Strategic Exploration in Reinforcement Learning - New Algorithms and Learning Guarantees

Christoph Dann

September 2019
CMU-ML-19-116

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Emma Brunskill (Stanford University), *Chair*
Barnabás Póczos (Carnegie Mellon University)
Benjamin Recht (University of California Berkeley)
Benjamin Van Roy (Stanford University)
Rémi Munos (Deepmind)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was funded by the National Science Foundation award IIS1350984, the Office of Naval Research award N000141612241, the Air Force Research Laboratory award FA87501720152, and gifts from Microsoft Corporation, Google and Verizon.

Keywords: Reinforcement Learning, Sequential Decision Making Under Uncertainty, Learning Theory, Exploration, Markov Decision Process, Accountability

Abstract

Reinforcement learning (RL) focuses on an essential aspect of intelligent behavior – how an agent can learn to make good decisions given experience and rewards in a stochastic world. Yet popular RL algorithms that have enabled exciting successes in domains with good simulators (Go, Atari, etc) still often fail to learn in other domains because they rely on simple heuristics for exploration. This provides additional empirical justification for essential questions around RL, specifically around algorithms that learn in a provably efficient manner through strategic exploration in any considered domain. This thesis provides new algorithms and theory that enable good performance with respect to existing theoretical frameworks for evaluating RL algorithms (specifically, probably approximately correct) and introduces new stronger evaluation criteria, that may be particularly of interest as RL is applied to more real world problems.

For the first line of work on probably approximately correct (PAC) RL algorithms, we introduce a series of algorithms for episodic tabular domains with substantially better PAC sample complexity bounds that culminate in a new algorithm with close to minimax optimal PAC and regret bounds. Look up tables are required by most sample efficient and computationally tractable algorithms, but cannot represent many practical domains. We therefore also present a new RL algorithm that can learn a good policy in environments with high dimensional observations and hidden deterministic states; unlike predecessors, this algorithm provably explores not only in a statistically but also computationally efficient manner assuming access to function classes with efficient optimization oracles.

To make progress it is critical to have the right measures of success. While empirical demonstrations are quite clear, we find that for theoretical properties, two of the most commonly used learning frameworks, PAC guarantees and regret guarantees, each allow undesirable algorithm behavior (e.g. ignoring new observations that could improve the policy). We present a new stronger learning framework called *Uniform-PAC* that unifies the existing frameworks and prevents undesirable algorithm properties.

One caveat of all existing learning frameworks is that for any particular episode, we do not know how well the algorithm will perform. To address this, we introduce the *IPOC* framework that requires algorithms to provide a certificate before each episode bounding how suboptimal the current policy can be. Such certifications may be of substantial interest in high stakes scenarios when an organization may wish to track or even pause an online RL system should the potential expected performance bound drop below a required expected outcome.

Acknowledgments

I would like to express my gratitude towards my advisor, Emma Brunskill. Throughout the past five years she has taught me to be a better researcher and I am particularly thankful for her repeated encouragement to think more broadly about the impact of my work. I always felt she has put my interests and development first, providing me with all resources and guidance a young researcher could hope for. Many thanks especially for taking the risk and letting an incoming PhD student with applied background try out theoretical research. Without this opportunity I might have never discovered my passion of machine learning theory. Many thanks also to Barnabás Póczos, Benjamin Recht, Benjamin Van Roy and Rémi Munos for serving on my thesis committee and providing me with fruitful advice, shaping the direction of this dissertation.

I would also like to extend many thanks to all collaborators who have been directly involved in the work included in this dissertation: to Tor Lattimore who can explain complex mathematical arguments in such amazingly simple and still precise terms and who has endless patience to use this skill helping a young PhD student entering RL theory. This was essential for the results in Chapters 3 and 4; to the team at Microsoft Research New York, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford and Rob Schapire, for an inspiring summer internship and great collaborations. The work in Chapter 6 was truly a group effort and would not exist without their enthusiasm and time commitment into this internship project; to Wei Wei for being a supportive local mentor during my internship at Google that lead to Chapter 5; and to Lihong Li who did not shy away from making many additional trips during an already busy summer to be able to work with me in front of a white board and share his impressively broad knowledge of RL.

Many thanks also to Sebastian Nowozin and Katja Hofmann for their guidance and the freedom to shape a new research project together, making my second summer internship in Cambridge, UK, such a wonderful experience. Thank you to Sebastian in particular for also giving me a first glimpse of how fulfilling industrial machine learning research can be during my first internship at Microsoft Research. I am also indebted to Peter Gehler who not only fueled my enthusiasm for machine learning in his lectures at TU Darmstadt but also introduced me to hands-on research as an undergraduate student. Many thanks to Jan Peters for the warm welcome to his research group as Master's student at TU Darmstadt, his invaluable career advice and his truly inspiring research enthusiasm. Without all this, I would have never ended up on a path to a PhD at CMU.

Many thanks to all current and former members of the AI for Human Impact Lab for the countless research discussions and fun interactions at CMU and Stanford. It was a pleasure to share the joys and hardships of being a PhD student. Many thanks especially to Phil Thomas for introducing me to the RL family during my first NeurIPS conference, for the fun times in front of the white board at CMU and for sharing his logic puzzles which kept my brain occupied for many hours.

I feel grateful for the five years I have spent in the Machine Learning Department (MLD) at CMU, a truly unique research community. Many thanks to all members, especially to Diane Stidle who deserves large credit for making MLD such a supportive and welcoming environment for PhD students; to Abulhair Saparov, Adarsh Prasad, Alnur Ali, Anthony Platanios, Arun Sai Suggala, Avinava Dubey, Chun-Liang Li, Chenghui Zhou, Dan Schwartz, Ezra Winston, Jacob Tyo, Mariya Toneva, Maruan Al-Shedivat, Mrinmaya Sachan, Otilia Stretcu, Peter Stojanov, Robin Schmucker, William Herlands, Willie Neisswanger, Xun Zheng

and to everyone else that I was fortunate enough to cross path with at CMU.

My endless gratitude goes towards my family; to my parents for their endless support throughout my entire life. I would not be who I am today without their hard work and countless sacrifices. They always encouraged my passion for sciences and mathematics but also made sure to pass on their priceless practical skill set; to my big brother for being a role model in so many ways and for teaching me how to ride a bicycle early on. Finally, I thank my best friend and partner Mariya, who I met during my first days as a PhD student and who has become a constant source of happiness and love in my life — to many more bikes, bakes and bacillus bulgaricus together.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Reinforcement learning (RL) is a branch of machine learning that studies sequential decision making under uncertainty and provides a general framework for many practical problems in artificial intelligence. In the basic RL setup, an agent interacts with an uncertain environment in order to perform a task by taking a sequence of actions. Reinforcement learning provides algorithmic tools to optimize the agent's strategy to perform the given task. There have been impressive recent empirical successes propelled by deep learning which demonstrate that reinforcement learning can solve challenging tasks. These include playing a range of Atari video games (Mnih et al., 2015), achieving human-level performance in Go (Silver et al., 2017) or beating professional players in Starcraft II (Vinyals et al., 2019). But the potential applications of RL go beyond games. It is a natural framework for optimizing recommender systems, e.g., for news (Li, Chu, et al., 2010) or videos (Chen et al., 2019), but also for optimizing adaptive treatments in health-care (Lei et al., 2012), dialog systems (Singh et al., 2002) or instruction schedules in intelligent tutoring systems (Atkinson, 1972; Mandel et al., 2014).

When designing reinforcement learning algorithms, we typically face three main challenges: *generalization*, *partial feedback*, and *long-term implications* (see Figure 1.1):

Generalization: Generalization means that we want our agent to act well in situations it never encountered before by generalizing from experience in similar situations. This is achieved by building on function approximation techniques from supervised machine learning, where generalizing from samples in the training data set to new samples in the text data set is a key challenge.

Partial feedback: Unlike in supervised learning where each training sample comes with the desired output, the agent does not get to know which action it should have taken after each interaction. The only feedback is a scalar reward which indicates how good the *chosen action* was, but no feedback on other actions. This partial feedback necessitates to explore different actions in order to learn about them.

Long-term implications: The final challenge is that consequences of a single action are not entirely captured by the immediate reward feedback but there can be long-term implications. For examples, deciding to undergo surgery causes risk and discomfort for the patient (low immediate reward) but will significantly improve the long-term health (high reward at later times). For this reason, a good reinforcement learning system must optimize for long-term total reward which is challenging as feedback can be severely delayed. This is sometimes also referred to as *credit assignment* challenge in RL because the agent must figure out which prior actions causes current feedback.

Many of the impressive empirical applications of reinforcement learning successfully address the generalization and partial feedback challenges. This includes large-scale services such as the Decision

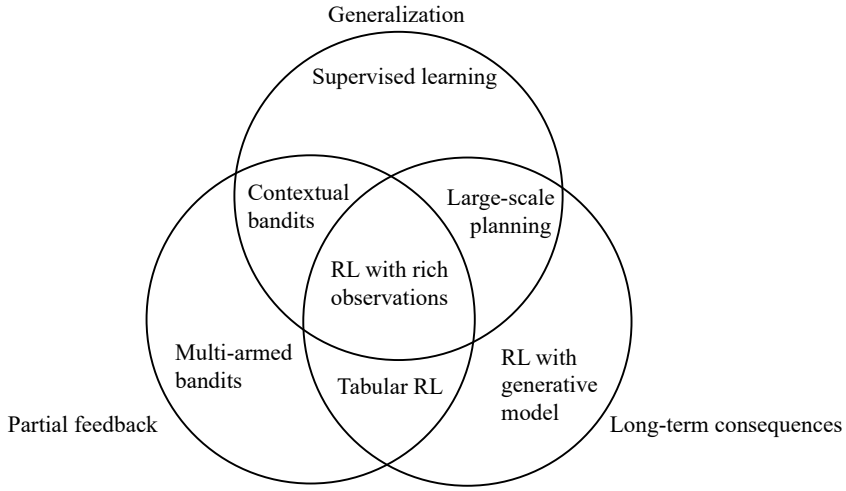

Figure 1.1: Main challenges in reinforcement learning define a landscape of different problems settings.

Service (Agarwal, Bird, et al., 2016) which explicitly ignores long-term consequences by modeling the decision problem as a so-called contextual bandit. Other successful empirical RL works, especially those leveraging deep learning techniques like Mnih et al. (2015), do consider delayed consequences but become very data-inefficient as the delay becomes longer. This is because they rely on heuristics to explore the effect of actions. For example, Mnih et al. (2015) uses ϵ-greedy exploration where the agent flips a biased coin before each action and with probability ϵ picks an action uniformly at random. This strategy works very well empirically (Bietti, Agarwal, and Langford, 2018) in bandit problems without long-term consequences but is also known require exponentially many interactions to learn long-term effects of actions (see for example Section 4 of Osband, Van Roy, Russo, et al. (2017) for an illustration). As a result, these approaches fail to learn good policies even with excessive amounts of data when long-term effects of actions are essential, e.g., in the Atari game Montezuma's revenge. There are several empirical efforts to more efficient exploration (e.g. through reward bonuses, see below) but problems with long-term effects remain a challenge.

If we want to unleash the full potential of reinforcement learning, we need algorithms that can deal in a principled efficient manner with long-term effects (as well as partial feedback and generalization). Many important applications have long-term consequences, including the following examples:

- **Architecture Search** (Zoph and Le, 2016): Searching for a good architecture of a neural network machine learning model can be formulated as a reinforcement learning task. Here, the agent incrementally defines an architecture (each action decides on an additional component) and only receives a non-zero reward (the performance of the architecture on a certain task) when the architecture is fully defined.
- **Optimize User Engagement in Recommender Systems** (Chen et al., 2019): Deciding what products to recommend to a user on a shopping platform or what songs/videos on a streaming platform is a reinforcement learning problem with long-term effects. While recommending music that is very similar to what the user usually listens is most engaging n the short-run, helping the user to develop a taste for an entirely different genre can be more beneficial on the long-run.
- **Sustainable Yield Optimization in Agriculture** (Binas, Luginbuehl, and Bengio, 2019): When modeling the control of plants in agriculture to maximize the long-term yield, long-term effects can

be crucial. For example, deciding to plant seedlings very densely can seem very promising for an extended amount of time until the plants eventually become too large and do not grow to their full potential due to limited space and nutrients.

- **Treatment Optimization in Personalized Health Care** (Liu et al., 2018): When optimizing the long-term well-being of individuals long-term effects are abundant. Take starting to exercise or undergoing coronary bypass surgery as an example. Both lead to short-term risk and discomfort but eventually improve long-term well-being.

These applications are also examples where *sample-efficiency* is key for successful reinforcement learning methods. That means the algorithm should learn to perform the task with as few interactions as possible. Compared to the cost of simulating a game of Go, chess or even Star Craft II, the costs for obtaining samples in these applications is significantly higher, ranging from excessive computational costs for training and evaluating a neural network architecture, over physical resources required for farming, to harming patients in health care applications. Especially in high-stakes applications that involve humans, our goal should be algorithms that are as sample-efficient as possible in a provable or at least reliable manner.

However, sample-efficiency alone is not sufficient for successful RL methods in high-stakes applications. Unlike in supervised learning, the performance of an RL algorithm is typically not monotonically increasing with more data due to the trial-and-error nature of RL that necessitates exploration. Even sharp drops in performance during learning are common, e.g., when the agent starts to explore a new part of the state space. We argue that RL methods should be *accountable*, which means that they can predict when such performance drops can happen. This allows domain experts to intervene or fallback systems to be triggered if necessary. For example, in the treatment optimization application listed above, a human doctor could be consulted if the RL algorithm is cannot ensure the performance of its treatment strategy for a particular patient.

1.2 Near-Optimal Sample-Efficiency and Accountability in Episodic RL

The goal of this dissertation is to make reinforcement learning more sample-efficient and accountable so that it becomes more suitable for real-world high-stakes applications such as the ones listed above. We focus on episodic reinforcement learning problems where the interaction between environment and agent happens in episodes of fixed number of decisions. This is a natural fit for many applications involving humans, e.g., one episode corresponds to treating one patient, to a web session of a customer or to teaching a certain topic to a student in automated tutoring systems.

Sample-efficient learning in tasks with long-term consequences as those above requires algorithms to explore the effects of actions is a strategic way that takes the long-term effects after multiple time steps into account. We refer to such exploration as *strategic exploration* (Dann, Jiang, et al., 2018; Dann, Li, et al., 2019; Du et al., 2019; Sun et al., 2018) but it can be found under many names in the literature: deep exploration (Osband, Blundell, et al., 2016), systematic exploration (Jiang, Krishnamurthy, et al., 2017; Houthooft et al., 2016), temporally-extended exploration (Osband, Blundell, et al., 2016) or sample-efficient exploration (Dann and Brunskill, 2015). There are two main principles to reliable strategic exploration: optimism in the face of uncertainty (OFU) principle (Auer, Cesa-Bianchi, and Fischer, 2002) and Thompson (posterior) sampling (Russo, Van Roy, et al., 2018). While there are other principles such as information directed sampling (Russo and Van Roy, 2014), optimism in the face of uncertainty and Thompson sampling have gained the most trust and acceptance in RL research through two main pillars:

- Theoretical guarantees about their sample-efficiency, mostly in simplified settings like multi-armed bandits and tabular reinforcement learning where no generalization is necessary.
  For OFU algorithms this includes Strehl and Littman (2005), Auer and Ortner (2005), Auer, Jaksch,

and Ortner (2009), Azar, Osband, and Munos (2017), and Jiang, Krishnamurthy, et al. (2017) and for Thompson-sampling algorithms see Osband, Russo, and Van Roy (2013), Osband and Van Roy (2017), and Russo (2019).

- Approximate implementations of these principles with empirical evaluations demonstrating that they can learn good policies in more complicated problem settings where the agent has to learn from rich observations such as images or text.
  In the case of Thompson sampling this includes exploration via randomized value functions (Osband, Blundell, et al., 2016; Osband, Van Roy, Russo, et al., 2017). The OFU principle is typically implemented through *reward bonuses* (Tang et al., 2017; Bellemare et al., 2016; Ostrovski et al., 2017; Burda et al., 2018)

The work in this dissertation provides a step towards more sample-efficient and accountable reinforcement learning by advancing the understanding of theoretical performance guarantees and strategic exploration. We contribute to both pillars above, with a focus on the first. To gain insight, we first leave generalization aside, and analyze in this dissertation *tabular reinforcement learning* in episodic Markov decision processes (MDPs) where the agent does not need to generalize across observations and can simply store all necessary information in look-up tables (hence the name tabular).

A key challenge in designing strategic exploration approaches is that the algorithm has to reason about its uncertainty about the environment and how this translates into a policy (strategy for taking actions). This policy should both achieve good performance (exploitation) and help to reduce the algorithm's uncertainty (exploration). While uncertainty in Thompson sampling algorithms is simply the Bayesian belief, it is often non-trivial how to computationally represent and update this Bayesian belief. OFU algorithms have the issue that one has to explicitly derive a representation of the uncertainty and the more accurate this representation, the more sample-efficient the resulting algorithm is (Osband and Van Roy, 2017). The first optimistic algorithms represented this uncertainty as a binary sets of known state-action pairs and unknown state-action pairs (Brafman and Tennenholtz, 2002) while later approaches (Strehl and Littman, 2008; Auer and Ortner, 2005) used confidence sets around the environment parameters: the average instantaneous reward as well as the transition probabilities to each successor state in all states and actions. Such representations yield algorithms that do not scale well with the problem size, namely number of states and horizon (number of decisions per episode). More precisely, for the episodic setting we consider, there were only a few methods that achieve a sample-complexity (roughly speaking the number of episodes until a good policy is found) that is polynomial in the problem size and their order are high.

This dissertation provides several insights on how to better represent uncertainty in OFU algorithms in episodic tabular MDPs which yields more sample-efficient strategic exploration. First, we leverage empirical Bernstein concentration bounds to more tightly quantify the uncertainty of the transition probabilities. This yields a theoretical algorithm that scales optimally with horizon. Second, to improve scaling with the number of states, we directly bound the uncertainty of average optimal next state value instead of the transition probabilities. It turns out this is more accurate (tighter confidence set) and still sufficient for determining the agent's next policy. While Azar, Osband, and Munos (2017) developed an algorithm that combines both of these insights, its sample-complexity does not quite scale optimally with the number of states and horizon. We address this by incorporating a final insight: we not only quantify the uncertainty over the optimal value function (expected rewards to go) but also the uncertainty of the algorithm's currently achieved value function. This results in an optimistic algorithm with sample-complexity that is minimax-optimal (best achievable for the worst-case scenario) in the dominating terms. Interestingly, this final insight was developed by aiming to improve accountability and not necessarily sample-efficiency.

While the benefit of empirical comparisons on benchmark problems are often clear, one should not underestimate the importance of theoretical analyses that yield guarantees. There is growing awareness

that evaluations on a restricted set of benchmarks can sometimes leads to irreproducible results (Henderson et al., 2017) and unlike empirical evaluations which are by nature limited to a finite (small) set of tasks, theoretical learning guarantees can inform us about the performance of an algorithm on any task in the considered problem class. As such they can be a helpful tool for comparing algorithms and can provide insights that help us design improved algorithms. However, for these guarantees to be meaningful, we need to ensure that they measure all aspects of the algorithm's behavior that we care about. As we will highlight in this dissertation, both PAC and regret bounds (Kearns and Singh, 2002; Auer, Jaksch, and Ortner, 2009), the two most common types of sample-efficiency guarantees have blind-spots. That means there are algorithms that enjoy good guarantees in these frameworks but still exhibit undesirable behavior such as not converging to the optimal policy. We address these issues by introducing new, stronger types of guarantees and use them for our new algorithms. This includes a learning framework that not only ensures that the algorithm deploys better and better policies but that it also can certify online how good the current policy is. This is the first step towards such online certificates with guaranteed accuracy in reinforcement learning.

