

$$\leq 1 + \frac{(|\mathcal{S}|-1) - 2\epsilon(l-1)}{1+\epsilon(l-1)} - \frac{\mathbb{E}\left[\sum_{i=1}^{|\mathbb{X}|} y_i(|\mathcal{S}^{x_i}| - \epsilon(l-1))\right]}{1+\epsilon(l-1)} \tag{3.38}$$

$$= \frac{|\mathcal{S}| - \epsilon(l-1)}{1+\epsilon(l-1)} - \frac{\sum_{i=1}^{|\mathbb{X}|}\mathbb{E}[y_i(|\mathcal{S}^{x_i}| - \epsilon(l-1))]}{1+\epsilon(l-1)}, \tag{3.39}$$

where (3.38) uses the fact that $|\mathbb{X}| \geq 2$.

We then use the following Lemma, the proof of which is deferred to the appendix.

Lemma 6 *Consider a sequence of Bernoulli random variables $Y_1, Y_2, \ldots$, where $Y_i \sim Ber(\tilde{p}_i)$, where $0 < \tilde{p}_i < 1$ for all $i \geq 1$. Define $X_m = \sum_{i=1}^{m} Y_i$. Then,*

1. *For fixed $m \geq 2$, $\dfrac{\mathbb{P}(X_m = j)}{\mathbb{P}(X_m \leq j)}$ is decreasing in j, for $1 \leq j \leq m$.*
2. *Suppose $\tilde{p}_i \geq \epsilon$ for all i. Then, for $\gamma \in \mathbb{Z}^+$, $\mathbb{E}[X_m | X_m \leq \gamma] \geq k\epsilon$.*

From this lemma, for every $1 \leq i \leq |\mathbb{X}|$,

$$\mathbb{E}[|\mathcal{S}^{x_i}| \,|\, |\mathcal{S}^{x_i}| \leq l-1] \geq \epsilon(l-1) \tag{3.40}$$

$$\iff \mathbb{E}[|\mathcal{S}^{x_i}| - \epsilon(l-1)|y_i] \geq 0 \tag{3.41}$$

$$\iff \mathbb{E}[y_i(|\mathcal{S}^{x_i}| - \epsilon(l-1))] \geq 0, \tag{3.42}$$

so the second term in (3.39) is negative, and so $\mathbb{E}[V^l_{(c,a)}|\mathcal{S}_{(c,a)}] \leq \dfrac{|\mathcal{S}| - \epsilon(l-1)}{1+\epsilon(l-1)}$, as desired.

Finally, applying the induction hypothesis on the empty context, and using super-additivity over the $|\mathbb{A}|$ actions yields the bound $\mathbb{E}[f(l)] \leq \tilde{f}(u', \epsilon, l)$.

Weakness of Approach

The main weakness of this alternative model is that the distribution of histories through time T is not captured by any randomization over different possible (g_1, g_2). This is because the

learning algorithm involves the actual outcome at a particular time affecting actions made by the agent in the ancestor nodes in the context tree at a future time; this also applies to the PMF's of the environment's response at invalid contexts.

The conditioning argument for the above bound was valid because the (action, response) pairs in a node's subtree does not affect the events that are being conditioned on, which is at which phrase-times the context will be reached. Applying the inductive argument to the original setting would thus not be valid due to possible influence on future contexts. As will be discussed below, however,

3.3.2 Possible Directions

There are several possible directions towards bounding U that can be explored in future work.

- Considering an adversarial specification of the functions g_1 and g_2 that instead depend on possibly the entire history: this resolves the issue of future actions being uncorrelated to the outcomes. This approach might be more tractable when only considering valid contexts, as the environment's response PMF will be fixed at any given (context, action) pair, so only the player's actions will need to be prescribed. This formulation can potentially be handled by similar recursive approaches, though with a more complicated setup.

- Redoing the previous sub-section with an expected value bound instead: this can potentially remove some of the multipliers that only arose due to application of Azuma's inequality. However, there might be difficulties from loosening the condition that the value estimates are always an overestimate, as this is invariant is used in some key steps in the proof.