These insights into quantifying uncertainty in optimistic algorithms and theoretical guarantees have demonstrated how to do reinforcement learning with near-optimal sample-efficiency and accountability in tabular episodic tasks. However, our understanding of how to do provably sample-efficient reinforcement learning is much more limited in problems where a succinct state representation is not given and the algorithm has to work with rich observation such as images or texts. We show in this thesis that there are significant computational challenges with strategic exploration in rich observation settings. On the negative side, we show that OLIVE (Jiang, Krishnamurthy, et al., 2017), the only known algorithm with provably polynomial sample-efficiency in a large class of rich observation problems, is computationally intractable and we provide a first step to alleviate this issue by proposing an new algorithm that is both statistically and computationally efficient in a more restricted class of rich observation problems.

1.3 Thesis Statement

The central thesis of this dissertation is that simple reinforcement learning algorithms with strategic exploration through carefully designed reward bonuses are provably near-optimally sample-efficient and accountable in finite episodic Markov decision processes. We demonstrate how to design such reward bonuses by leveraging the decision process structure and introduce new types of performance guarantees for sample-efficiency and accountability. These guarantees not only subsume existing learning frameworks like PAC and regret but also guarantee that algorithms are accountable by accurately certifying their current performance online. For environments beyond those with finite state spaces, we provide new insights into the computational difficulty of strategic exploration.

1.4 Organization

Chapter 2 covers some background on Markov decision processes and episodic reinforcement learning and introduces the common types of sample-complexity guarantees.

Chapter 3 introduces our analysis of sample-complexity of episodic MDPs with finite state and action spaces (Dann and Brunskill, 2015). We prove a lower bound on the (worst-case) sample-complexity achievable by any algorithm and provide an algorithm with a sample-complexity bound that matches the lower bound up to a factor in the size of the state space and log-factors.

In Chapter 4 we discuss the existing performance guarantees, regret and PAC (probably approximately correct) bounds and show that they are inherently incomparable and each allow undesirable algorithm

behavior. We further propose a new type of learning guarantee, Uniform-PAC bounds, which are stronger than both regret and PAC bounds (Dann, Lattimore, and Brunskill, 2017). In fact, a Uniform-PAC bound implies both a strong PAC and regret bound and prohibits undesirable algorithm behavior allowed by either existing guarantee. To demonstrate these benefits, we provide a simple algorithm with a strong Uniform-PAC bound that empirically outperforms other algorithms with known sample-complexity bounds.

While Uniform-PAC bounds address many issues of previous guarantees, it also does not provide a guarantee on the performance of a policy in a single episode. We aim to address this short-coming by proposing stronger *IPOC* (Individual policy certificates) guarantees in Chapter 5. In this learning framework, the algorithm is required to output an upper bound on the suboptimality of the policy it is about to execute before each episode. This not only allows the user to intervene in high-stakes applications but also extract good policies at any time from the algorithm. We demonstrate this with an algorithm called `ORLC` for episodic finite MDPs that leverages lower-confidence bounds to provide certificates in addition to the upper confidence bounds that guide exploration. It turns out that this is a key insight that also improves sample-efficiency and allows us to prove IPOC, regret and PAC bounds that are smaller that any prior work and minimax-optimal up to lower-order terms.

In Chapter 6 we move from tabular problems to working on how to implement strategic exploration in problems with rich observations. We present `VALOR`, an algorithm for reinforcement learning in episodic reactive POMDPs with rich observations and deterministic hidden state dynamics (Dann, Jiang, et al., 2018). Unlike predecessors this algorithm not only enjoys a polynomial sample-complexity bound but is also provably computationally tractable in an oracle-model. Here, we assume that linear programs for the chosen value function class and cost-sensitive classification problems for the policy class can be solved efficiently. We hope that this work is a step toward provably sample- and computationally-efficient reinforcement learning with function approximation and provides insights into extending this work to more general problem settings.

1.5 Summary of Contributions

- **Chapter 3:** We quantify the difficulty of reinforcement learning in tabular episodic MDPs by proving a lower bound for problem-independent PAC guarantees in this problem class. This bound is tight up logarithmic terms.
- **Chapter 3:** We propose an optimism-based algorithm for tabular episodic MDPs and prove a PAC bound with optimal dependency on the episode length (up to log-terms).
- **Chapter 4:** We quantify to what extend regret and PAC bounds can be converted to each other in episodic problems.
- **Chapter 4:** As regret and PAC are not easily convertible and each allow undesirable algorithm behavior, we introduce a new framework for learning guarantees called *Uniform-PAC*. We prove that it is stronger than existing prevalent frameworks, including Mistake-PAC and regret.
- **Chapter 4:** We propose a simple optimism-based algorithm for tabular episodic MDPs and prove that is has a Uniform-PAC bound.
- **Chapter 5:** We propose that algorithms output *policy certificates* during learning to make them more accountable and introduce a new framework for providing learning guarantees that also ensure accuracy of learning guarantees. This framework called *IPOC* is stronger than existing frameworks, including Uniform-PAC and supervised-style PAC bounds.
- **Chapter 5:** We propose a simple optimistic algorithm that not only maintains upper confidence

bounds but also lower confidence bounds. We then prove that this technique allows it to achieve minimax-optimal IPOC, PAC and regret bounds up to lower-order terms.

- **Chapter 6:** We prove that the only known statistically efficient algorithm for general problems with rich observation spaces (problems with so-called low Bellman rank) is computationally intractable, even when applied to tabular MDPs.
- **Chapter 6:** We propose a new algorithm called Valor for reinforcement learning for sub-class of problems with rich observations (whose underlying unobserved states transition deterministically). We show that this algorithm is not only statistically efficient but can also be implemented efficiently with standard optimization oracles.

1.6 Excluded Work

This dissertation contains my main line of work on provably sample-efficient and accountable reinforcement learning. I have contributed to other works during my PhD studies which are to varying extent beyond this scope. These works are:

- Reinforcement learning with strategic exploration for risk-averse return objectives like conditional value-at-risk. This work is the basis for future work on providing policy certificates for criteria beyond expected return:
  Ramtin Keramati, Alex Tamkin, Christoph Dann, and Emma Brunskill. "Being Optimistic to Be Conservative: Quickly Learning a CVaR Policy". In: *in preparation* (2019)
- Provably sample-efficient reinforcement learning in stopping problems:
  Karan Goel, Christoph Dann, and Emma Brunskill. "Sample efficient policy search for optimal stopping domains". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press. 2017, pp. 1711–1717
- Analytical tool for determining how much history (information from observations before the current) an observed behavior policy uses which can help to learn policies faster by restricting the policy class to this amount of history:
  Christoph Dann, Katja Hofmann, and Sebastian Nowozin. "Memory Lens: How Much Memory Does an Agent Use?" In: *arXiv preprint arXiv:1611.06928* (2016)
- New class of natural gradient algorithms that leverage more information than classic natural gradient techniques which rely on the Fisher information matrix:
  Philip Thomas, Bruno Castro Silva, Christoph Dann, and Emma Brunskill. "Energetic natural gradient descent". In: *International Conference on Machine Learning*. 2016, pp. 2887–2895
- Generalization of the natural gradient algorithm idea beyond algorithms that follow the gradient direction, such as algorithm with momentum:
  Philip Thomas, Christoph Dann, and Emma Brunskill. "Decoupling Gradient-Like Learning Rules from Representations". In: *International Conference on Machine Learning*. 2018, pp. 4924–4932
- Improving gernalization performance of Gaussian process regression by learning a kernel from human predictions:
  Andrew G Wilson, Christoph Dann, Chris Lucas, and Eric P Xing. "The human kernel". In: *Advances in neural information processing systems*. 2015, pp. 2854–2862
- Fast scalable inference and training for Guassian processes:
  Andrew Gordon Wilson, Christoph Dann, and Hannes Nickisch. "Thoughts on massively scalable Gaussian processes". In: *arXiv preprint arXiv:1511.01870* (2015)

- Algorithm for automated matching of detected defects in pipeline inspections (application work): Markus R Dann and Christoph Dann. “Automated matching of pipeline corrosion features from in-line inspection data”. In: *Reliability Engineering & System Safety* 162 (2017), pp. 40–50

Chapter 2

Background on Episodic Reinforcement Learning and Notation

We first introduce the formal problem setting considered in Chapters 3 – 5 and then define additional helpful notation. The setting in Chapter 6 is substantially more general asking for different notation, which we introduce there when needed.

2.1 Episodic Finite-Horizon Markov Decision Processes

A *Markov decision process* or short *MDP* is a random process which formalizes sequential decision making of an agent interacting with an environment (Puterman, 1994). This dissertation focuses on the episodic finite horizon version of this process, called *episodic fixed-horizon MDP* defined by a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, P, p_0, H)$.

- The *state space* $\mathcal{S}$ is a set of states that the process can generate. We assume this set is finite of size S and without loss of generality set $\mathcal{S} = \{1, 2, \dots, S\}$.
- The *action space* $\mathcal{A}$ is a set of actions the agent is allowed to take after observing the state. Similar to the state space, we assume this is a finite set of size A, that is, $\mathcal{A} = \{1, 2, \dots, A\}$.
- The *reward distribution* $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}_{[0,1]}$ is a mapping from states and actions to distributions with support $[0, 1]$. For their expected value, we use $r(s, a) = \mathbb{E}_{r\sim\mathcal{R}(s,a)}[r]$.
- The *transition distribution* or next state distribution $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}_{\mathcal{S}}$ maps state and action pairs to distributions over states. We use $P(s'|s, a)$ to mean the probability that s' is the successor state of s when action a was taken.
- The *initial state distribution* $p_0 \in \mathcal{P}_{\mathcal{S}}$ is a distribution over states which the process is initialized to at the beginning of each episode.
- The *horizon* $H \in \mathbb{N}$ is the number of time steps in each episode ("length" of the episode).

The process specified by the MDP describes an episodic interaction of an agent with an environment. We typically use k to index episodes and at the beginning of each episode k, the initial state $s_{k,1} \sim p_0$ is drawn from p_0. For each *time step* within the episode (typically indexed by t or H), the agent observes state $s_{k,t} \in \mathcal{S}$ and takes an action $a_{k,t} \in \mathcal{A}$. It then receives a reward $r_{k,t} \sim \mathcal{R}(s_{k,t}, a_{k,t})$ and observes the state $s_{k,t+1} \sim P(s_{k,t}, a_{k,t})$ of the next time step $t + 1$. This interaction loop continues for a total of H time steps before the next episode $k + 1$ begins. For notational convenience, we assume that the agent still observes the $H + 1$th state in the episode $s_{k,H+1}$ but then there is no following interaction. Formally, each *episode*

k is a sequence $(s_{k,1}, a_{k,1}, r_{k,1}, s_{k,2}, a_{k,2}, r_{k,2}, \dots s_{k,H}, a_{k,H}, r_{k,H}, s_{k,H+1})$.

Expected total reward and return. The objective for the agent is to play actions that maximize the *total expected reward* in the episode $\mathbb{E}\left[\sum_{t=1}^{H} r_{k,t} | a_{k,t} \sim \pi\right]$ which depends on how actions are taken. The *policy* of the algorithm formalizes the strategy with which the agent picks the actions given all observations so far. It is well known that there is always a *deterministic Markov policy* that maximizes the total expected reward (Puterman, 1994). These policies simply pick the action as a function of the current state and time step $\Pi : \mathcal{S} \times [H] \to \mathcal{A}$ where $[H] = \{1, 2, \dots H\}$. It is useful to interpret reinforcement learning algorithms as learners of these deterministic policies. Formally, we can write the total expected reward as a function of the algorithm's (current) policy $\pi \in \Pi$

$$\rho(\pi) = \mathbb{E}\left[\sum_{t=1}^{H} r_{k,t} | a_{k,t} = \pi(s_{k,t}, t)\right],$$

where k is arbitrary as the total expected reward is identical across episodes as long as the algorithm follows the same policy. Below, we omit the episode index k and only use the time step when the episode index is arbitrary for readability. The quantity $\sum_{t=1}^{H} r_{k,t}$ is also known as *return* and $\rho(\pi)$ as the *expected return* of policy π.

Value functions and optimal policies. The *value function* V_h^π and Q-function Q_h^π of a policy at time step $h \in [H]$ are defined as

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H} r_t \,\middle|\, s_h = s,\ a_{h:H} \sim \pi\right]$$
$$Q_h^\pi(s,a) = \mathbb{E}\left[\sum_{t=h}^{H} r_t \,\middle|\, s_h = s,\ a_h = a,\ a_{h+1:H} \sim \pi\right]$$

where we use the notation $a_{h:H} \sim \pi$ to mean that all actions $a_h, a_{h+1}, \dots a_H$ to be taken according to π, that is $a_t = \pi(s_t, t)$ for $t \in \{h, h+1, \dots H\}$. These functions tell us how much total reward a policy is expected to achieve until the end of the episode from a certain state or state-action pair. Value functions satisfy the following relations

$$\begin{aligned} V_h^\pi(s) &= Q_h^\pi(s, \pi(s,h)) \\ Q_h^\pi(s,a) &= \mathbb{E}\left[r_h + V_{h+1}^\pi(s_{h+1}) \mid s_h = s,\ a_h = a\right] \\ &= r(s,a) + \mathbb{E}_{s' \sim P(s,a)}\left[V_{h+1}^\pi(s')\right] = r(s,a) + P(s,a)V_{h+1}^\pi. \end{aligned} \tag{2.1}$$

It is understood here that $V_{H+1}^\pi(s) = 0$ for all $s \in \mathcal{S}$ and we use a handy notation that interprets $P(s,a)f := \sum_{s' \in \mathcal{S}} P(s'|s,a) f(s')$ as a linear functional that maps state functions $f : \mathcal{S} \to \mathbb{R}$ to reals $\mathbb{R}$. To write Equation (2.1) even more concisely, we often even omit the state and action inputs and write it in the functional form as

$$Q_h^\pi = r + PV_{h+1}^\pi.$$

The expected return $\rho(\pi)$ of a policy is simply the value function of the initial states, that is $\rho(\pi) = \mathbb{E}_{s \sim p_0}[V_1^\pi(s)]$.

We call a policy optimal if it achieves the maximum value for all time steps and states. That is $\pi^\star$ is optimal if and only if

$$V_h^{\pi^\star}(s) = \max_\pi V_h^\pi(s) \qquad \forall\, h \in [H],\ s \in \mathcal{S}.$$

There always exist an optimal policy that is deterministic and Markov (Puterman, 1994), but optimal policies are not necessarily unique. Their values are unique, however, and we use the following short-hand notations for these optimal value- and Q-functions

$$V_h^\star = \max_\pi V_h^\pi \qquad\qquad Q_h^\star = \max_\pi Q_h^\pi.$$

The optimal value function also satisfy the following identity, known as *Bellman equation*,

$$\begin{aligned} Q_h^\star(s,a) &= r(s,a) + P(s,a)V_{h+1}^\star \\ V_h^\star(s) &= \max_{a\in\mathcal{A}} Q_h^\star(s,a). \end{aligned}$$

This relation allows us to see an alternative definition of optimal policies: The set of optimal policies is exactly the set of greedy policies for $Q^\star$, i.e., those that pick an action from $\operatorname{argmax}_{a\in\mathcal{A}} Q_h^\star(s,a)$ in each state s and time step h. The Bellman equation also gives us a means to compute the optimal Q and value function for a given MDP by dynamic programming: Start by computing $Q_H^\star(s,a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}$, then $V_H^\star(s)$ for all $s \in \mathcal{S}$ and then move on to earlier time steps, $Q_{H-1}^\star, V_{H-1}^\star, Q_{H-2}^\star, V_{H-2}^\star \dots$ until $Q_1^\star$ and $V_1^\star$. Computing the optimal values and policy for a given MDP where we have full access to the reward distributions and transition probabilities is also known as *planning* which is in contrast to *reinforcement learning*.

2.2 Problem Setting: Reinforcement learning in episodic fixed-horizon MDPs

The main problem setting in this dissertation is *reinforcement learning in tabular episodic finite-horizon MDPs*. Here an agent interacts with an environment as prescribed by an episodic finite-horizon MDP (see above). The agent is assumed to

- know the state space $\mathcal{S}$, action space $\mathcal{A}$ and time horizon H
- does not know the transition distribution P, initial state distribution p_0 and reward distribution $\mathcal{R}$.

The agent can only learn about these distributions through interaction. The agent's goal is to **learn good policies as quickly as possible through interaction**. This informally stated objective is interpreted in several ways in the existing literature. It is most natural to contrast these different view-points by comparing the formal evaluation criteria or performance guarantees used for algorithms in those viewpoints, which we will do in the next section.

2.3 Existing Theoretical Learning Guarantees

Before discussing the different formalizations and evaluations of reinforcement learning, it is useful to introduce some helpful notation. The *optimal (expected) return* is the best achievable expected return $\rho^\star = \max_{\pi\in\Pi} \rho(\pi)$ and a reference point for any algorithm. We use π_k to denote the policy played by algorithm in the k-th episode. The *optimality gap* is then defined as the difference between best and achieved expected return

$$\Delta_k := \rho^\star - \rho(\pi_k) = \mathbb{E}\left[\sum_{t=1}^H r_{k,t} \,\middle|\, a_{k,t} = \pi^\star(s_{k,t},t)\right] - \mathbb{E}\left[\sum_{t=1}^H r_{k,t} \,\middle|\, a_{k,t} = \pi_k(s_{k,t},t)\right].$$

It is important to note that this assumes that the algorithm follows a fixed policy for the entire episode and that the optimality gap measures the *expected* difference in the sum of rewards given the algorithm's policy.

2.3.1 Probably-Approximately Correct (PAC) Learning

Probably-approximately correct (PAC) learning was introduced by Valiant (1984) in the context of supervised learning. It is primarily concerned with the *sample-complexity* of a machine learning algorithm. In the context of supervised learning, the sample-complexity tells us the following: Given a fixed desired accuracy $\epsilon > 0$ and a failure tolerance $\delta > 0$, the sample-complexity is the number of training samples required by the algorithm to guarantee that its predictor has test error at most ϵ with probability at least $1-\delta$. An algorithm is said to PAC-learn a concept class (like bounded linear functions of certain dimension) if it has polynomial sample-complexity in $1/\epsilon$ and $1/\delta$. Formally this also requires the algorithm to have polynomial computational complexity but this condition is often ignored. There are two main adaptations of sample-complexity and PAC learning to reinforcement learning:

Supervised-style PAC bounds The first is what we call supervised-style PAC bounds. Here, the learning protocol is as follows: The learning algorithm gets as input a desired accuracy ϵ and failure probability δ. It then interacts with the environment until it decides to stop and return a policy π. With probability at least $1-\delta$, this policy has to be ϵ-optimal, i.e., $\rho(\pi) \geq \rho^\star - \epsilon$ and the number of episodes (or total number of time steps) the algorithm interacted with the environment is called sample-complexity. A PAC bound is a polynomial upper-bound on the sample-complexity and in the case of episodic finite MDPs depends on the number of states S, the number of actions A, the episode length H, the inverse accuracy $1/\epsilon$ and inverse failure probability $1/\delta$. A slightly alternative version of this protocol is where the algorithm does not explicitly stop but continues to a play policies indefinitely that are guaranteed to be at least ϵ-optimal with high probability. As long as the algorithm knows its sample-complexity, this is an equivalent notion. We make this version for our problem setting concrete with the following definition:

Definition 1 (Supervised-style PAC bound)**.** *An algorithm satisfies a supervised-style PAC bound $F(1/\epsilon, 1/\delta, \dots)$ if for a given input $\epsilon, \delta > 0$, it satisfies the following condition for any episodic fixed-horizon MDP. With probability at least $1-\delta$, the algorithm plays only policies that are at least ϵ-optimal after at most F episodes. That is, with probability at least $1-\delta$*

$$\max\{k \in \mathbb{N} \,:\, \Delta_k > \epsilon\} \leq F(1/\epsilon, 1/\delta, \dots),$$

where F is a polynomial that can depend on properties of the problem instance.