Appendix A

Proof of Lemma 6

A.1 Proof of Part 1

For integer $l \in [0, m]$, define S_l as the lth elementary symmetric polynomial in $(q_1, \ldots, q_m) = \left(\frac{\tilde{p}_1}{1-\tilde{p}_1}, \ldots, \frac{\tilde{p}_m}{1-\tilde{p}_m}\right)$. Define $p_l = \mathbb{P}(X_m = l)$, and based on the previous definition,

$$p_l = \mathbb{P}(X_m = l) = S_l \prod_{i=1}^{m} (1 - \tilde{p}_i).$$

By Newton's Inequalities, the elementary symmetric means of $(q_1, \ldots, q_m)$, which are $S_l \binom{n}{l}^{-1}$, form a log-concave sequence in $l = 0, \ldots, m$. As $\binom{n}{l}^{-1}$ is a log-convex sequence in $l = 0, \ldots, m$, S_l and thus p_l are also log-concave sequences in $l = 0, \ldots, m$.

We now show that for all $0 \leq j \leq m-1$,

$$\frac{\mathbb{P}(X_m = j)}{\mathbb{P}(X_m \leq j)} > \frac{\mathbb{P}(X_m = j+1)}{\mathbb{P}(X_m \leq j+1)}.$$

Indeed, taking the difference, and using the fact that $p_j p_i \geq p_{j+1} p_{i-1}$ for all $1 \leq i \leq l$ which follows from log-concavity of p_l,

$$\frac{\mathbb{P}(X_m = j)}{\mathbb{P}(X_m \leq j)} - \frac{\mathbb{P}(X_m = j+1)}{\mathbb{P}(X_m \leq j+1)} \tag{A.1}$$

$$= \frac{p_j}{\sum_{i=1}^{j} p_i} - \frac{p_{j+1}}{\sum_{i=1}^{j+1} p_i} \tag{A.2}$$

$$= \frac{(p_j \sum_{i=1}^{j+1} p_i) - (p_{j+1} \sum_{i=1}^{j} p_i)}{\sum_{i=1}^{j} p_i \ \sum_{i=1}^{j+1} p_i} = \frac{p_j p_1 + \sum_{i=1}^{j} (p_j p_i - p_{j+1} p_{i-1})}{(\sum_{i=1}^{j} p_i)(\sum_{i=1}^{j+1} p_i)} > 0. \tag{A.3}$$

A.2 Proof of Part 2

The statement is clearly true for $m \leq \gamma$. For $m > \gamma$, we show it by induction on m, by showing that the function $g(m) = \mathbb{E}[X_m | X_m \leq \gamma]$ is non-decreasing in m.

Consider the function

$$h(m, l) = \frac{\mathbb{P}(l \leq X_m \leq \gamma)}{\mathbb{P}(X_m \leq \gamma)}.$$

Then, $g(m) = \sum_{i=1}^{\gamma} h(m, i)$, so it suffices to show that $h(m, l)$ is non-decreasing in m, over positive integer $m > \gamma$, for each fixed $l \leq \gamma$.

This is equivalent to showing that

$$h(m+1, l) - h(m, l) = \frac{\mathbb{P}(l \leq X_{m+1} \leq \gamma)}{\mathbb{P}(X_{m+1} \leq \gamma)} - \frac{\mathbb{P}(l \leq X_m \leq \gamma)}{\mathbb{P}(X_m \leq \gamma)} \geq 0.$$

We use the recurrence relations

$$\mathbb{P}(l \leq X_{m+1} \leq \gamma) = \mathbb{P}(l \leq X_m \leq \gamma) + \tilde{p}_{m+1}\mathbb{P}(X_m = l-1) - \tilde{p}_{m+1}\mathbb{P}(X_k = \gamma), \text{ and} \quad \text{(A.4)}$$

$$\mathbb{P}(X_{m+1} \leq \gamma) = \mathbb{P}(X_m \leq \gamma) - \tilde{p}_{m+1}\mathbb{P}(X_m = \gamma), \quad \text{(A.5)}$$