Supervised-style PAC bounds are for example used by Kearns and Singh (2002), Brafman and Tennenholtz (2002), and Jiang, Krishnamurthy, et al. (2017) and in Chapter 6 for the `VALOR` algorithm.

Mistake-style PAC bounds The second type of PAC bounds are slightly weaker and more popular than supervised-style PAC bounds and have a flavor of mistake bounds. As before, the algorithm gets as input a desired accuracy ϵ and failure probability δ. It then interacts with the environment forever. Here, the sample-complexity is the number of episodes the algorithm may not follow a policy that is at least ϵ-optimal with probability at least $1-\delta$. As before, mistake-style PAC bounds are polynomial upper-bounds on this notion of sample-complexity. This notion of PAC bound is weaker as it does not prescribe *when* "mistakes" – an episode with optimality gap $\Delta_k > \epsilon$ – happen while supervised-style PAC bounds prescribe that mistakes need to happen in the beginning.

Definition 2 (Mistake-style PAC bound)**.** *An algorithm satisfies a mistake-style PAC bound $F(1/\epsilon, 1/\delta, \dots)$ if for a given input $\epsilon, \delta > 0$, it satisfies the following condition for any episodic fixed-horizon MDP. With*

probability at least $1-\delta$*, the algorithm plays policies that are not at least* ϵ*-optimal in at most* F *episodes. That is, with probability at least* $1-\delta$

$$\sum_{k=1}^{\infty} \mathbf{1}\{\Delta_k > \epsilon\} \leq F(1/\epsilon, 1/\delta, \dots),$$

where F *is a polynomial that can depend on properties of the problem instance.*

Examples of mistake-style PAC bounds in reinforcement learning include Strehl, Li, Wiewiora, et al. (2006), Strehl, Li, and Littman (2009), Szita and Szepesvári (2010), and Lattimore and Hutter (2012) as well as our first sample-complexity result for episodic fixed-horizon MDPs in Chapter 3.

2.3.2 No-Regret Learning

Using regret as an evaluation criterion for algorithms originates in online learning (Mohri, Rostamizadeh, and Talwalkar, 2018). For episodic reinforcement learning, the most commonly used definition of regret is the cumulative sum of optimality gaps. That is, the regret of an algorithm after T episodes is

$$R(T) = \sum_{k=1}^{T} \Delta_k = T\rho^\star - \sum_{k=1}^{T} \rho(\pi_k).$$

If $R(T)$ is sub-linear in T, i.e., the average suboptimality goes to zero, $R(T)/T \to 0$, one calls the algorithm a no-regret learner. One can distinguish two learning protocols. In the first, the algorithm receives the T as an input and interacts with the environment of exactly T episodes. In the second, T is not provided and the algorithm continues to interact with the environment indefinitely. It is important to realize that the regret $R(T)$ is a random quantity because, even though it considers expected sum of rewards per episode given the policy, the sequence of policies $\pi_1, \dots, \pi_T$ is random. This gives rise to different notions of regret bounds:

Definition 3 (Expected regret bound)**.** *An algorithm satisfies an expected bound* $F(T, \dots)$ *if it satisfies for any episodic fixed-horizon MDP the following condition. The expected regret is bounded for all* $T \in \mathbb{N}$ *as*

$$\mathbb{E}\left[R(T)\right] \leq F(T, \dots) \qquad \forall\, T \in \mathbb{N}$$

where F *can depend on properties of the problem instance.*

Definition 4 (High-probability regret bound)**.** *An algorithm satisfies a high-probability regret bound* $F(T, 1/\delta, \dots)$ *if for a given input* $T \in \mathbb{N}$ *it satisfies for any episodic fixed-horizon MDP the following condition. With probability at least* $1-\delta$*, the regret after* T *episodes is bounded as*

$$R(T) \leq F(T, 1/\delta, \dots)$$

where F *can depend on properties of the problem instance.*

Definition 5 (Uniform high-probability regret bound)**.** *An algorithm satisfies a uniform high-probability regret bound* $F(T, 1/\delta, \dots)$ *if it satisfies for any episodic fixed-horizon MDP the following condition. With probability at least* $1-\delta$*, the regret after* T *episodes* for all episodes T *is bounded as*

$$R(T) \leq F(T, 1/\delta, \dots) \qquad \forall\, T \in \mathbb{N}$$

where F *can depend on properties of the problem instance.*

It is obvious that a uniform-high probability regret bound is a stronger statement than a high-probability regret bound. For an extended discussion of these different bounds, see Chapter 4. We can find each type of regret bound used in he literature in a variety of settings: (uniform) high-probability regret: (Azar, Osband, and Munos, 2017; Zanette and Brunskill, 2019; Jaksch, Ortner, and Auer, 2010; Agarwal, Hsu, et al., 2014; Srinivas et al., 2010); expected regret: (Russo, 2019; Audibert, Munos, and Szepesvári, 2009; Auer, 2000; Bubeck and Cesa-Bianchi, 2012; Auer and Ortner, 2005)).

2.3.3 Our Focus: Worst-Case Problem-Independent Bounds

Worst-case vs. Bayesian guarantees. All types of learning guarantees are formulated as worst-case bounds which hold for any problem instance in the class (and finite episodic fixed-horizon MDP). Especially for Bayesian algorithms like those using Thompson sampling it can be more natural to provide Bayesian guarantees (Osband, Russo, and Van Roy, 2013; Osband and Van Roy, 2017). These typically come in the form of expected regret bounds that hold only in expectation over the problem instance sampled from the assumed prior belief. As Bayesian guarantees are weaker, we focus in this dissertation on the stronger worst-case guarantees.

Problem-independent vs. problem-dependent bounds. One distinguishes between two types of bounds. Those that depend only on properties of the considered problem class that are known to the agent (like number of states S and actions A and the horizon H) are called problem-independent bounds. Bounds that also depend on properties of the specific problem instance (like variance of the optimal value function Zanette and Brunskill (2019)) are called problem-dependent. The work in this dissertation is primarily concerned with problem-independent guarantees.

2.4 Helpful Notation

The notation $\tilde{O}$ is similar to the usual O-notation but ignores log-terms. More precisely $f = \tilde{O}(g)$ if there are constants c_1, c_2 such that $f \leq c_1 g(\ln g)^{c_2}$ and analogously for $\tilde{\Omega}$. The natural logarithm is $\ln$ and $\log = \log_2$ is the base-2 logarithm.

Chapter 3

Horizon-Optimal PAC Bounds for Episodic Reinforcement Learning

This chapter is based on the work published as:
Christoph Dann and Emma Brunskill. "Sample complexity of episodic fixed-horizon reinforcement learning". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2818–2826

3.1 Introduction and Motivation

Consider test preparation software that tutors students for a national advanced placement exam taken at the end of a year, or maximizing business revenue by the end of each quarter. Each individual task instance requires making a sequence of decisions for a fixed number of steps H (e.g., tutoring one student to take an exam in spring 2015 or maximizing revenue for the end of the second quarter of 2014). Therefore, they can be viewed as a finite-horizon sequential decision making under uncertainty problem, in contrast to an infinite horizon setting in which the number of time steps is infinite. When the domain parameters (e.g. Markov decision process parameters) are not known in advance, and there is the opportunity to repeat the task many times (teaching a new student for each year's exam, maximizing revenue for each new quarter), this can be treated as episodic fixed-horizon reinforcement learning (RL). One important question is to understand how much experience is required to act well in this setting. We formalize this as the sample complexity of reinforcement learning (Strehl and Littman, 2005), which is the number of time steps on which the algorithm may select an action whose value is not near-optimal. RL algorithms with a sample complexity that is a polynomial function of the domain parameters are referred to as Probably Approximately Correct (PAC) (see Section 2.3 and Kearns and Koller, 1999; Brafman and Tennenholtz, 2003; Kakade, 2003; Strehl and Littman, 2005). Though there has been significant work on PAC RL algorithms for the infinite horizon setting, there has been relatively little work on the finite horizon scenario.

In this chapter we present the first lower bound, and a new upper bound on the sample complexity of episodic finite horizon PAC reinforcement learning in discrete state-action spaces. Our bounds are tight up to log-factors in the time horizon H, the accuracy ϵ, the number of actions A and up to an additive constant in the failure probability δ. These bounds improve upon existing results by a factor of at least H. Our results also apply when the reward model is a function of the within-episode time step in addition to the state and action space. While we assume a stationary transition model, our results can be extended readily to time-dependent state-transitions. Our proposed `UCFH` (Upper-confidence fixed-horizon RL) algorithm that achieves our upper PAC guarantee can be applied directly to wide range of fixed-horizon episodic

MDPs with known rewards.[1] It does not require additional structure such as assuming access to a generative model (Azar, Munos, and Kappen, 2012) or that the state transitions are sparse or acyclic (Lattimore and Hutter, 2012).

The limited prior research on upper bound PAC results for finite horizon MDPs has focused on different settings, such as partitioning a longer trajectory into fixed length segments (Kakade, 2003; Strehl and Littman, 2005), or considering a sliding time window (Kolter and Ng, 2009). The tightest dependence on the horizon in terms of the number of episodes presented in these approaches is at least H^3 whereas our dependence is only H^2. More importantly, such alternative settings require the optimal policy to be stationary, whereas in general in finite horizon settings the optimal policy is nonstationary (e.g. is a function of both the state and the within episode time-step).[2] Fiechter (Fiechter, 1994; Fiechter, 1997) and Reveliotis and Bountourelis (2007) do tackle a closely related setting, but find a dependence that is at least H^4.

Our work builds on recent work (Lattimore and Hutter, 2012; Azar, Munos, and Kappen, 2012) on PAC infinite horizon discounted RL that offers much tighter upper and lower sample complexity bounds than was previously known. To use an infinite horizon algorithm in a finite horizon setting, a simple change is to augment the state space by the time step (ranging over $1, \ldots, H$), which enables the learned policy to be non-stationary in the original state space (or equivalently, stationary in the newly augmented space). Unfortunately, since these recent bounds are in general a quadratic function of the state space size, the proposed state space expansion would introduce at least an additional H^2 factor in the sample complexity term, yielding at least a H^4 dependence in the number of episodes for the sample complexity.

Somewhat surprisingly, we prove an upper bound on the sample complexity for the finite horizon case that only scales quadratically with the horizon. A key part of our proof is that the variance of the value function in the finite horizon setting satisfies a Bellman equation. We also leverage recent insights that state–action pairs can be estimated to different precisions depending on the frequency to which they are visited under a policy, extending these ideas to also handle when the policy followed is nonstationary. Our lower bound analysis is quite different than some prior infinite-horizon results, and involves a construction of parallel multi-armed bandits where it is required that the best arm in a certain portion of the bandits is identified with high probability to achieve near-optimality.

3.2 Problem Setting and Notation

We consider episodic fixed-horizon MDPs as introduced in Chapter 2. As a brief reminder, these MDPs can be formalized as a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, P, p_0, H)$. Both, the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are finite sets. The learning agent interacts with the MDP in episodes of H time steps. At time $t = 1 \ldots H$, the agent observes a state s_t and chooses an action a_t based on a policy π that potentially depends on the within-episode time step, i.e., $a_t = \pi(s_t, t)$ for $t = 1, \ldots, H$. The next state is sampled from the stationary transition kernel $s_{t+1} \sim P(\cdot|s_t, a_t)$ and the initial state from $s_1 \sim p_0$. In addition the agent receives a reward drawn from a distribution $\mathcal{R}(s_t)$ [3] with mean $r(s_t)$ determined by the reward function. The reward function r is possibly time-dependent (i.e., we denote it by r_t in this case) and takes values in $[0, 1]$. The quality of a policy π is evaluated by the *total expected reward* of an episode $\rho_M(\pi) = \mathbb{E}\left[\sum_{t=1}^{H} r_t(s_t)\right]$.

[1] Previous works (Auer and Ortner, 2005) have shown that the complexity of learning state transitions usually dominates learning reward functions. We therefore follow existing sample complexity analyses (Lattimore and Hutter, 2012; Szita and Szepesvári, 2010) and assume known rewards for simplicity. The algorithm and PAC bound can be extended readily to the case of unknown reward functions.

[2] The best action will generally depend on the state and the number of remaining time steps. In the tutoring example, even if the student has the same state of knowledge, the optimal tutor decision may be to space practice if there is many days till the test and provide intensive short-term practice if the test is tomorrow.

[3] It is straightforward to have the reward depend on the state, or state/action or state/action/next state.

When the MDP M is unambiguous, we omit the subscript. For simplicity,[1] we assume that the reward function r is known to the agent but the transition kernel P is unknown. The question we study is how many episodes does a learning agent follow a policy π that is not ϵ-optimal, i.e., we look for a mistake-style PAC bound (see Definition 2).

Notation. In the following sections, we reason about the true MDP M, an empirical MDP $\hat{M}$ and an optimistic MDP $\tilde{M}$ which are identical except for their transition probabilities P, $\hat{P}$ and $\tilde{P}_t$. We will provide more details about these MDPs later. We introduce the notation explicitly only for M but the quantities carry over to $\tilde{M}$ and $\hat{M}$ with additional tildes or hats by replacing P with $\tilde{P}_t$ or $\hat{P}$. We add a time index t as a subscript to the optimistic transition probabilities as this MDP can have time-dependent dynamics (see technical details below).

The (linear) operator $P_i^\pi f(s) := \mathbb{E}[f(s_{i+1})|s_i = s] = \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s,i)) f(s')$ takes any function $f : \mathcal{S} \to \mathbb{R}$ and returns the expected value of f with respect to the next time step.[4] For convenience, we define the multi-step version as $P_{i:j}^\pi f := P_i^\pi P_{i+1}^\pi \dots P_j^\pi f$. The value function from time h on is defined as $V_h^\pi(s) := \mathbb{E}\left[\sum_{t=h}^H r_t(s_t)|s_i = s\right] = \sum_{t=h}^H P_{h:t-1}^\pi r_t = \left(P_h^\pi V_{h+1}^\pi\right)(s) + r_h(s)$ and $V_h^\star$ is the optimal value-function. When the policy is clear, we omit the superscript π.

We denote by $\mathcal{S}(s,a) \subseteq \mathcal{S}$ the set of possible successor states of state s and action a. The maximum number of them is denoted by $C = \max_{s,a \in \mathcal{S} \times \mathcal{A}} |\mathcal{S}(s,a)|$. In general, without making further assumptions, we have $C = S$, though in many practical domains (robotics, user modeling) each state can only transition to a subset of the full set of states (e.g. a robot can't teleport across the building, but can only take local moves). The notation $\tilde{O}$ is similar to the usual O-notation but ignores log-terms. More precisely $f = \tilde{O}(g)$ if there are constants c_1, c_2 such that $f \leq c_1 g (\ln g)^{c_2}$ and analogously for $\tilde{\Omega}$. The natural logarithm is $\ln$ and $\log = \log_2$ is the base-2 logarithm.

3.3 Upper PAC-Bound

We now introduce a new model-based algorithm, `UCFH`, for RL in finite horizon episodic domains (Algorithm 1). We will later prove `UCFH` is PAC with an upper bound on its sample complexity that is smaller than prior approaches. Like many other PAC RL algorithms (Brafman and Tennenholtz, 2002; Strehl, Li, Wiewiora, et al., 2006; Strehl, Li, and Littman, 2009; Auer, Jaksch, and Ortner, 2009), `UCFH` uses an optimism under uncertainty approach to balance exploration and exploitation. The algorithm generally works in phases comprised of optimistic planning, policy execution and model updating that take several episodes each. Phases are indexed by k. As the agent acts in the environment and observes (s, a, r, s') tuples, `UCFH` maintains a confidence set over the possible transition parameters for each state-action pair that are consistent with the observed transitions. Defining such a confidence set that holds with high probability can be be achieved using concentration inequalities like the Hoeffding inequality. One innovation in our work is to use a particular new set of conditions to define the confidence set that enables us to obtain our tighter bounds. We will discuss the confidence sets further below. The collection of these confidence sets together form a class of MDPs $\mathcal{M}_k$ that are consistent with the observed data. We define $\hat{M}_k$ as the maximum likelihood estimate of the MDP given the previous observations.

Given $\mathcal{M}_k$, `UCFH` computes a policy π_k by performing optimistic planning. Specifically, we use a finite horizon variant of extended value iteration (EVI) Auer and Ortner, 2005; Strehl and Littman, 2005. EVI performs modified Bellman backups that are optimistic with respect to a given set of parameters. That is, given a confidence set of possible transition model parameters, it selects in each time step the model

[4] The definition also works for time-dependent transition probabilities.

within that set that maximizes the expected sum of future rewards. Section 3.7 provides more details about fixed horizon EVI.

UCFH then executes π_k until there is a state-action pair (s, a) that has been visited often enough since its last update (defined precisely in the until-condition in UCFH). After updating the model statistics for this (s, a)-pair, a new policy π_{k+1} is obtained by optimistic planning again. We refer to each such iteration of planning-execution-update as a *phase* with index k. If there is no ambiguity, we omit the phase indices k to avoid cluttered notation.

UCFH is inspired by the infinite-horizon UCRL-γ algorithm by Lattimore and Hutter (2012) but has several important differences. First, the policy can only be updated at the end of an episode, so there is no need for explicit delay phases as in UCRL-γ. Second, the policies π_k in UCFH are time-dependent. Finally, UCFH can directly deal with non-sparse transition probabilities, whereas UCRL-γ only directly allows two possible successor states for each (s, a)-pair ($C = 2$).

Confidence sets. The class of MDPs $\mathcal{M}_k$ consists of fixed-horizon MDPs M' with the known true reward function r and where the transition probability $p'_t(s'|s, a)$ from any $(s, a) \in \mathcal{S} \times \mathcal{A}$ to $s' \in \mathcal{S}(s, a)$ at any time t is in the confidence set induced by $\hat{p}(s'|s, a)$ of the empirical MDP $\hat{M}$. Solely for the purpose of computationally more efficient optimistic planning, we allow time-dependent transitions (allows choosing different transition models in different time steps to maximize reward), but this does not affect the theoretical guarantees as the true stationary MDP is still in $\mathcal{M}_k$ with high probability. Unlike the confidence intervals used by Lattimore and Hutter (2012), we not only include conditions based on Hoeffding's inequality[5] and Bernstein's inequality (Eq. 3.2), but also require that the standard deviation $\sqrt{p(1-p)}$ of the Bernoulli random variable associated with this transition is close to the empirical one (Eq. 3.1). This additional condition (Eq. 3.1) is key for making the algorithm directly applicable to generic MDPs (in which states can transition to any number of next states, e.g. $C > 2$) while only having a linear dependency on C in the PAC bound.

3.3.1 PAC Analysis

For simplicity we assume that each episode starts in a fixed start state s_0. This assumption is not crucial and can easily be removed by additional notational effort.

Theorem 6. *For any $0 < \epsilon, \delta \leq 1$, the following holds. With probability at least $1 - \delta$, UCFH produces a sequence of policies π_k, that yield at most*

$$\tilde{O}\left(\frac{SAH^2C}{\epsilon^2} \ln \frac{1}{\delta}\right)$$

episodes with $\rho^\star - \rho(\pi_k) = V_1^\star(s_0) - V_1^{\pi_k}(s_0) > \epsilon$. The maximum number of possible successor states is denoted by $1 < C \leq S$.

Similarities to other analyses. The proof of Theorem 6 is quite long and involved, but builds on similar techniques for sample-complexity bounds in reinforcement learning (see e.g. Brafman and Tennenholtz (2002) and Strehl and Littman (2008)). The general proof strategy is closest to the one of UCRL-γ (Lattimore and Hutter, 2012) and the obtained bounds are similar if we replace the time horizon H with the equivalent in the discounted case $1/(1-\gamma)$. However, there are important differences that we highlight now briefly.

[5]The first condition in the min in Equation (3.2) is actually not necessary for the theoretical results to hold. It can be removed and all $6/\delta_1$ can be replaced by $4/\delta_1$.

Algorithm 1: `UCFH`: **U**pper-**C**onfidence **F**ixed-**H**orizon episodic reinforcement learning algorithm

Input : desired accuracy $\epsilon \in (0,1]$, failure tolerance $\delta \in (0,1]$, fixed-horizon MDP M

Result: with probability at least $1-\delta$: ϵ-optimal policy

$k := 1, \quad w_{\min} := \frac{\epsilon}{4HS}, \quad \delta_1 := \frac{\delta}{2U_{\max}C}, \quad U_{\max} := SA \log_2 \frac{SH}{w_{\min}}$;

$m := 512(\log_2 \log_2 H)^2 \frac{CH^2}{\epsilon^2} \log^2 \left(\frac{8H^2S^2}{\epsilon}\right) \ln \frac{6SAC \log_2^2(4S^2H^2/\epsilon)}{\delta}$;

$n(s,a) = v(s,a) = n(s,a,s') := 0 \quad \forall, s \in \mathcal{S}, a \in \mathcal{A}, s' \in \mathcal{S}(s,a)$;

while do

`/* Optimistic planning */`

$\hat{P}(s'|s,a) := n(s,a,s')/n(s,a)$, for all (s,a) with $n(s,a) > 0$ and $s' \in \mathcal{S}(s,a)$;

$\mathcal{M}_k := \big\{\tilde{M} \in \mathcal{M}_{\text{nonst.}} : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, t = 1 \dots H, s' \in \mathcal{S}(s,a)$

$\tilde{P}_t(s'|s,a) \in \texttt{ConfidenceSet}\,(\hat{P}(s'|s,a), n(s,a))\big\}$;

$\tilde{M}_k, \pi_k := \texttt{FixedHorizonEVI}\,(\mathcal{M}_k)$;

`/* Execute policy */`

repeat

`SampleEpisode`(π_k) ; `// from` M `using` π_k

until *there is a* $(s,a) \in \mathcal{S} \times \mathcal{A}$ *with* $v(s,a) \geq \max\{mw_{\min}, n(s,a)\}$ *and* $n(s,a) < SmH$;

`/* Update model statistics for one` (s,a)`-pair with condition above */`

$n(s,a) := n(s,a) + v(s,a)$;

$n(s,a,s') := n(s,a,s') + v(s,a,s') \quad \forall s' \in \mathcal{S}(s,a)$;

$v(s,a) := v(s,a,s') := 0 \quad \forall s' \in \mathcal{S}(s,a)$;

$k := k + 1$

Procedure `SampleEpisode`(π)

$s_0 \sim p_0$;

for $t = 0$ **to** $H-1$ **do**

$a_t := \pi(s_t, t)$ and $s_{t+1} \sim p(\cdot|s_t, a_t)$;

$v(s_t, a_t) := v(s_t, a_t) + 1$ and $v(s_t, a_t, s_{t+1}) := v(s_t, a_t, s_{t+1}) + 1$;

Function `ConfidenceSet`(p, n)

$$\mathcal{P} := \Bigg\{ p' \in [0,1] : \text{if } n > 1 : \left|\sqrt{p'(1-p')} - \sqrt{p(1-p)}\right| \leq \sqrt{\frac{2\ln(6/\delta_1)}{n-1}}, \tag{3.1}$$

$$|p - p'| \leq \min\left(\sqrt{\frac{\ln(6/\delta_1)}{2n}}, \sqrt{\frac{2p(1-p)}{n}\ln(6/\delta_1)} + \frac{7}{3(n-1)}\ln\frac{6}{\delta_1}\right)\Bigg\} \tag{3.2}$$

return $\mathcal{P}$

- A central quantity in the analysis by Lattimore and Hutter (2012) is the local variance of the value function. The exact definition for the fixed-horizon case will be given below. The key insight for the almost tight bounds of Lattimore and Hutter (2012) and Azar, Munos, and Kappen (2012) is to leverage the fact that these local variances satisfy a Bellman equation (Sobel, 1982) and so the discounted sum of local variances can be bounded by $O((1-\gamma)^{-2})$ instead of $O((1-\gamma)^{-3})$. We prove in Lemma 10 that local value function variances σ_h^2 also satisfy a Bellman equation for fixed-horizon MDPs even if transition probabilities and rewards are time-dependent. This allows us to bound the total sum of local variances by $O(H^2)$ and obtain similarly strong results in this setting.
- Lattimore and Hutter (2012) assumed there are only two possible successor states (i.e., $C = 2$) which allows them to easily relate the local variances σ_h^2 to the difference of the expected value of successor states in the true and optimistic MDP $(P - \tilde{P}_h)\tilde{V}_{h+1}$. For $C > 2$, the relation is less clear, but we address this by proving a bound with tight dependencies on C (Lemma 18).
- To avoid super-linear dependency on C in the final PAC bound, we add the additional condition in Equation (3.1) to the confidence set. We show that this allows us to upper-bound the optimality $\rho^\star - \rho(\pi_k)$ of policy π_k with terms that either depend on σ_h^2 or decrease linearly in the number of samples. This gives the desired linear dependency on C in the final bound. We therefore avoid assuming $C = 2$ which makes `UCFH` directly applicable to generic MDPs with $C > 2$ without the impractical transformation argument used by Lattimore and Hutter (2012).

We will now introduce the notion of *knownness* and *importance* of state-action pairs that is essential for the analysis of `UCFH` and subsequently present several lemmas necessary for the proof of Theorem 6. We only sketch proofs here but detailed proofs for all results are available in Section 3.9.

Fine-grained categorization of (s,a)-pairs. Many PAC RL sample complexity proofs (Brafman and Tennenholtz, 2002; Kakade, 2003; Strehl, Li, Wiewiora, et al., 2006; Strehl and Littman, 2008) only have a binary notion of "knownness", distinguishing between known (transition probability estimated sufficiently accurately) and unknown (s,a)-pairs. However, as recently shown by Lattimore and Hutter (2012) for the infinite horizon setting, it is possible to obtain much tighter sample complexity results by using a more fine grained categorization. In particular, a key idea is that in order to obtain accurate estimates of the value function of a policy from a starting state, it is sufficient to have only a loose estimate of the parameters of (s,a)-pairs that are unlikely to be visited under this policy.

Let the *weight* of a (s,a)-pair given policy π_k be its expected frequency in an episode

$$w_k(s,a) := \sum_{t=1}^{H} \mathbb{P}(s_t = s, \pi_k(s_t,t) = a) = \sum_{t=1}^{H} P_{1:t-1}\mathbb{I}\{s = \cdot, a = \pi_k(s,t)\}(s_0).$$

The *importance* ι_k of (s,a) is its relative weight compared to $w_{\min} := \frac{\epsilon}{4HS}$ on a log-scale

$$\iota_k(s,a) := \min\left\{z_i \,:\, z_i \geq \frac{w_k(s,a)}{w_{\min}}\right\} \quad \text{where } z_1 = 0 \text{ and } z_i = 2^{i-2}\ \forall i = 2,3,\ldots.$$

Note that $\iota_k(s,a) \in \{0,1,2,4,8,16\ldots\}$ is an integer indicating the influence of the state-action pair on the value function of π_k. Similarly, we define the *knownness*

$$\kappa_k(s,a) := \max\left\{z_i \,:\, z_i \leq \frac{n_k(s,a)}{m w_k(s,a)}\right\} \in \{0,1,2,4,\ldots\}$$

which indicates how often (s,a) has been observed relative to its importance. The constant m is defined in Algorithm 1. We can now categorize (s,a)-pairs into subsets

$$X_{k,\kappa,\iota} := \{(s,a) \in X_k \,:\, \kappa_k(s,a) = \kappa, \iota_k(s,a) = \iota\} \quad \text{and} \quad \bar{X}_k = \mathcal{S} \times \mathcal{A} \setminus X_k$$

where $X_k = \{(s,a) \in \mathcal{S} \times \mathcal{A} \,:\, \iota_k(s,a) > 0\}$ is the *active set* and $\bar{X}_k$ the set of state-action pairs that are very unlikely under the current policy. Intuitively, the model of `UCFH` is accurate if only few (s,a) are in categories with low knownness – that is, important under the current policy but have not been observed often so far. Recall that over time observations are generated under many policies (as the policy is recomputed), so this condition does not always hold. We will therefore distinguish between phases k where $|X_{k,\kappa,\iota}| \leq \kappa$ for all κ and ι and phases where this condition is violated. The condition essentially allows for only a few (s,a) in categories that are less known and more and more (s,a) in categories that are more well known. In fact, we will show that the policy is ϵ-optimal with high probability in phases that satisfy this condition.

We first show the validity of the confidence sets $\mathcal{M}_k$.

Lemma 7 (Capturing the true MDP whp.)**.** *$M \in \mathcal{M}_k$ for all k with probability at least $1 - \delta/2$.*

Proof Sketch. By combining Hoeffding's inequality, Bernstein's inequality and the concentration result on empirical standard deviations by Maurer and Pontil (2009) with the union bound, we get that $p(s'|s,a) \in \mathcal{P}$ with probability at least $1 - \delta_1$ for a single phase k, fixed $s,a \in \mathcal{S} \times \mathcal{A}$ and fixed $s' \in \mathcal{S}(s,a)$. We then show that the number of model updates is bounded by $U_{\max}$ and apply the union bound. □

The following lemma bounds the number of episodes in which $\forall \kappa, \iota : |X_{k,\kappa,\iota}| \leq \kappa$ is violated with high probability.

Lemma 8. *Let E be the number of episodes k for which there are κ and ι with $|X_{k,\kappa,\iota}| > \kappa$, i.e. $E = \sum_{k=1}^{\infty} \mathbb{I}\{\exists(\kappa,\iota) \,:\, |X_{k,\kappa,\iota}| > \kappa\}$ and assume that $m \geq \frac{6H^2}{\epsilon} \ln \frac{2E_{\max}}{\delta}$. Then $\mathbb{P}(E \leq 6NE_{\max}) \geq 1 - \delta/2$ where $N = SAm$ and $E_{\max} = \log_2 \frac{H}{w_{\min}} \log_2 S$.*

Proof Sketch. We first bound the total number of times a fixed pair (s,a) can be observed while being in a particular category $X_{k,\kappa,\iota}$ in all phases k for $1 \leq \kappa < S$. We then show that for a particular (κ,ι), the number of episodes where $|X_{k,\kappa,\iota}| > \kappa$ is bounded with high probability, as the value of ι implies a minimum probability of observing each (s,a) pair in $X_{k,\kappa,\iota}$ in an episode. Since the observations are not independent we use martingale concentration results to show the statement for a fixed (κ,ι). The desired result follows with the union bound over all relevant κ and ι. □

The next lemma states that in episodes where the condition $\forall \kappa, \iota : |X_{k,\kappa,\iota}| \leq \kappa$ is satisfied and the true MDP is in the confidence set, the expected optimistic policy value is close to the true value. This lemma is the technically most involved part of the proof.

Lemma 9 (Bound mismatch in total reward)**.** *Assume $M \in \mathcal{M}_k$. If $|X_{k,\kappa,\iota}| \leq \kappa$ for all (κ,ι) and $0 < \epsilon \leq 1$ and $m \geq 512 \frac{CH^2}{\epsilon^2} (\log_2 \log_2 H)^2 \log_2^2 \left(\frac{8H^2S^2}{\epsilon}\right) \ln \frac{6}{\delta_1}$. Then $|\tilde{V}_1^{\pi_k}(s_0) - V_1^{\pi_k}(s_0)| \leq \epsilon$.*

Proof Sketch. Using basic algebraic transformations, we show that $|p - \tilde{p}| \leq \sqrt{\tilde{p}(1-\tilde{p})} O\left(\sqrt{\frac{1}{n} \ln \frac{1}{\delta_1}}\right) + O\left(\frac{1}{n} \ln \frac{1}{\delta_1}\right)$ for each $\tilde{p}, p \in \mathcal{P}$ in the confidence set as defined in Eq. 3.2. Since we assume $M \in \mathcal{M}_k$, we know that $P(s'|s,a)$ and $\tilde{P}(s'|s,a)$ satisfy this bound with $n(s,a)$ for all s,a and s'. We use that to bound the difference of the expected value function of the successor state in M and $\tilde{M}$, proving that $|(P - \tilde{P}_h)\tilde{V}_{h+1}(s)| \leq O\left(\frac{CH}{n(s,\pi(s,h))} \ln \frac{1}{\delta_1}\right) + O\left(\sqrt{\frac{C}{n(s,\pi(s,h))} \ln \frac{1}{\delta_1}}\right) \tilde{\sigma}_h(s)$, where the local variance of the value

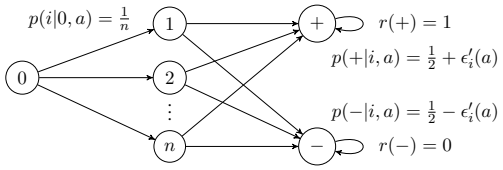

Figure 3.1: Class of a hard-to-learn finite horizon MDPs. The function ϵ' is defined as $\epsilon'(a_1) = \epsilon/2$, $\epsilon'(a_i^\star) = \epsilon$ and otherwise $\epsilon'(a) = 0$ where $a_i^\star$ is an unknown action per state i and ϵ is a parameter.

function is defined as $\sigma_h^2(s,a) := \mathbb{E}\left[(V_{h+1}^\pi(s_{h+1}) - P_h^\pi V_{h+1}^\pi(s))^2 | s_h = s, a_h = a\right]$ and $\sigma_h^2(s) := \sigma_h^2(s, \pi(s,h))$. This bound then is applied to $|\tilde{V}_1(s_0) - V_1(s_0)| \le \sum_{t=1}^H P_{1:t}|(P - \tilde{P}_t)\tilde{V}_{t+1}(s)|$. The basic idea is to split the bound into a sum of two parts by partitioning of the (s,a) space by knownness, e.g. that is $(s_t, a_t) \in \bar{X}_{\kappa,\iota}$ for all κ and ι and $(s_t, a_t) \in \bar{X}$. Using the fact that $w(s_t, a_t)$ and $n(s_t, a_t)$ are tightly coupled for each (κ, ι), we can bound the expression eventually by ϵ. The final key ingredient in the remainder of the proof is to bound $\sum_{t=1}^H P_{1:t-1}\sigma_t(s)^2$ by $O(H^2)$ instead of the trivial bound $O(H^3)$. To this end, we show the lemma below. □

Lemma 10. *The variance of the value function defined as* $\mathcal{V}_h^\pi(s) := \mathbb{E}\left[\left(\sum_{t=h}^H r_t(s_t) - V_h^\pi(s_h)\right)^2 | s_h = s\right]$ *satisfies a Bellman equation* $\mathcal{V}_h^\pi = P_h^\pi \mathcal{V}_{h+1}^\pi + \sigma_h^2$ *which gives* $\mathcal{V}_h^\pi = \sum_{t=h}^j P_{h:t-1}^\pi \sigma_t^2$. *Since* $0 \le \mathcal{V}_1^\pi \le H^2 r_{\max}^2$, *it follows that* $0 \le \sum_{t=1}^j P_{h:t-1}\sigma_t^2(s) \le H^2 r_{\max}^2$ *for all* $s \in \mathcal{S}$.

Proof Sketch. The proof works by induction and uses fact that the value function satisfies the Bellman equation and the tower-property of conditional expectations. □

Proof Sketch for Theorem 6. The proof of Theorem 6 consists of the following major parts:

1. The true MDP is in the set of MDPs $\mathcal{M}_k$ for all phases k with probability at least $1 - \frac{\delta}{2}$ (Lemma 7).
2. The `FixedHorizonEVI` algorithm computes a value function whose optimistic value is higher than the optimal reward in the true MDP with probability at least $1 - \delta/2$ (Lemma 12).
3. The number of episodes with $|X_{k,\kappa,\iota}| > \kappa$ for some κ and ι are bounded with probability at least $1 - \delta/2$ by $\tilde{O}(|\mathcal{S} \times \mathcal{A}|\, m)$ if $m = \tilde{\Omega}\left(\frac{H^2}{\epsilon} \ln \frac{S}{\delta}\right)$ (Lemma 8).
4. If $|X_{k,\kappa,\iota}| \le \kappa$ for all κ, ι, i.e., relevant state-action pairs are sufficiently known and $m = \tilde{\Omega}\left(\frac{CH^2}{\epsilon^2} \ln \frac{1}{\delta_1}\right)$, then the optimistic value computed is ϵ-close to the true MDP value. Together with part 2, we get that with high probability, the policy π_k is ϵ-optimal in this case.
5. From parts 3 and 4, with probability $1 - \delta$, there are at most $\tilde{O}\left(\frac{SAH^2C}{\epsilon^2} \ln \frac{1}{\delta}\right)$ episodes that are not ϵ-optimal.

3.4 Lower PAC Bound

Theorem 11. *There exist positive constants* c_1, c_2, δ_0, ϵ_0 *such that for every* $\delta \in (0, \delta_0)$ *and* $\epsilon \in (0, \epsilon_0)$ *and for every algorithm A that satisfies a PAC guarantee for* (ϵ, δ) *and outputs a deterministic policy, there is a fixed-horizon episodic MDP* M_{hard} *with*

$$\mathbb{E}[n_A] \ge \frac{c_1(H-2)^2(A-1)(S-3)}{\epsilon^2} \ln\left(\frac{c_2}{\delta + c_3}\right) = \Omega\left(\frac{|\mathcal{S} \times \mathcal{A}|H^2}{\epsilon^2} \ln\left(\frac{c_2}{\delta + c_3}\right)\right) \tag{3.3}$$

where n_A is the number of episodes until the algorithm's policy is (ϵ, δ)-accurate. The constants can be set to $\delta_0 = \frac{e^{-4}}{80} \approx \frac{1}{5000}$, $\epsilon_0 = \frac{H-2}{640e^4} \approx H/35000$, $c_2 = 4$ and $c_3 = e^{-4}/80$.

The ranges of possible δ and ϵ are of similar order than in other state-of-the-art lower bounds for multi-armed bandits (Mannor and Tsitsiklis, 2004) and discounted MDPs (Strehl and Littman, 2008; Lattimore and Hutter, 2012). They are mostly determined by the bandit result by Mannor and Tsitsiklis (2004) we build on. Increasing the parameter limits δ_0 and ϵ_0 for bandits would immediately result in larger ranges in our lower bound, but this was not the focus of our analysis.

Proof Sketch. The basic idea is to show that the class of MDPs shown in Figure 3.1 require at least a number of observed episodes of the order of Equation (3.3). From the start state 0, the agent ends up in states 1 to n with equal probability, independent of the action. From each such state i, the agent transitions to either a good state $+$ with reward 1 or a bad state $-$ with reward 0 and stays there for the rest of the episode. Therefore, each state $i = 1, \dots, n$ is essentially a multi-armed bandit with binary rewards of either 0 or $H-2$. For each bandit, the probability of ending up in $+$ or $-$ is equal except for the first action a_1 with $P(s_{t+1} = +|s_t = i, a_t = a_1) = 1/2 + \epsilon/2$ and possibly an unknown optimal action $a_i^\star$ (different for each state i) with $P(s_{t+1} = +|s_t = i, a_t = a_i^\star) = 1/2 + \epsilon$.

In the episodic fixed-horizon setting we are considering, taking a suboptimal action in one of the bandits does not necessarily yield a suboptimal episode. We have to consider the average over all bandits instead. In an ϵ-optimal episode, the agent therefore needs to follow a policy that would solve at least a certain portion of all n multi-armed bandits with probability at least $1 - \delta$. We show that the best strategy for the agent to achieve this is to try to solve all bandits with equal probability. The number of samples required to do so then results in the lower bound in Equation (3.3). □

Similar MDPs that essentially solve multiple of such multi-armed bandits have been used to prove lower sample-complexity bounds for discounted MDPs (Strehl and Littman, 2008; Lattimore and Hutter, 2012). However, the analysis in the infinite horizon case as well as for the sliding-window fixed-horizon optimality criterion considered by Kakade (2003) is significantly simpler. For these criteria, every time step the agent follows a policy that is not ϵ-optimal counts as a "mistake". Therefore, every time the agent does not pick the optimal arm in any of the multi-armed bandits counts as a mistake. This contrasts with our fixed-horizon setting where we must instead consider taking an average over all bandits.