which can be computed by casework on the ranges of values of X_m and Y_{m+1}. Finally, we can evaluate, and substituting (A.4) and (A.5) to (A.7):

$$\frac{\mathbb{P}(l \leq X_{m+1} \leq \gamma)}{\mathbb{P}(X_{m+1} \leq \gamma)} - \frac{\mathbb{P}(l \leq X_m \leq \gamma)}{\mathbb{P}(X_m \leq \gamma)} \geq 0 \quad \text{(A.6)}$$

$$\iff \mathbb{P}(l \leq X_{m+1} \leq \gamma)\mathbb{P}(X_m \leq \gamma) - \mathbb{P}(X_{m+1} \leq \gamma)\mathbb{P}(l \leq X_m \leq \gamma) \geq 0 \quad \text{(A.7)}$$

$$\iff \mathbb{P}(X_m = l-1)\mathbb{P}(X_m \leq \gamma) - \mathbb{P}(X_m = \gamma)\mathbb{P}(X_m \leq \gamma) + \mathbb{P}(X_m = \gamma)\mathbb{P}(l \leq X_m \leq \gamma) \geq 0 \quad \text{(A.8)}$$

$$\iff \mathbb{P}(X_m = l-1)\mathbb{P}(X_m \leq \gamma) \geq \mathbb{P}(X_m = \gamma)\mathbb{P}(X_m \leq l-1) \quad \text{(A.9)}$$

$$\iff \frac{\mathbb{P}(X_m = l-1)}{\mathbb{P}(X_m \leq l-1)} \geq \frac{\mathbb{P}(X_m = \gamma)}{\mathbb{P}(X_m \leq \gamma)}, \quad \text{(A.10)}$$

which is true by Part 1.

(To go from (A.7) to (A.8), substitute (A.4) and (A.5) into the relevant expressions, expand, cancel out the like terms $\mathbb{P}(l \leq X_k \leq \gamma)\mathbb{P}(X_k \leq \gamma)$, and then cancel out the common factor of $\tilde{p}_{m+1}$.)

References

[1] Dimitri P Bertsekas. *Reinforcement learning and optimal control / by Dimitri P. Bertsekas*. eng. Athena Scientific optimization and computation series. Belmont, Massachusetts: Athena Scientific, 2019. ISBN: 9781886529397.

[2] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006. ISBN: 0471241954.

[3] Shi Dong, Benjamin Van Roy, and Zhengyuan Zhou. "Simple Agent, Complex Environment: Efficient Reinforcement Learning with Agent State". In: *CoRR* abs/2102.05261 (2021). arXiv: 2102.05261. URL: https://arxiv.org/abs/2102.05261.

[4] Vivek F. Farias et al. "Universal Reinforcement Learning". In: *CoRR* abs/0707.3087 (2007). arXiv: 0707.3087. URL: http://arxiv.org/abs/0707.3087.

[5] Chi Jin et al. "Is Q-learning Provably Efficient?" In: *CoRR* abs/1807.03765 (2018). arXiv: 1807.03765. URL: http://arxiv.org/abs/1807.03765.

[6] Xiuyuan Lu and Benjamin Van Roy. *Information-Theoretic Confidence Bounds for Reinforcement Learning*. 2019. DOI: 10.48550/ARXIV.1911.09724. URL: https://arxiv.org/abs/1911.09724.

[7] Xiuyuan Lu et al. "Reinforcement Learning, Bit by Bit". In: *CoRR* abs/2103.04047 (2021). arXiv: 2103.04047. URL: https://arxiv.org/abs/2103.04047.

[8] Yi Ouyang et al. "Learning Unknown Markov Decision Processes: A Thompson Sampling Approach". In: *CoRR* abs/1709.04570 (2017). arXiv: 1709.04570. URL: http://arxiv.org/abs/1709.04570.

[9] Chen-Yu Wei et al. "Model-free Reinforcement Learning in Infinite-horizon Average-reward Markov Decision Processes". In: *CoRR* abs/1910.07072 (2019). arXiv: 1910.07072. URL: http://arxiv.org/abs/1910.07072.