3.5 Related Work on Fixed-Horizon Sample Complexity Bounds

We are not aware of any lower sample complexity bounds beyond multi-armed bandit results that directly apply to our setting. Our upper bound in Theorem 6 improves upon existing results by at least a factor of H. We briefly review those existing results in the following.

Timestep bounds. Kakade (2003, Chapter 8) proves upper and lower PAC bounds for a similar setting where the agent interacts indefinitely with the environment but the interactions are divided in segments of equal length and the agent is evaluated by the expected sum of rewards until the end of each segment. The bound states that there are not more than $\tilde{O}\left(\frac{S^2AH^6}{\epsilon^3} \ln \frac{1}{\delta}\right)$[6] time steps in which the agents acts ϵ-suboptimal. Strehl, Li, and Littman (2009) improves the state-dependency of these bounds for their delayed Q-learning algorithm to $\tilde{O}\left(\frac{SAH^5}{\epsilon^4} \ln \frac{1}{\delta}\right)$. However, in episodic MDP it is more natural to consider performance on the entire episode since suboptimality near the end of the episode is no issue as long as the total reward on the

[6]For comparison we adapt existing bounds to our setting. While the original bound stated by Kakade (2003) only has H^3, an additional H^3 comes in through ϵ^{-3} due to different normalization of rewards.

entire episode is sufficiently high. Kolter and Ng (2009) use an interesting sliding-window criterion, but prove bounds for a Bayesian setting instead of PAC. Timestep-based bounds can be applied to the episodic case by augmenting the original statespace with a time-index per episode to allow resets after H steps. This adds H dependencies for each S in the original bound which results in a horizon-dependency of at least H^6 of these existing bounds. Loosely translating the regret bounds of UCRL2 in Corollary 3 by Jaksch, Ortner, and Auer (2010) yields a PAC-like bound on the number of episodes of at least $\tilde{O}\left(\frac{S^2AH^3}{\epsilon^2}\ln\frac{1}{\delta}\right)$ even if one ignores the reset after H time steps. Timestep-based lower PAC-bounds cannot be applied directly to the episodic reward criterion.

Episode bounds. Similar to us, Fiechter (1994) uses the value of initial states as optimality-criterion, but defines the value w.r.t. the γ-discounted infinite horizon. His results of order $\tilde{O}\left(\frac{S^2AH^7}{\epsilon^2}\ln\frac{1}{\delta}\right)$ episodes of length $\tilde{O}(1/(1-\gamma)) \approx \tilde{O}(H)$ are therefore not directly applicable to our setting. Auer and Ortner (2005) investigate the same setting as we and propose a UCB-type algorithm that has no-regret, which translates into a basic PAC-like bound of order $\tilde{O}\left(\frac{S^{10}AH^7}{\epsilon^3}\ln\frac{1}{\delta}\right)$ episodes. We improve on this bound substantially in terms of its dependency on H, S and ϵ. Reveliotis and Bountourelis (2007) also consider the episodic undiscounted fixed-horizon setting and present an efficient algorithm in cases where the transition graph is acyclic and the agent knows for each state a policy that visits this state with a known minimum probability q. These assumptions are quite limiting and rarely hold in practice and their bound of order $\tilde{O}\left(\frac{SAH^4}{\epsilon^2 q}\ln\frac{1}{\delta}\right)$ explicitly depends on $1/q$.

3.6 Summary

We have shown upper and lower bounds on the sample complexity of episodic fixed-horizon RL that are tight up to log-factors in the time horizon H, the accuracy ϵ, the number of actions A and up to an additive constant in the failure probability δ. These bounds improve upon existing results by a factor of at least H. However, their dependency on the number of states S which we will address in the next chapters. Our proposed UCFH algorithm that achieves our PAC bound can be applied to directly to a wide range of fixed-horizon episodic MDPs with known rewards and does not require additional structure such as sparse or acyclic state transitions assumed in previous work.

3.7 Fixed-Horizon Extended Value Iteration

We want to find a policy π^k and optimistic $\tilde{M}_k \in \mathcal{M}_k$ which have the highest total reward $\rho_{\tilde{M}_k}(\pi_k) = \max_{\pi, M' \in \mathcal{M}_k} \rho_{M'}(\pi)$. Note that π^k is an optimal policy for M_k but not necessarily for M. To facilitate planning, we relax this problem and instead compute a policy and optimistic MDP with $\rho_{\tilde{M}_k}(\pi_k) = \max_{\pi, M' \in \mathcal{M}'_k} \rho_{M'}(\pi_)$ with

$$\mathcal{M}'_k := \big\{\tilde{M} \in \mathcal{M}_{\text{nonst.}} : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, t = 1 \dots H, s' \in \mathcal{S}(s,a) \\ \tilde{p}_t(s'|s,a) \in \text{conv}(\texttt{ConfidenceSet}\,(\hat{p}(s'|s,a), n(s,a)))\big\}.$$

Our statistical analysis only requires the transition probabilities to be in the convex hull of the confidence sets instead of the confidence sets. Since this is a relaxation, we have $\mathcal{M}_k \subseteq \mathcal{M}'_k$. We can find such a policy by dynamic programming similar to extended value iteration (Strehl and Littman, 2008; Auer and

Ortner, 2005). The optimal Q-function can be computed as $\tilde{Q}_H(s,a) = r_H(s)$ and for $i = H-1,\dots,2,1$ as

$$\tilde{Q}_h(s,a) = r_h(s) + \max_{\tilde{P}_h \in \mathcal{P}_{s,a}} \left\{ \sum_{s' \in \mathcal{S}(s,a)} \tilde{P}_h(s,a) \max_{b \in \mathcal{A}} \tilde{Q}_{h+1}(s',b) \right\}.$$

The feasible set is defined as

$$\mathcal{P}_{s,a} := \{p \in [0,1]^{|\mathcal{S}(s,a)|} \;:\; \|p\|_1 = 1, \quad \forall s' \in \mathcal{S}(s,a) : \\ p(s') \in \mathrm{conv}(\texttt{ConfidenceSet}(\hat{P}(s'|s,a), n(s,a)))\}.$$

The optimal policy $\pi_k(s,t)$ at time t is then simply the maximizer of the inner max operator and the transition probability $\tilde{P}_t(\cdot|s,a)$ is the maximizer of the outer maximum. The inner max can be solved efficiently by enumeration and the outer maximum similar to extended value iteration (Strehl and Littman, 2008). The basic idea is to put as much probability mass as possible to successor states with highest value. See the following algorithm for the implementation details. Note that due to the nonlinear constraint in

Algorithm 2: FixedHorizonEVI($\mathcal{M}$) subroutine for `UCFH`

$\tilde{Q}_H(s,a) = r_H(s) \quad \forall s,a \in \mathcal{S} \times \mathcal{A}$; // $O(SA)$
for $t = H-1$ **to** 1 **do** // $O(HS\log S + HSAC))$
  $\pi(s,t+1) := \mathrm{argmax}_{a \in \mathcal{A}} \tilde{Q}_{t+1}(s,a) \quad \forall s \in \mathcal{S}$; // $O(SA)$
  sort states $s^{(1)},\dots s^{(S)}$ such that
    $\tilde{Q}_{t+1}(s^{(i)}, \pi(s^{(i)}, t+1)) \geq \tilde{Q}_{t+1}(s^{(i+1)}, \pi(s^{(i+1)}, t+1))$; // $O(S\log S)$
  for $s,a \in \mathcal{S} \times \mathcal{A}$ **do** // $O(SAC)$
    $\tilde{P}_t(s'|s,a) := \min \texttt{ConfidenceSet}(\hat{P}(s'|s,a), n(s,a)) \quad \forall s' \in \mathcal{S}(s,a)$; // $O(C)$
    $\Delta := 1 - \sum_{s' \in \mathcal{S}(s,a)} \tilde{P}_t(s'|s,a)$; // $O(C)$
    $i := 1$; // $O(1)$
    while $\Delta > 0$ **do** // $O(C)$
      $s' := s^{(i)}$;
      $\Delta' := \min\{\Delta, \max \texttt{ConfidenceSet}(\hat{P}(s'|s,a), n(s,a)) - \tilde{P}_t(s'|s,a)\}$;
      $\tilde{P}_t(s'|s,a) := \tilde{P}_t(s'|s,a) + \Delta'$;
      $\Delta := \Delta - \Delta'; i := i+1$;
    $\tilde{Q}_t(s,a) = \sum_{s' \in \mathcal{S}(s,a)} \tilde{P}_t(s'|s,a) \tilde{Q}_{t+1}(s', \pi(s', t+1))$; // $O(C)$
$\pi(s,1) := \mathrm{argmax}_{a \in \mathcal{A}} \tilde{Q}_1(s,a) \quad \forall s \in \mathcal{S}$; // $O(SA)$
return *MDP with transition probabilities* $\tilde{p}_t$*, optimal policy* π

Equation (3.1), $\texttt{ConfidenceSet}(\hat{P}(s'|s,a), n(s,a))$ may be the union of two disjoint intervals instead of one interval. Still, min- and max-operations on the confidence sets can be computed readily in constant time. Therefore, the transition probabilities $\tilde{P}_t(\cdot|s,a)$ for a single time step t and state-action pair s,a can be computed in $O(SAC)$ given sorted states. Sorting the states takes $O(S\log S)$ which results in $O(HS\log S + HSAC)$ runtime complexity of `FixedHorizonEVI` (see comments in Function 2). The Algorithm requires $O(HSAC)$ additional space besides the storage requirements of the input MDP $\mathcal{M}$ as the transition probabilities $\tilde{P}_t$ are returned by the algorithm. If those are not required and only the optimal policy is of interest, the additional space can be reduced to $O(SA)$.

Lemma 12 (Validity of optimistic planning). `FixedHorizonEVI` $(\mathcal{M}_k)$ *returns*

$$\tilde{M}_k, \pi_k = \operatorname*{argmax}_{M \in \mathcal{M}'_k, \pi} \rho_M(\pi).$$

Since $\mathcal{M}_k \subseteq \mathcal{M}'_k$*, it also holds that* $\rho_{\tilde{M}_k}(\pi_k) \geq \max_{M \in \mathcal{M}_k, \pi} \rho_M(\pi)$.

Proof Sketch. This result can be proved straight-forwardly by showing that π_k is optimal in the last time step H with highest possible reward and then subsequently for all previous time steps inductively. It follows directly from the definition of the algorithm in Function 2 that the returned MDP is in $\mathcal{M}'_k$. □

3.8 Runtime- and Space-Complexity of `UCFH`

Sampling one episode and updating the respective v variables has $O(H)$ runtime. Each update of the policy involves updating the n variables and $\mathcal{M}_k$ which takes runtime $O(C)$ and a call of `FixedHorizonEVI` with runtime cost $O(HSAC + HS \log S)$. From Lemma 13 below, we know that the policy can be updated at most $U_{\max}$ times which a gives total runtime for policy updates of

$$\begin{aligned} O(U_{\max} HS(AC + \log S)) =& O\left(HS^2A(AC + \log S) \log \frac{S^2H^2}{\epsilon}\right) \\ =& \tilde{O}\left(HS^2A^2C \log \frac{1}{\epsilon}\right). \end{aligned}$$

The space complexity of `UCFH` is dominated by the requirement to store statistics for each possible transition which gives $O(SAC)$ complexity.

3.9 Detailed Proofs for the Upper PAC Bound

3.9.1 Bound on the Number of Policy Changes of `UCFH`

Lemma 13. *The total number of updates is bounded by* $U_{\max} = |\mathcal{S} \times \mathcal{A}| \log_2 \frac{|\mathcal{S}|H}{w_{\min}}$.

Proof. First note that $n(s,a)$ is never never decreasing and no updates happen once $n(s,a) \geq SmH$ for all (s,a). In each update, the $n(s,a)$ of exactly one (s,a) pair increases by $\max\{mw_{\min}, n(s,a)\}$. For a single (s,a) pair, such updates can happen only $\log_2(SmH) - \log_2(mw_{\min})$ times. Hence, there are at most $|\mathcal{S} \times \mathcal{A}| \log_2 \frac{SmH}{w_{\min} m}$ updates in total. □

3.9.2 Proof of Lemma 7 – Capturing the true MDP

Proof. For a single (s,a) pair, $s' \in \mathcal{S}(s,a)$ and k, we can treat the event that s' is the successor state of s when chosing action a as a Bernoulli random variable with probability $p(s'|s,a)$. Using Hoeffding's inequality,[7] we then realize that

$$|P(s'|s,a) - \hat{P}(s'|s,a)| \leq \sqrt{\frac{\ln(6/\delta_1)}{2n}}$$

[7] While the considered random variables are strictly speaking not necessarily independent, they can be treated as such for the concentration inequalities applied here. See Appendix A of Strehl and Littman (2008) for details. In the analyses in later Chapters we directly use Martingale concentration results to avoid this additional argument.

and by Bernstein's inequality

$$|P(s'|s,a) - \hat{P}(s'|s,a)| \leq \sqrt{\frac{2P(s'|s,a)(1-P(s'|s,a))\ln(6/\delta_1)}{n}} + \frac{1}{3n}\ln(6/\delta_1)$$

with probability at least $1-\delta_1/3$ respectively. Using both inequalities of Theorem 10 by Maurer and Pontil (2009)[8], we have

$$|\sqrt{P(s'|s,a)(1-P(s'|s,a))} - \sqrt{\hat{P}(s'|s,a)(1-\hat{P}(s'|s,a))}| \leq \sqrt{\frac{2\ln(6/\delta_1)}{n-1}} \tag{3.4}$$

for $n > 1$ with probability at least $1-\delta_1/3$. All three inequalities hold with probability $1-\delta_1$ by the union bound. Applying Inequality (3.4) to Bernstein's inequality, we obtain

$$\begin{aligned}
|P(s'|s,a) - \hat{P}(s'|s,a)| \leq & \sqrt{\frac{2P(s'|s,a)(1-P(s'|s,a))\ln(6/\delta_1)}{n}} + \frac{1}{3n}\ln(6/\delta_1) \\
\leq & \left(\sqrt{\hat{P}(s'|s,a)(1-\hat{P}(s'|s,a))} + \sqrt{\frac{2\ln(6/\delta_1)}{n-1}}\right)\sqrt{\frac{2\ln(6/\delta_1)}{n}} + \frac{1}{3n}\ln(6/\delta_1) \\
\leq & \sqrt{\frac{2\hat{P}(s'|s,a)(1-\hat{P}(s'|s,a))\ln(6/\delta_1)}{n}} + \frac{7}{3(n-1)}\ln(6/\delta_1).
\end{aligned}$$

By Lemma 13, there are at most $U_{\max}$ updates and so there are at most $U_{\max}$ different k to consider. Since in each update, only a single (s,a) pair with at most C successor states is updated, for all k and (s,a), there are only $U_{\max}C$ different $\hat{P}(s'|s,a)$ to consider. Applying the union bound, we get that $M_k \notin \mathcal{M}_k$ for any k with probability at most $U_{\max}C\delta_1$. By setting $\delta_1 = \frac{\delta}{2CU_{\max}}$ we get the desired result. □

3.9.3 Bounding the number of episodes with $\kappa > |X_{k,\kappa,\iota}|$ for some κ, ι

Before presenting the proof of Lemma 8 which bounds the total number of episodes where there is a κ and ι such that $\kappa > |X_{k,\kappa,\iota}|$, we establish a bound for each individual κ and ι in the following two additional lemmas.

Lemma 14 (Bound on observations of $X_{\cdot,\kappa,\iota}$)**.** *The total number of observations of $(s,a) \in X_{k,\kappa,\iota}$ where $\kappa \in [1, S-1]$ and $\iota > 0$ over all phases k is at most $3|\mathcal{S}\times\mathcal{A}|mw_\iota\kappa$. The variable w_ι is the smallest possible weight of a (s,a)-pair that has importance ι.*

Proof. We denote the smallest possible weight for any (s,a) pair such that $\iota(s,a) = \iota$ by $w_\iota := \min\{w(s,a) : \iota_k(s,a) = \iota\}$. Note that $w_{\iota+1} = 2w_\iota$ for $\iota > 0$. Consider any phase k and fix $(s,a) \in X_{k,\kappa,\iota}$ with $\iota > 0$. Since we assumed $\iota_k(s,a) = \iota > 0$, we have $w_\iota \leq w_k(s,a) < 2w_\iota$. From $\kappa_k(s,a) = \kappa$, it follows that

$$\frac{n_k(s,a)}{2mw_k(s,a)} \leq \kappa \leq \frac{n_k(s,a)}{mw_k(s,a)}$$

which implies that

$$mw_\iota\kappa \leq mw_k(s,a)\kappa \leq n_k(s,a) \leq 2mw_k(s,a)\kappa \leq 4mw_\iota\kappa. \tag{3.5}$$

Hence, each state can only be observed $3mw_\iota$ times while being in $\{(s,a) \in X_{k,\kappa,\iota} : k \in \mathbb{N}\}$. □

[8]The empirical variance denoted by $V_n(\mathbf{X})$ by Maurer and Pontil (2009) is $\hat{P}(s'|s,a)(1-\hat{P}(s'|s,a))$ in our case and $\mathbb{E}V_n$ is the true variance which amounts to $P(s'|s,a)(1-P(s'|s,a))$ for us.

Lemma 15. *The number of episodes $E_{\kappa,\iota}$ in phases with $|X_{k,\kappa,\iota}| > \kappa$ is bounded for every $\alpha \geq 3$ with high probability,*

$$\mathbb{P}(E_{\kappa,\iota} > \alpha N) \leq \exp\left(-\frac{\beta w_\iota(\kappa+1)N}{H}\right)$$

where $N = |\mathcal{S} \times \mathcal{A}|m$ and $\beta = \frac{\alpha(3/\alpha-1)^2}{7/3-1/\alpha}$.

Proof. Let $\nu_i := \sum_{t=1}^{H} \mathbb{I}\{(s_t, a_t) \in X_{k,\kappa,\iota}\}$ be the number of observations of (s, a) in $X_{k,\kappa,\iota}$ in the ith epsiode with $X_{k,\kappa,\iota} > \kappa$. We have $i \in \{1, \dots E_{\kappa,\iota}\})$ and k is the phase that episode i belongs to.

Since $X_{k,\kappa,\iota} \geq \kappa + 1$ and all states in partition (κ, ι) have $w_k(s, a) \geq w_\iota$, we get

$$\mathbb{E}[\nu_i|\nu_1, \dots \nu_{i-1}] \geq (\kappa+1)w_\iota. \tag{3.6}$$

Also $\mathrm{Var}[\nu_i|\nu_1 \dots \nu_{i-1}] \leq \mathbb{E}[\nu_i|\nu_1, \dots \nu_{i-1}]H$ as $\nu_i \in [0, H]$.

To reason about $E_{\kappa,\iota}$, we define the continuation

$$\nu_i^+ := \begin{cases} \nu_i & \text{if } i \leq E_{\kappa,\iota} \\ w_\iota(\kappa+1) & \text{otherwise} \end{cases}$$

and the centered auxiliary sequence

$$\bar{\nu}_i := \frac{\nu_i^+ w_\iota(\kappa+1)}{\mathbb{E}[\nu_i^+|\nu_1^+, \dots \nu_{i-1}^+]}.$$

By construction

$$\mathbb{E}[\bar{\nu}_i|\bar{\nu}_1, \dots \bar{\nu}_{i-1}] = w_\iota(\kappa+1)\frac{\mathbb{E}[\nu_i^+|\bar{\nu}_1, \dots, \bar{\nu}_{i-1}]}{\mathbb{E}[\nu_i^+|\nu_1^+, \dots \nu_{i-1}^+]} = w_\iota(\kappa+1).$$

By Lemma 14, we have that $E_{\kappa,\iota} > \alpha N$ only if

$$\sum_{i=1}^{\alpha N} \bar{\nu}_i \leq 3N w_\iota \kappa \leq 3N w_\iota(\kappa+1).$$

Define now the martingale

$$B_i := \mathbb{E}\left[\sum_{j=1}^{\alpha N} \bar{\nu}_j|\bar{\nu}_1, \dots \bar{\nu}_i\right] = \sum_{j=1}^{i} \bar{\nu}_j + \sum_{j=i+1}^{\alpha N} \mathbb{E}[\bar{\nu}_j|\bar{\nu}_1 \dots \bar{\nu}_i]$$

which gives $B_0 = \alpha N w_\iota(\kappa+1)$ and $B_{\alpha N} = \sum_{i=1}^{\alpha N} \bar{\nu}_i$. Further, since $\nu_i^+ \in [0, H]$ and Equation (3.6), we have

$$\begin{aligned} |B_{i+1} - B_i| = |\bar{\nu}_i - \mathbb{E}[\bar{\nu}_i|\bar{\nu}_1, \dots, \bar{\nu}_{i-1}]| &= \left|\frac{w_\iota(\kappa+1)(\nu_i^+ - \mathbb{E}[\nu_i^+|\bar{\nu}_1, \dots \bar{\nu}_{i-1}])}{\mathbb{E}[\nu_i^+|\nu_1^+, \dots \nu_{i-1}^+]}\right| \\ &\leq \left|\nu_i^+ - \mathbb{E}[\nu_i^+|\bar{\nu}_1, \dots \bar{\nu}_{i-1}]\right| \leq H. \end{aligned}$$

Using

$$\begin{aligned}\sigma^2 :=& \sum_{i=1}^{\alpha N} \mathrm{Var}[B_i - B_{i-1} | B_1 - B_0, \dots B_{i-1} - B_{i-2}] \\ =& \sum_{i=1}^{\alpha N} \mathrm{Var}[\bar{\nu}_i | \bar{\nu}_1, \dots \bar{\nu}_{i-1}] \leq \alpha N H w_\iota(\kappa+1) = H B_0\end{aligned}$$

we can apply Theorem 22 by Chung and Lu (2006) and obtain

$$\begin{aligned}\mathbb{P}(E_{\kappa,\iota} > \alpha N) \leq& \mathbb{P}\left(\sum_{i=1}^{\alpha N} \bar{\nu}_i \leq 3N w_\iota(\kappa+1)\right) \\ =& \mathbb{P}(B_{\alpha N} - B_0 \leq 3B_0/\alpha - B_0) = \mathbb{P}(B_{\alpha N} - B_0 \leq -(1-3/\alpha)\, B_0) \\ \leq& \exp\left(-\frac{(3/\alpha-1)^2 B_0^2}{2\sigma^2 + H(1/3-1/\alpha)B_0}\right)\end{aligned}$$

for $\alpha \geq 3$. We can further simplify the bound to

$$\begin{aligned}\mathbb{P}(E_{\kappa,\iota} > \alpha N) \leq& \exp\left(-\frac{(3/\alpha-1)^2 B_0^2}{2HB_0 + H(1/3-1/\alpha)B_0}\right) \\ \leq& \exp\left(-\frac{(3/\alpha-1)^2}{2+(-1/\alpha+1/3)}\frac{B_0}{H}\right) \\ =& \exp\left(-\frac{\alpha(3/\alpha-1)^2}{7/3-1/\alpha}\frac{N w_\iota(\kappa+1)}{H}\right).\end{aligned}$$

□

We are now ready to prove Lemma 8 by combining the bound in the previous lemma for all κ and ι.

Proof of Lemma 8. Since $w_k(s,a) \leq H$, we have that $\frac{w_k(s,a)}{w_{\min}} < \frac{H}{w_{\min}}$ and so $\iota_k(s,a) \leq H/w_{\min} = 4H^2S/\epsilon$. In addition, $|X_{k,\kappa,\iota}| \leq S$ for all k, κ, ι and so $|X_{k,\kappa,\iota}| > \kappa$ can only be true for $\kappa \leq S$. Hence, only

$$E_{\max} = \log_2 \frac{H}{w_{\min}} \log_2 S$$

possible values for (κ, ι) exists that can have $|X_{k,\kappa,\iota}| > \kappa$. Using the union bound over all (κ, ι) and Lemma 15, we get that

$$\begin{aligned}\mathbb{P}(E \leq \alpha N E_{\max}) \geq& \mathbb{P}(\max_{(\kappa,\iota)} E_{\kappa,\iota} \leq \alpha N) \geq 1 - E_{\max} \exp\left(-\frac{\beta w_\iota(\kappa+1)N}{H}\right) \\ \geq& 1 - E_{\max} \exp\left(-\frac{\beta w_{\min} N}{H}\right) = 1 - E_{\max} \exp\left(-\frac{\beta w_{\min} m|\mathcal{S}\times\mathcal{A}|}{H}\right) \\ =& 1 - E_{\max} \exp\left(-\frac{\beta\epsilon m|\mathcal{S}\times\mathcal{A}|}{4H^2S}\right)\end{aligned}$$

Bounding the right hand-side by $1-\delta/2$ and solving for m gives

$$1 - E_{\max} \exp\left(-\frac{\beta\epsilon m|\mathcal{S}\times\mathcal{A}|}{4H^2S}\right) \geq 1-\delta/2 \quad \Leftrightarrow \quad m \geq \frac{4H^2S}{|\mathcal{S}\times\mathcal{A}|\beta\epsilon} \ln\frac{2E_{\max}}{\delta}$$

Hence, the condition

$$m \geq \frac{4H^2}{\beta\epsilon} \ln \frac{2E_{\max}}{\delta}$$

is sufficient for the desired result to hold. By plugging in $\alpha = 6$ and $\beta = \frac{\alpha(3/\alpha-1)^2}{7/3-1/\alpha} = \frac{9}{13} \geq \frac{2}{3}$, we obtain the statement to show. □

3.9.4 Bound on the value function difference for episodes with $\forall\kappa, \iota : |X_{k,\kappa,\iota}| \leq \kappa$

To prove Lemma 9, it is sufficient to consider a fixed phase k. To avoid notational clutter, we therefore omit the phase indices k in this section.

For the proof, we reason about a sequence of MDPs M_d which have the same transition probabilities but different reward functions $r^{(d)}$. For $d = 0$, the reward function is the original reward function r of M, i.e. $r_t^{(0)} = r_t$ for all $t = 1 \dots H$. The following reward functions are then defined recursively as $r_t^{(2d+2)} = \sigma_t^{(d),2}$, where $\sigma_t^{(d),2}$ is the local variance of the value function w.r.t. the rewards $r^{(d)}$. Note that for every d and $t = 1 \dots H$ and $s \in \mathcal{S}$, we have $r_t^{(d)}(s) \in [0, H^d]$. In complete analogy, we define $\tilde{M}_d$ and $\hat{M}_d$.

We first prove a sequence of lemmas necessary for Lemma 9.

Lemma 16.

$$V_h - \tilde{V}_h = \sum_{t=h}^{H-1} P_{h:t-1}(P_t - \tilde{P}_t)\tilde{V}_{t+1}$$

Proof.

$$\begin{aligned} V_h(s) - \tilde{V}_h(s) =& r(s) + P_h V_{h+1}(s) - r(s) - \tilde{P}_h\tilde{V}_{h+1}(s) + P_h\tilde{V}_{h+1}(s) - P_h\tilde{V}_{h+1}(s) \\ =& P_h(V_{h+1} - \tilde{V}_{h+1}) + (P_h - \tilde{P}_h)\tilde{V}_{h+1}(s) \end{aligned}$$

Since we have $V_H(s) = r_H(s) = \tilde{V}_H(s)$, we can recursively expand the first difference until $i = j$ and get

$$V_h - \tilde{V}_h = \sum_{t=h}^{H-1} P_{h:t-1}(P_t - \tilde{P}_t)\tilde{V}_{t+1}$$

□

Lemma 17. *Assume $p, \hat{p}, \tilde{p} \in [0,1]$ satisfy $p \in \mathcal{P}$ and $\tilde{p} \in \mathrm{conv}(\mathcal{P})$ where*

$$\begin{aligned} \mathcal{P} := \Bigg\{ p' \in [0,1] :& |\hat{p} - p'| \leq \sqrt{\frac{\ln(6/\delta_1)}{2n}}, \\ & |\hat{p} - p'| \leq \sqrt{\frac{2\hat{p}(1-\hat{p})}{n}\ln(6/\delta_1)} + \frac{7}{3(n-1)}\ln(6/\delta_1), \\ & \textit{if } n > 1 : \left|\sqrt{p'(1-p')} - \sqrt{\hat{p}(1-\hat{p})}\right| \leq \sqrt{\frac{2\ln(6/\delta_1)}{n-1}} \Bigg\}. \end{aligned}$$

Then

$$|p - \tilde{p}| \leq \sqrt{\frac{8\tilde{p}(1-\tilde{p})}{n}\ln(6/\delta_1)} + \frac{26}{3(n-1)}\ln(6/\delta_1).$$

Proof. We have $\mathcal{P} = \mathcal{P}_1 \cap \mathcal{P}_2$ with

$$\mathcal{P}_1 = \Bigg\{ p' \in [0,1] : |\hat{p} - p'| \leq \sqrt{\frac{\ln(6/\delta_1)}{2n}}, \\ |\hat{p} - p'| \leq \sqrt{\frac{2\hat{p}(1-\hat{p})}{n} \ln(6/\delta_1)} + \frac{7}{3(n-1)} \ln(6/\delta_1), \\ \text{if } n > 1 : \left(\max\left\{ 0, \sqrt{\hat{p}(1-\hat{p})} - \sqrt{\frac{2\ln(6/\delta_1)}{n-1}} \right\} \right)^2 \leq p'(1-p') \Bigg\}.$$

and

$$\mathcal{P}_2 = \left\{ p' \in \mathbb{R} : \text{if } n > 1 : \sqrt{p'(1-p')} \leq \sqrt{\hat{p}(1-\hat{p})} + \sqrt{\frac{2\ln(6/\delta_1)}{n-1}} \right\}.$$

Note that the last condition of $\mathcal{P}_1$ is equivalent to $\sqrt{\hat{p}(1-\hat{p})} \leq \sqrt{p'(1-p')} + \sqrt{\frac{2\ln(6/\delta_1)}{n-1}}$ as $p' \in [0,1]$. As an intersection of a polytope and the superlevel set of a concave function $p'(1-p')$, the set $\mathcal{P}_1$ is convex. Hence $\operatorname{conv}(\mathcal{P}) = \operatorname{conv}(\mathcal{P}_1 \cap \mathcal{P}_2) \subseteq \operatorname{conv}(\mathcal{P}_1) = \mathcal{P}_1$. It therefore follows that $\tilde{p} \in \mathcal{P}_1$. We now bound

$$\begin{aligned} |p - \tilde{p}| \leq & |p - \hat{p}| + |\hat{p} - \tilde{p}| \leq 2\sqrt{\frac{2\hat{p}(1-\hat{p})}{n} \ln(6/\delta_1)} + 2\frac{7}{3(n-1)} \ln(6/\delta_1) \\ = & \sqrt{\hat{p}(1-\hat{p})} \sqrt{\frac{8}{n} \ln(6/\delta_1)} + \frac{14}{3(n-1)} \ln(6/\delta_1) \\ \leq & \left(\sqrt{\tilde{p}(1-\tilde{p})} + \sqrt{\frac{2\ln(6/\delta_1)}{n-1}} \right) \sqrt{\frac{8}{n} \ln(6/\delta_1)} + \frac{14}{3(n-1)} \ln(6/\delta_1) \\ \leq & \sqrt{\frac{8\tilde{p}(1-\tilde{p})}{n} \ln(6/\delta_1)} + \frac{26}{3(n-1)} \ln(6/\delta_1) \end{aligned}$$

□

Lemma 18. *Assume*

$$|p(s'|s,a) - \tilde{p}_i(s'|s,a)| \leq c_1(s,a) + c_2(s,a)\sqrt{\tilde{p}_i(s'|s,a)(1 - \tilde{p}_i(s'|s,a))}$$

for $a = \pi(s,i)$ and all $s', s \in \mathcal{S}$. Then

$$|(P_i - \tilde{P}_i)\tilde{V}_{i+1}(s)| \leq c_1(s,a)|\mathcal{S}(s,a)|\|\tilde{V}_{i+1}\|_\infty + c_2(s,a)\sqrt{|\mathcal{S}(s,a)|}\tilde{\sigma}_i(s)$$

for any $(s,a) \in \mathcal{S} \times \mathcal{A}$ where $\mathcal{S}(s,a)$ denotes the set of possible successor states of state s and action a.

Proof. Let s and $a = \pi(s,i)$ be fixed and define for this fixed s the constant function $\bar{V}(s') = \tilde{P}_i\tilde{V}_{i+1}(s)$ *[sic]* as the expected value function of the successor states of s. Note that $\bar{V}(s')$ is a constant function and

so $\bar{V} = \tilde{P}_i \bar{V} = P_i \bar{V}$.

$$
\begin{aligned}
&|(P_i - \tilde{P}_i)\tilde{V}_{i+1}(s)| = |(P_i - \tilde{P}_i)\tilde{V}_{i+1}(s) + \bar{V}(s) - \bar{V}(s)| \\
=&|(P_i - \tilde{P}_i)(\tilde{V}_{i+1} - \bar{V})(s)| \\
\leq& \sum_{s' \in \mathcal{S}(s,a)} |p(s'|s,a) - \tilde{p}_i(s'|s,a)||\tilde{V}_{i+1}(s') - \bar{V}(s')| \qquad (3.7) \\
\leq& \sum_{s' \in \mathcal{S}(s,a)} \left(c_1(s,a) + c_2(s,a)\sqrt{\tilde{p}_i(s'|s,a)(1 - \tilde{p}_i(s'|s,a))} \right) |\tilde{V}_{i+1}(s') - \bar{V}(s')| \\
\leq& |\mathcal{S}(s,a)| c_1(s,a) \|\tilde{V}_{i+1}\|_\infty + c_2(s,a) \sum_{s' \in \mathcal{S}(s,a)} \sqrt{\tilde{p}_i(s'|s,a)(1 - \tilde{p}_i(s'|s,a))(\tilde{V}_{i+1}(s') - \bar{V}(s'))^2} \\
\leq& |\mathcal{S}(s,a)| c_1(s,a) \|\tilde{V}_{i+1}\|_\infty + c_2(s,a) \sqrt{|\mathcal{S}(s,a)| \sum_{s' \in \mathcal{S}(s,a)} \tilde{p}_i(s'|s,a)(1 - \tilde{p}_i(s'|s,a))(\tilde{V}_{i+1}(s') - \bar{V}(s'))^2} \qquad (3.8) \\
\leq& |\mathcal{S}(s,a)| c_1(s,a) \|\tilde{V}_{i+1}\|_\infty + c_2(s,a) \sqrt{|\mathcal{S}(s,a)| \sum_{s' \in \mathcal{S}(s,a)} \tilde{p}_i(s'|s,a)(\tilde{V}_{i+1}(s') - \bar{V}(s'))^2} \\
=& |\mathcal{S}(s,a)| c_1(s,a) \|\tilde{V}_{i+1}\|_\infty + c_2(s,a) \sqrt{|\mathcal{S}(s,a)|} \tilde{\sigma}_i(s)
\end{aligned}
$$

In Inequality (3.7), we wrote out the definition of P_i and $\tilde{P}_i$ and applied the triangle inequality. We then applied the assumed bound and bounded $|\tilde{V}_{i+1}(s') - \bar{V}(s')|$ by $\|V_{i+1}\|_\infty$ as all value functions are nonnegative. In Inequality (3.8), we applied the Cauchy-Schwarz inequality and subsequently used the fact that each term is the sum is nonnegative and that $(1 - \tilde{p}_i(s'|s,a)) \leq 1$. The final equality follows from the definition of $\tilde{\sigma}_i$. □

Bounding the difference in value function

Lemma 19. *Assume $M \in \mathcal{M}_k$. If $|X_{\kappa,\iota}| \leq \kappa$ for all (κ, ι). Then*

$$
|V_1^{(d)}(s_0) - \tilde{V}_1^{(d)}(s_0)| =: \Delta_d \leq \hat{A}_d + \hat{B}_d + \min\{\hat{C}_d, \hat{C}'_d + \hat{C}''\sqrt{\Delta_{2d+2}}\}
$$

where

$$
\hat{A}_d = \frac{\epsilon}{4} H^d, \qquad \hat{B}_d = \frac{52 H^{d+1} |\mathcal{K} \times \mathcal{I}| C}{3m} \ln \frac{6}{\delta_1},
$$

and

$$
\hat{C}'_d = \sqrt{C |\mathcal{K} \times \mathcal{I}| \frac{8}{m} H^{2d+2} \ln \frac{6}{\delta_1}} \qquad \hat{C}_d = \hat{C}'_d \sqrt{H}, \qquad \hat{C}'' = \sqrt{C |\mathcal{K} \times \mathcal{I}| \frac{8}{m} \ln \frac{6}{\delta_1}}.
$$

Proof.

$$\begin{aligned}
\Delta_d =& |V_1^{(d)}(s_0) - \tilde{V}_1^{(d)}(s_0)| = \left| \sum_{t=1}^{H-1} P_{1:t-1}(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}(s_0) \right| \\
&\leq \sum_{t=1}^{H-1} P_{1:t-1}|(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}|(s_0) \\
&= \sum_{t=1}^{H-1} P_{1:t-1} \left(\sum_{s,a \in \mathcal{S}\times\mathcal{A}} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}|(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}| \right)(s_0) \\
&= \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{t=1}^{H-1} P_{1:t-1} \left(\mathbb{I}\{s = \cdot, a = \pi(s,t)\}|(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}| \right)(s_0) \\
&= \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{t=1}^{H-1} P_{1:t-1} \left(\mathbb{I}\{s = \cdot, a = \pi(s,t)\}|(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}(s)| \right)(s_0)
\end{aligned}$$

The first equality follows from Lemma 16, the second step from the fact that $V_{t+1} \geq 0$ and $P_{1:t-1}$ being non-expansive. In the third, we introduce an indicator function which does not change the value as we sum over all (s, a) pairs. The fourth step relies on the linearity of the $P_{i:j}$ operators. In the fifth step, we realize that $\mathbb{I}\{s = \cdot, a = \pi(s,t)\}|(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}(\cdot)$ is a function that takes nonzero values only for input s. We can therefore replace the argument of the second term with s without changing the value. The term then becomes constant and by linearity of $P_{i:j}$, we can write

$$\begin{aligned}
&|V_1^{(d)}(s_0) - \tilde{V}_1^{(d)}(s_0)| = \Delta_d \leq \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{t=1}^{H-1} |(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}(s)|(P_{1:t-1}\mathbb{I}\{s = \cdot, a = \pi(s,t)\})(s_0) \\
&\leq \sum_{s,a \notin X} \sum_{t=1}^{H-1} \|\tilde{V}_{t+1}^{(d)}\|_\infty (P_{1:t-1}\mathbb{I}\{s = \cdot, a = \pi(s,t)\})(s_0) \\
&\quad + \sum_{s,a \in X} \sum_{t=1}^{H-1} |(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}(s)|(P_{1:t-1}\mathbb{I}\{s = \cdot, a = \pi(s,t)\})(s_0) \\
&\leq \sum_{s,a \notin X} \sum_{t=1}^{H-1} H^{d+1} (P_{1:t-1}\mathbb{I}\{s = \cdot, a = \pi(s,t)\})(s_0) \\
&\quad + \sum_{s,a \in X} \sum_{t=1}^{H-1} |(P_t - \tilde{P}_t)\tilde{V}_{t+1}^{(d)}(s)|(P_{1:t-1}\mathbb{I}\{s = \cdot, a = \pi(s,t)\})(s_0) \\
&\leq \sum_{s,a \notin X} \sum_{t=1}^{H-1} H^{d+1} (P_{1:t-1}\mathbb{I}\{s = \cdot, a = \pi(s,t)\})(s_0) \\
&\quad + \sum_{s,a \in X} \sum_{t=1}^{H-1} \left| |\mathcal{S}(s,a)| c_1(s,a) H^{d+1} + c_2(s,a)\sqrt{|\mathcal{S}(s,a)|}\tilde{\sigma}_t^{(d)}(s,a) \right| (P_{1:t-1}\mathbb{I}\{s = \cdot, a = \pi(s,t)\})(s_0)
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{s,a\notin X}\sum_{t=1}^{H} H^{d+1}(P_{1:t-1}\mathbb{I}\{s=\cdot,a=\pi(s,t)\})(s_0)\\
&\quad+\sum_{s,a\in X}\sum_{t=1}^{H}\left||\mathcal{S}(s,a)|c_1(s,a)H^{d+1}\right|(P_{1:t-1}\mathbb{I}\{s=\cdot,a=\pi(s,t)\})(s_0)\\
&\quad+\sum_{s,a\in X}\sum_{t=1}^{H-1}\left|c_2(s,a)\sqrt{|\mathcal{S}(s,a)|}\tilde{\sigma}_t^{(d)}(s,a)\right|(P_{1:t-1}\mathbb{I}\{s=\cdot,a=\pi(s,t)\})(s_0)\\
&\leq \sum_{s,a\notin X} H^{d+1}w(s,a)+\sum_{s,a\in X}|\mathcal{S}(s,a)|c_1(s,a)H^{d+1}w(s,a)\\
&\quad+\sum_{s,a\in X}\sqrt{|\mathcal{S}(s,a)|}c_2(s,a)\sum_{t=1}^{H-1}\tilde{\sigma}_t^{(d)}(s,a)(P_{1:t-1}\mathbb{I}\{s=\cdot,a=\pi(s,t)\})(s_0)\\
&\leq \sum_{s,a\notin X} H^{d+1}w(s,a)+\sum_{s,a\in X}Cc_1(s,a)H^{d+1}w(s,a)\\
&\quad+\sum_{s,a\in X}\sqrt{C}c_2(s,a)\sum_{t=1}^{H-1}\tilde{\sigma}_t^{(d)}(s,a)(P_{1:t-1}\mathbb{I}\{s=\cdot,a=\pi(s,t)\})(s_0)
\end{aligned}
$$

In the second inequality, we split the sum over all (s,a) pairs and used the fact that P_t and $\tilde{P}_t$ are non-expansive, i.e., $|(P_t-\tilde{P}_t)\tilde{V}_{t+1}^{(d)}(s)|\leq\|V_{t+1}^{(d)}\|_\infty$. The next step follows from $\|V_{t+1}^{(d)}\|_\infty\leq\|V_1^{(d)}\|_\infty\leq H^{d+1}$. We then apply Lemma 18 and subsequently use that all terms are nonnegative and the definition of $w(s,a)$. Eventually, we use that $|\mathcal{S}(s,a)|\leq C$ for all s,a. Using the assumption that $M\in\mathcal{M}_k$ and $\tilde{M}\in\mathcal{M}_k'$ from Lemma 12, we can apply Lemma 17 and get that

$$
c_2(s,a)=\sqrt{\frac{8}{n(s,a)}\ln\frac{6}{\delta_1}}\quad\text{and}\quad c_1(s,a)=\frac{26}{3(n(s,a)-1)}\ln\frac{6}{\delta_1}.
$$

Hence, we can bound

$$
\Delta_d\leq A(s_0)+B(s_0)+C(s_0)
$$

as a sum of three terms which we will consider individually in the following. The first term is

$$
A(s_0)=\sum_{s,a\notin X}H^{d+1}w(s,a)\leq w_{\min}SH^{d+1}\leq\frac{\epsilon H^{d+1}S}{4HS}=\frac{\epsilon}{4}H^d=\hat{A}_d
$$

as $w(s,a)\leq w_{\min}$ for all s,a not in the active set and that the policy is deterministic, which implies that there are only S nonzero w. The next term is

$$
\begin{aligned}
B(s_0)=&C\sum_{s,a\in X}w(s,a)H^{d+1}\frac{26}{3(n(s,a)-1)}\ln\frac{6}{\delta_1}\\
=&H^{d+1}C\ln\frac{6}{\delta_1}\sum_{\kappa,\iota}\sum_{s,a\in X_{\kappa,\iota}}w(s,a)\frac{26}{3(n(s,a)-1)}\\
\leq&H^{d+1}\frac{26C}{3}\ln\frac{6}{\delta_1}\sum_{\kappa,\iota}\sum_{s,a\in X_{\kappa,\iota}}\frac{w(s,a)}{n(s,a)}\frac{n(s,a)}{n(s,a)-1}.
\end{aligned}
$$

For $s, a \in X_{\kappa,\iota}$, we have $n(s,a) \geq mw(s,a)\kappa$ (see Equation (3.5)) and so

$$\frac{w(s,a)}{n(s,a)} \leq \frac{1}{\kappa m}. \tag{3.9}$$

Further, for all relevant (s,a)-pairs, we have $n(s,a) > 1$ (follows from $|X_{\kappa,\iota}| \leq \kappa$) which implies

$$B(s_0) \leq H^{d+1} \frac{52C}{3} \ln \frac{6}{\delta_1} \sum_{\kappa,\iota} \frac{|X_{\kappa,\iota}|}{\kappa m}$$

and since we assumed $|X_{\kappa,\iota}| \leq \kappa$

$$B(s_0) \leq \frac{52H^{d+1} \left|\mathcal{K} \times \mathcal{I}\right| C}{3m} \ln \frac{6}{\delta_1} = \hat{B}_d$$

where $\mathcal{K} \times \mathcal{I}$ is the set of all possible (κ, ι)-pairs. The last term is

$$\begin{aligned}
C(s_0) =& \sqrt{C} \sum_{s,a \in X} c_2(s,a) \sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d)}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\} \\
\leq& \sqrt{C} \sum_{s,a \in X} c_2(s,a) \sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d)}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\} \\
\leq& \sqrt{C} \sum_{s,a \in X} c_2(s,a) \sqrt{\sum_{t=1}^{H-1} P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}} \sqrt{\sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d),2}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}} \\
\leq& \sqrt{C} \sum_{s,a \in X} \sqrt{\frac{8w(s,a)}{n(s,a)} \ln \frac{6}{\delta_1} \sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d),2}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}}
\end{aligned}$$

where we first applied the Cauchy-Schwarz inequality and then used the definition of $c_2(s,a)$ and $w(s,a)$.

$$\begin{aligned}
C(s_0) \leq& \sqrt{C} \sum_{\kappa,\iota} \sum_{s,a \in X_{\kappa,\iota}} \sqrt{\frac{8w(s,a)}{n(s,a)} \ln \frac{6}{\delta_1} \sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d),2}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}(s_0)} \\
\leq& \sqrt{C} \sum_{\kappa,\iota} \sqrt{|X_{\kappa,\iota}| \sum_{s,a \in X_{\kappa,\iota}} \frac{8w(s,a)}{n(s,a)} \ln \frac{6}{\delta_1} \sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d),2}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}(s_0)} \\
\leq& \sqrt{C} \sum_{\kappa,\iota} \sqrt{\sum_{s,a \in X_{\kappa,\iota}} \frac{8}{m} \ln \frac{6}{\delta_1} \sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d),2}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}(s_0)} \\
\leq& \sqrt{C \left|\mathcal{K} \times \mathcal{I}\right| \frac{8}{m} \ln \frac{6}{\delta_1} \sum_{s,a \in X} \sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d),2}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}(s_0)} \\
\leq& \sqrt{C \left|\mathcal{K} \times \mathcal{I}\right| \frac{8}{m} \ln \frac{6}{\delta_1} \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{t=1}^{H-1} \tilde{\sigma}_t^{(d),2}(s,a)) P_{1:t-1} \mathbb{I}\{s = \cdot, a = \pi(s,t)\}(s_0)}
\end{aligned}$$

$$
\begin{aligned}
&= \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8}{m}\ln\frac{6}{\delta_1}\sum_{t=1}^{H-1} P_{1:t-1}\tilde{\sigma}_t^{(d),2}(s_0)} \qquad (3.10)\\
&\leq \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8H^{2d+3}\ln(6/\delta_1)}{m}} = \hat{C}_d
\end{aligned}
$$

We first split the sum and applied the Cauchy-Schwarz inequality. Then we used again Inequality (3.9) and $|X_{\kappa,\iota}| \leq \kappa$. In the fourth step, we applied Cauchy-Schwarz and the final inequality follows from $\|\tilde{\sigma}_t^{(d),2}\|_\infty \leq H^{2d+2}$ and the fact that $P_{1:t-1}$ is non-expansive. Alternatively, we can rewrite the bound in Equation (3.10) as

$$
\begin{aligned}
C(s_0) \leq& \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8}{m}\ln\frac{6}{\delta_1}\sum_{t=1}^{H-1} P_{1:t-1}\tilde{\sigma}_t^{(d),2}(s_0)}\\
=& \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8}{m}\ln\frac{6}{\delta_1}\sum_{t=1}^{H-1} P_{1:t-1}\tilde{\sigma}_t^{(d),2}(s_0) - \tilde{P}_{1:t-1}\tilde{\sigma}_t^{(d),2}(s_0) + \tilde{P}_{1:t-1}\tilde{\sigma}_t^{(d),2}(s_0)}.
\end{aligned}
$$

Lemma 10 shows that the variance $\tilde{\mathcal{V}}_1^{(d)}$ also satisfies the Bellman equation with the local variances $\tilde{\sigma}_i^{(d),2}$. This insight allows us to bound $\sum_{t=1}^{H-1}\tilde{P}_{1:t-1}\tilde{\sigma}_t^{(d),2}(s_0) = \tilde{\mathcal{V}}_1^{(d)}(s_0) \leq H^{2d+2}$. Also, note that $\tilde{\sigma}_t^{(d),2} = r_t^{(2d+2)}$ which gives us

$$
\begin{aligned}
C(s_0) \leq& \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8}{m}\ln\frac{6}{\delta_1}\left(H^{2d+2} + \sum_{t=1}^{H-1} P_{1:t-1}r_t^{(2d+2)}(s_0) - \tilde{P}_{1:t-1}r_t^{(2d+2)}(s_0)\right)}\\
=& \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8}{m}\ln\frac{6}{\delta_1}\left(H^{2d+2} + V_1^{(2d+2)}(s_0) - \tilde{V}_1^{(2d+2)}(s_0)\right)}\\
\leq& \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8}{m}\ln\frac{6}{\delta_1}\left(H^{2d+2} + \Delta_{2d+2}\right)}\\
\leq& \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8}{m}H^{2d+2}\ln\frac{6}{\delta_1}} + \sqrt{C\,|\mathcal{K}\times\mathcal{I}|\,\frac{8}{m}\Delta_{2d+2}\ln\frac{6}{\delta_1}} = \hat{C}_d' + \hat{C}''\sqrt{\Delta_{2d+2}}
\end{aligned}
$$

□

Proof of Lemma 10 (Bellman equation of local value function variances)

Proof of Lemma 10.

$$
\begin{aligned}
\mathcal{V}_i(s) =& \mathbb{E}\left[\left(\sum_{t=i}^{H} r_t(s_t) - V_i(s_i)\right)^2 |s_i = s\right]\\
=& \mathbb{E}\left[\left(\sum_{t=i+1}^{H} r_t(s_t) - V_{i+1}(s_{i+1}) + V_{i+1}(s_{i+1}) + r_i(s_i) - V_i(s_i)\right)^2 |s_i = s\right]\\
=& \mathbb{E}\left[\left(\sum_{t=i+1}^{H} r(s_t) - V_{i+1}(s_{i+1})\right)^2 |s_i = s\right]
\end{aligned}
$$

$$
\begin{aligned}
&+ 2\mathbb{E}\left[\left(\sum_{t=i+1}^{H} r_t(s_t) - V_{i+1}(s_{i+1})\right)\left(V_{i+1}(s_{i+1}) + r_i(s_i) - V(s_i)\right)|s_i = s\right] \\
&+ \mathbb{E}\left[\left(V_{i+1}(s_{i+1}) + r_i(s_i) - V_i(s_i)\right)^2 |s_i = s\right] \\
=&\mathbb{E}\left[\mathcal{V}_{i+1}(s_{i+1})|s_i = s\right] \\
&+ 2\mathbb{E}\left[\mathbb{E}\left[\left(\sum_{t=i+1}^{H} r_t(s_t) - V_{i+1}(s_{i+1})\right)\left(V_{i+1}(s_{i+1}) + r_i(s_i) - V_i(s_i)\right)|s_{i+1}\right]|s_i = s\right] \\
&+ \mathbb{E}\left[\left(V_{i+1}(s_{i+1}) - P_i V_{i+1}(s_i)\right)^2 |s_i = s\right]
\end{aligned}
$$

where the final equality follows from the tower property of conditional expectations, and the fact that $V_i(s_i) = P_i V_{i+1}(s_i) + r_i(s_i)$. Since by the definition of the value function

$$
\mathbb{E}\left[\left(\sum_{t=i+1}^{H} r_t(s_t) - V_{i+1}(s_{i+1})\right)|s_{i+1}\right] = 0
$$

the middle term vanishes and the last term is by definition $\sigma_i^2(s)$ we obtain

$$
\mathcal{V}_i(s) = P_i \mathcal{V}_{i+1}(s) + \sigma_i^2(s).
$$

Noting that $\mathcal{V}_H(s) = (r_H(s) - r_H(s))^2 = 0$, we can unroll the equation and obtain

$$
\mathcal{V}_i(s) = \sum_{t=i}^{H} P_{i:t-1}\sigma_t^2(s).
$$

From the definition of $\mathcal{V}_1$ and the fact that $0 \leq r(\cdot) \leq r_{\max}$, we see that $0 \leq \mathcal{V}_1 \leq H^2 r_{\max}^2$ and the final statement of the lemma follows.

□

Proof of Lemma 9

Proof of Lemma 9. The recursive bound from Lemma 19

$$
\Delta_d \leq \hat{A}_d + \hat{B}_d + \hat{C}'_d + \hat{C}''\sqrt{\Delta_{2d+2}}
$$

has the form $\Delta_d \leq Y_d + Z\sqrt{\Delta_{2d+2}}$. Expanding this form and using the triangle inequality gives

$$
\begin{aligned}
\Delta_0 \leq& Y_0 + Z\sqrt{\Delta_2} \leq Y_0 + Z\sqrt{Y_2 + Z\sqrt{\Delta_6}} \leq Y_0 + Z\sqrt{Y_2} + Z^{3/2}\Delta_6^{1/4} \\
\leq& Y_0 + Z\sqrt{Y_2} + Z^{3/2}Y_6^{1/4} + Z^{7/4}\Delta_{14}^{1/8} \leq \dots
\end{aligned}
$$

and by doing this up to level $\gamma = \lceil \frac{\ln H}{2 \ln 2} \rceil$, we obtain

$$
\Delta_0 \leq \sum_{d \in \mathcal{D} \setminus \{\gamma\}} Z^{\frac{2d}{2+d}} Y_d^{\frac{2}{2+d}} + Z^{\frac{2\gamma}{2+\gamma}} \Delta_\gamma^{\frac{2}{2+\gamma}}
$$

where $\mathcal{D} = \{0, 2, 6, 14, \dots \gamma\}$. Note that the exponent of H compared to m is the larger in $\hat{C}'_d$ than in $\hat{B}_d$. Therefore, for sufficiently large m, $\hat{C}'_d$ dominates the other term. More precisely, for

$$m \geq \frac{338H}{9} C \left|\mathcal{K} \times \mathcal{I}\right| \ln \frac{6}{\delta_1} \tag{3.11}$$

we have $\hat{B}_d \leq \hat{C}'_d$. We can therefore consider $Z = \hat{C}''$ and $Y_d = 2\hat{C}'_d + \hat{A}_d$. Also, since $\hat{C}_d \geq \hat{C}'_d$, we can bound $\Delta_\gamma \leq \hat{A}_d + 2\hat{C}_d$. For notational simplicity, we will use the auxiliary variable

$$m_1 = \frac{8C|\mathcal{K} \times \mathcal{I}|H^2}{m\epsilon^2} \ln \frac{6}{\delta_1}.$$

and get

$$\begin{aligned} Z = \hat{C}'' &= \sqrt{m_1}\frac{\epsilon}{H} \quad \text{and} \\ Y_d &= \hat{A}_d + 2\hat{C}'_d = (1/4 + 2\sqrt{m_1})H^d\epsilon \quad \text{and} \\ \Delta_\gamma &\leq \hat{A}_\gamma + 2\hat{C}_\gamma = (1/4 + 2\sqrt{m_1 H})H^\gamma\epsilon. \end{aligned}$$

Then

$$\left(Z^{2d}Y_d^2\right)^{(2+d)^{-1}} = \left(m_1^d\epsilon^{2d+2}(1/4 + 2\sqrt{m_1})^2\right)^{(2+d)^{-1}} = \epsilon\left(m_1^d\epsilon^d(1/4 + 2\sqrt{m_1})^2\right)^{(2+d)^{-1}}$$

and

$$\left(Z^{2\gamma}\Delta_\gamma\right)^{(2+\gamma)^{-1}} = \left(m_1^\gamma\epsilon^{2\gamma+2}(1/4 + 2\sqrt{m_1 H})^2\right)^{(2+\gamma)^{-1}} = \epsilon\left(m_1^\gamma\epsilon^\gamma(1/4 + 2\sqrt{m_1 H})^2\right)^{(2+\gamma)^{-1}}.$$

Putting these pieces together, we obtain

$$\begin{aligned} \frac{\Delta_0}{\epsilon} \leq& \sum_{d\in\mathcal{D}\setminus\{\gamma\}} (\epsilon m_1)^{\frac{d}{2+d}}\left(\frac{1}{4} + 2\sqrt{m_1}\right)^{\frac{2}{d+2}} + (\epsilon m_1)^{\frac{\gamma}{\gamma+2}}\left(\frac{1}{4} + 2\sqrt{Hm_1}\right)^{\frac{2}{\gamma+2}} \\ =& \frac{1}{4} + 2\sqrt{m_1} + \sum_{d\in\mathcal{D}\setminus\{0,\gamma\}} (\epsilon m_1)^{\frac{d}{2+d}}\left(\frac{1}{4} + 2\sqrt{m_1}\right)^{\frac{2}{d+2}} + (\epsilon m_1)^{\frac{\gamma}{\gamma+2}}\left(\frac{1}{4} + 2\sqrt{Hm_1}\right)^{\frac{2}{\gamma+2}} \\ \leq& \frac{1}{4} + 2\sqrt{m_1} + \sum_{d\in\mathcal{D}\setminus\{0,\gamma\}} (\epsilon m_1)^{\frac{d}{2+d}}\left[\left(\frac{1}{4}\right)^{\frac{2}{d+2}} + (2\sqrt{m_1})^{\frac{2}{d+2}}\right] \\ &+ (\epsilon m_1)^{\frac{\gamma}{\gamma+2}}\left[\left(\frac{1}{4}\right)^{\frac{2}{\gamma+2}} + \left(2\sqrt{Hm_1}\right)^{\frac{2}{\gamma+2}}\right] \end{aligned}$$

where we used the fact that $(a+b)^\phi \leq a^\phi + b^\phi$ for $a, b > 0$ and $0 < \phi < 1$. We now bound the $H^{1/(2+\gamma)}$ by using the definition of γ. Since

$$\frac{1}{2+\gamma} = \frac{2\ln 2}{4\ln 2 + \ln H} \leq 2\log_H 2$$

and since $H \geq 1$, we have $H^{1/(2+\gamma)} \leq 4$. Therefore

$$\begin{aligned}
\frac{\Delta_0}{\epsilon} \leq & \frac{1}{4} + 2\sqrt{m_1} + \sum_{d \in \mathcal{D}\setminus\{0,\gamma\}} (\epsilon m_1)^{\frac{d}{2+d}} \left[\left(\frac{1}{4}\right)^{\frac{2}{d+2}} + (2\sqrt{m_1})^{\frac{2}{d+2}} \right] \\
& + (\epsilon m_1)^{\frac{\gamma}{\gamma+2}} \left[\left(\frac{1}{4}\right)^{\frac{2}{\gamma+2}} + 4\,(2\sqrt{m_1})^{\frac{2}{\gamma+2}} \right] \\
\leq & \frac{1}{4} + 2\sqrt{m_1} + \sum_{d \in \mathcal{D}\setminus\{0\}} (\epsilon m_1)^{\frac{d}{2+d}} \left[\left(\frac{1}{4}\right)^{\frac{2}{d+2}} + 4\,(2\sqrt{m_1})^{\frac{2}{d+2}} \right] \\
\leq & \frac{1}{4} + 2\sqrt{m_1} + \sum_{i=1}^{\log_2 \gamma} (\epsilon m_1)^{1-2^{-i}} \left[\left(\frac{1}{4}\right)^{2^{-i}} + 4\,(2\sqrt{m_1})^{2^{-i}} \right] \\
\leq & \frac{1}{4} + 2\sqrt{m_1} + \sum_{i=1}^{\log_2 \gamma} m_1^{1-2^{-i}} \left[\left(\frac{1}{4}\right)^{2^{-i}} + 4\,(2\sqrt{m_1})^{2^{-i}} \right]
\end{aligned}$$

In the first inequality, we used the bound for $H^{1/(2+\gamma)}$ and in the second inequality we simplified the expression by noting that all terms are nonnegative. In the next step, we re-parameterized the sum. In the final inequality, we used the assumption that $0 < \epsilon \leq 1$ and therefore $\epsilon^{1-2^{-i}} \leq 1$.

$$\begin{aligned}
\frac{\Delta_0}{\epsilon} \leq & \frac{1}{4} + 2\sqrt{m_1} + \frac{1}{4} \sum_{i=1}^{\log_2 \gamma} (4m_1)^{1-2^{-i}} + 4 \sum_{i=1}^{\log_2 \gamma} (m_1)^{1-2^{-i}} (4m_1)^{2^{-i-1}} \\
\leq & \frac{1}{4} + 2\sqrt{m_1} + \frac{1}{4} \sum_{i=1}^{\log_2 \gamma} (4m_1)^{1-2^{-i}} + 16 \sum_{i=1}^{\log_2 \gamma} \left(\frac{m_1}{4}\right)^{1-2^{-i-1}}.
\end{aligned}$$

By requiring that

$$m_1 \leq \frac{1}{4}$$

and noting that $1 - 2^{-i} \geq 1/2$ and $1 - 2^{-i-1} \geq 3/4$ for $i \geq 1$, we can bound the expression by

$$\frac{\Delta_0}{\epsilon} \leq \frac{1}{4} + 2\sqrt{m_1} + \frac{1}{4} \log_2(\gamma) \sqrt{4m_1} + 16 \log_2(\gamma) \left(\frac{m_1}{4}\right)^{3/4}.$$

By requiring that $m_1 \leq 1/64$ and $m_1 \leq (2 \log_2 \gamma)^{-2}$ and $m_1 \leq 1/64 (\log_2 \gamma)^{-4/3}$, we can assure that $\Delta_0 \leq \epsilon$. Taking all assumptions on m_1 we made above together, we realize that

$$m_1 \leq \left(\frac{1}{8 \log_2 \log_2 H}\right)^2 \leq \left(\frac{1}{8 \log_2 \gamma}\right)^2$$

is sufficient for them to hold where we used $\log_2 \gamma = \log_2(\lceil \frac{1}{2} \log_2 H \rceil) \leq \log_2 \log_2 H$. This gives the following condition on m

$$m \geq 512 C (\log_2 \log_2 H)^2 |\mathcal{K} \times \mathcal{I}| \frac{H^2}{\epsilon^2} \ln \frac{6}{\delta_1}$$

which is a stronger condition that the one in Equation (3.11).

By construction of $\iota(s,a)$, we have $\iota(s,a) \leq 2\frac{H}{w_{\min}} = \frac{8SH^2}{\epsilon} = \frac{8H^2S}{\epsilon}$. Also, $\kappa_k(s,a) \leq \frac{SmH}{mw_{\min}} = \frac{4S^2H^2}{\epsilon}$. Therefore

$$|\mathcal{K} \times \mathcal{I}| \leq \log_2 \frac{4S^2H^2}{\epsilon} \log_2 \frac{8H^2S}{\epsilon} \leq \log_2^2 \frac{8H^2S^2}{\epsilon}$$

which let us conclude that

$$m \geq 512 \frac{CH^2}{\epsilon^2} (\log_2 \log_2 H)^2 \log_2^2 \left(\frac{8H^2S^2}{\epsilon}\right) \ln \frac{6}{\delta_1}$$

is a sufficient condition and thus, the statement to show, holds. □

3.9.5 Proof of Theorem 6

Proof of Theorem 6. By Lemma 8, we know that the number of episodes where $|X_{\kappa,\iota}| > \kappa$ for some κ, ι is bounded by $6E_{\max}|\mathcal{S} \times \mathcal{A}|m$ with probability at least $1 - \delta/2$. For all other episodes, we have by Lemma 9 that $|\rho_{\tilde{M}}(\pi_k) - \rho(\pi_k)| < \epsilon$. Since, with probability at least $1 - \delta/2$, we have by Lemma 7 $M \in \mathcal{M}_k$, we can use Lemma 12 which gives $\rho_{\tilde{M}}(\pi_k) > \rho^\star \geq \rho(\pi_k)$ to conclude that with probabilty at least $1 - \delta/2$, for all episodes with $|X_{\kappa,\iota}| \leq \kappa$ for all κ, ι, we have $\rho^\star - \rho(\pi_k) < \epsilon$. Applying the union bound, we get the desired result, if m satisfies

$$\begin{aligned} m \geq & 512 \frac{CH^2}{\epsilon^2} (\log_2 \log_2 H)^2 \log_2^2 \left(\frac{8H^2S^2}{\epsilon}\right) \ln \frac{6}{\delta_1} \quad \text{and} \\ m \geq & \frac{6H^2}{\epsilon} \ln \frac{2E_{\max}}{\delta}. \end{aligned}$$

From the definitions, we get

$$\ln \frac{6}{\delta_1} = \ln \frac{6CU_{\max}}{\delta} = \ln \frac{6|\mathcal{S} \times \mathcal{A}|C \log_2(SH/w_{\min})}{\delta} = \ln \frac{6|\mathcal{S} \times \mathcal{A}|C \log_2(4S^2H^2/\epsilon)}{\delta}$$

and

$$E_{\max} = \log_2 S \log_2 \frac{4H^2S}{\epsilon} \leq \log_2^2 \frac{4H^2S}{\epsilon}$$

and

$$\begin{aligned} \ln \frac{2E_{\max}}{\delta} = & \ln \frac{2 \log_2 S \log_2(4H^2S/\epsilon)}{\delta} \leq \ln \frac{2 \log_2^2(4H^2S/\epsilon)}{\delta} \\ \leq & \ln \frac{6 \, |\mathcal{S} \times \mathcal{A}| \log_2^2(4S^2H^2/\epsilon)}{\delta}. \end{aligned}$$

Setting

$$m = 512 (\log_2 \log_2 H)^2 \frac{CH^2}{\epsilon^2} \log^2 \left(\frac{8H^2S^2}{\epsilon}\right) \ln \frac{6 \, |\mathcal{S} \times \mathcal{A}| \, C \log_2^2(4S^2H^2/\epsilon)}{\delta}$$

is therefore a valid choice for m to ensure that with probability at least $1 - \delta$, there are at most

$$\begin{aligned} 6mE_{\max} = & 3072 (\log_2 \log_2 H)^2 \frac{CH^2 \, |\mathcal{S} \times \mathcal{A}|}{\epsilon^2} \\ & \times \log_2^2 \left(\frac{4H^2S}{\epsilon}\right) \log^2 \left(\frac{8H^2S^2}{\epsilon}\right) \ln \frac{6|\mathcal{S} \times \mathcal{A}|C \log_2^2(4S^2H^2/\epsilon)}{\delta} \end{aligned}$$

ϵ-suboptimal episodes.

□

3.10 Proof of the Lower PAC Bound

Proof of Theorem 11. We consider the class of MDPs shown in Figure 3.1. The MDPs essentially consist of n parallel multi-armed bandits. For each bandit, there exist $m+1 = A$ possible instantiations, which we denote by $I_i = 0 \dots m$. The instantiation, or *hypothesis*, $I_i = 0$ corresponds to $\epsilon_i(a) = \mathbb{I}\{a = a_0\}\epsilon'/2$, that is, only action a_0 has a small bias. The other hypotheses $I_i = j$ for $j = 1 \dots m$ correspond to $\epsilon_i(a) = \mathbb{I}\{a = a_0\}\epsilon'/2 + \mathbb{I}\{a = a_j\}\epsilon'$. We use $I = (I_1, \dots I_n)$ to indicate the instance of the entire MDP.

We define $G_i = \{\omega \in \Omega : \pi(i) = a_{I_i}\}$, the event that π, the policy generated by A chooses optimally in bandit i. For a given instance I, the difference between the optimal expected cumulative reward $\rho_I^\star$ and the expected cumulative reward $\rho_I(\pi)$ of policy π is at least

$$\rho_I^\star - \rho_I(\pi) \geq (H-2)\left(1 - \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{G_i\}\right)\frac{\epsilon'}{2}.$$

For π to be ϵ-optimal, we therefore need

$$\begin{aligned}
\epsilon \geq& \rho_I^\star - \rho_I(\pi) \geq (H-2)\left(1 - \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{G_i\}\right)\frac{\epsilon'}{2}, \\
\frac{2\epsilon}{(H-2)\epsilon'} \geq& \left(1 - \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{G_i\}\right), \\
\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{G_i\} \geq& \left(1 - \frac{2\epsilon}{(H-2)\epsilon'}\right), \\
\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{G_i\} \geq& \left(1 - \frac{2\epsilon(H-2)\eta}{(H-2)16\epsilon e^4}\right) = 1 - \frac{\eta}{8e^4}
\end{aligned}$$

where we chose value $\epsilon' := \frac{16\epsilon e^4}{(H-2)\eta}$ for ϵ'. We will specify the exact value of parameter η later. The condition basically states that at least a fraction of $\phi := 1 - \eta/(8e^4)$ bandits need to be solved optimally by A for the resulting policy π to be ϵ-accurate. For A to be (ϵ, δ)-correct, we therefore need

$$\mathbb{P}_I\left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{G_i\} \geq \phi\right) \geq \mathbb{P}_I(\rho_I^\star - \rho_I(\pi) \geq \epsilon) \geq 1 - \delta$$

for each instance I. Using Markov's inequality, we obtain

$$1 - \delta \leq \mathbb{P}_I\left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{G_i\} \geq \phi\right) \leq \frac{1}{n\phi}\sum_{i=1}^{n} \mathbb{E}_I[\mathbb{I}\{G_i\}] \leq \frac{1}{n\phi}\sum_{i=1}^{n} \mathbb{P}_I(G_i)$$

All G_i are independent of each other by construction of the MDP. In fact $\sum_{i=1}^{n} \mathbb{I}\{G_i\}$ is Poisson-binomial distributed as $\mathbb{I}\{G_i\}$ are independent Bernoulli random variables with potentially different mean. Therefore, upper bounds δ_i must exist such that $\delta_i \geq P_I(G_i^C)$ for all hypotheses I and such that $1 - \delta \leq \frac{1}{n\phi}\sum_{i=1}^{n}(1 - \delta_i)$ or equivalently $n(1 + \delta\phi - \phi) \geq \sum_{i=1}^{n} \delta_i$. Since all G_i are independent of each other and

$$\epsilon' = \frac{16\epsilon e^4}{(H-2)\eta} \leq \frac{16(H-2)e^4\eta}{(H-2)64e^4\eta} = \frac{1}{4}$$

we can apply Theorem 1 by Mannor and Tsitsiklis (2004) in cases where

$$\delta_i \leq \frac{1}{\eta}(1-\phi+\delta\phi) \leq \frac{1}{\eta}(1-\phi+\delta) \leq \frac{1}{8e^4} + \frac{\delta}{\eta} \leq \frac{2}{8e^4}.$$

This result gives us the minimum expected number of times $\mathbb{E}_I[n_i]$ we need to observe state i to ensure that $P_I(G_i^C) \leq \delta_i$

$$\mathbb{E}_I[n_i] \geq \left[\frac{c_1(A-1)}{\epsilon'^2} \ln\left(\frac{c_2}{\delta_i}\right)\right] \mathbb{I}\{\eta\delta_i \leq 1-\phi+\phi\delta\},$$

for appropriate constants c_1 and c_2 (e.g. $c_1 = 400$ and $c_2 = 4$). We can find a valid lower bound for the total number of samples for any $\delta_1, \dots \delta_n$ by considering the worst bound over all $\delta_1, \dots \delta_n$. The following optimization problem encodes this idea

$$\begin{aligned} \min_{\delta_1,\dots\delta_n} & \sum_{i=1}^{n} \ln\frac{1}{\delta_i} \mathbb{I}\{\eta\delta_i \leq 1-\phi+\phi\delta\} \\ \text{s.t.} & \sum_{i=1}^{n} \delta_i \leq n(1+\phi\delta-\phi) \end{aligned} \tag{3.12}$$

As shown in Lemma 20 in the supplementary material, the optimal solution of the optimization problem in Equation (3.12) is $\delta_1 = \dots = \delta_n = c$ if $\eta(1-\ln c) \leq 1$ with $c = 1+\delta\phi-\phi$. Since the left-hand side of this condition is decreasing in c, we can plug in a lower bound of $c \geq 1-\phi = \frac{\eta}{8e^4}$ and get the sufficient condition

$$\eta(1-\ln\frac{\eta}{8e^4}) = \eta(1-\ln\eta+4+\ln 8) \leq 1.$$

It is easy to verify that $\eta = 1/10$ satisfies this condition. Hence $\delta_1 = \dots = \delta_n = c$ is the optimal solution to the problem in Equation (3.12). In each episode, we only observe a single state i and therefore, there need to be at least

$$\mathbb{E}_I[n_A] \geq \sum_{i=1}^{n} \mathbb{E}_I[n_i] \geq \frac{c_1(A-1)n}{\epsilon'^2} \ln\left(\frac{c_2}{\delta_i}\right) \geq \frac{c_1(A-1)n}{\epsilon'^2} \ln\left(\frac{c_2}{\delta+\frac{\eta}{8e^4}}\right)$$

observed episodes for appropriate constants c_1 and c_2. Plugging in ϵ' and $n = S-3$, we obtain the desired statement.

□

Lemma 20. *The optimization problem*

$$\begin{aligned} \min_{\delta_1\dots\delta_n\in[0,1]} & \sum_{i=1}^{n} \ln\frac{1}{\delta_i} \mathbb{I}\{\eta\delta_i \leq c\} \\ \textit{s.t.} & \sum_{i=1}^{n} \delta_i \leq nc \end{aligned}$$

with $c \in [0,1]$ *and*

$$\eta(1-\ln c) \leq 1$$

has optimal solution $\delta_1 = \dots = \delta_n = c$.

Proof. Without the indicator part in the objective, we can show that $\delta_1 = \cdots = \delta_n = c$ is an optimal solution by checking the KKT conditions and noting that the problem is convex. Let k denote the number of δ_j that are set such that the indicator function is 0. Without loss of generality we can assume that their value is $\delta_P := c/\eta$ and the remaining δ_j take the same value δ_A (for a fixed δ_P and k, the problem reduces to the one without the indicator functions). Then the problem transforms into

$$\min_{\delta_A \in (0,1), k \in \{0,1,\dots n\}} (n-k) \ln \frac{1}{\delta_A}$$
$$(n-k)\delta_A + k\delta_P \leq nc$$

We can rewrite the constraint as

$$\begin{aligned}(n-k)\delta_A + k\delta_P &\leq nc \\ (n-k)\delta_A &\leq nc - k\delta_P = \left(n - \frac{k}{\eta}\right) c \\ \delta_A &\leq \frac{n - \frac{k}{\eta}}{n-k} c.\end{aligned}$$

Since the objective decreases with δ_A, it is optimal to choose δ_A as large as possible. The optimization problem then reduces to

$$\min_{k \in \{0,\dots \lfloor n/\gamma \rfloor\}} (n-k) \ln \left(\frac{n-k}{n-\gamma k} c^{-1} \right).$$

where we used for convenience $\gamma := 1/\eta$. We want to show that the optimal solution to this problem is $k = 0$. We can therefore relax the problem to the continuous domain without loss of generality

$$\min_{k \in [0, n/\gamma]} (n-k) \ln \left(\frac{n-k}{n-\gamma k} c^{-1} \right).$$

By reparameterizing the problem with $\alpha = k/n$, we get

$$\min_{\alpha \in [0, 1/\gamma]} n(1-\alpha) \ln \left(\frac{1-\alpha}{c(1-\gamma\alpha)} \right).$$

We realize that the minimizer does not depend on n (while the value does). The second derivative of the objective function is

$$n \frac{(\gamma - 1)^2}{(1-\gamma\alpha)^2 (1-\alpha)},$$

which is nonnegative for $\alpha \in [0, 1/\gamma]$. Hence, the objective is convex in the feasible region and the minimizer of this problem is $\alpha = 0$ if the derivative of the objective is nonnegative in 0. The derivative of the objective in 0 is given by

$$n(\gamma - 1 + \ln(c)).$$

A sufficient condition for $\alpha = 0$ being optimal is therefore

$$\gamma \geq 1 - \ln c$$

or, in terms of the original problem with $\eta = 1/\gamma$, $\delta_1 = \dots \delta_n = c$ is optimal if

$$\eta(1 - \ln c) \leq 1$$

□

Chapter 4

Unifying PAC and Regret: Uniform-PAC Bounds for Episodic Reinforcement Learning

This chapter is based on the work published as:
Christoph Dann, Tor Lattimore, and Emma Brunskill. "Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5713–5723

4.1 Introduction

The recent empirical successes of deep reinforcement learning (RL) are tremendously exciting, but the performance of these approaches still varies significantly across domains, each of which requires the user to solve a new tuning problem (François-Lavet, Fonteneau, and Ernst, 2015). Ultimately we would like reinforcement learning algorithms that simultaneously perform well empirically and have strong theoretical guarantees. Such algorithms are especially important for high stakes domains like health care, education and customer service, where non-expert users demand excellent outcomes.

In this chapter, we propose a new framework for measuring the performance of reinforcement learning algorithms called Uniform-PAC. Briefly, an algorithm is Uniform-PAC if with high probability it simultaneously for all $\epsilon > 0$ selects an ϵ-optimal policy on all episodes except for a number that scales polynomially with $1/\epsilon$. Algorithms that are Uniform-PAC converge to an optimal policy with high probability and immediately yield both PAC and high probability regret bounds, which makes them superior to algorithms that come with only PAC or regret guarantees. Indeed,

(a) Neither PAC nor regret guarantees imply convergence to optimal policies with high probability;

(b) (ϵ, δ)-PAC algorithms may be $\epsilon/2$-suboptimal in every episode;

(c) Algorithms with small regret may be maximally suboptimal infinitely often.

Uniform-PAC algorithms suffer none of these drawbacks. One could hope that existing algorithms with PAC or regret guarantees might be Uniform-PAC already, with only the analysis missing. Unfortunately this is not the case and modification is required to adapt these approaches to satisfy the new performance metric. The key insight for obtaining Uniform-PAC guarantees is to leverage time-uniform concentration bounds such as the finite-time versions of the law of iterated logarithm, which obviates the need for horizon-dependent confidence levels.

We provide a new optimistic algorithm for episodic RL called UBEV that is Uniform PAC. Unlike its predecessors, UBEV uses confidence intervals based on the law of iterated logarithm (LIL) which hold uniformly over time. They allow us to more tightly control the probability of failure events in which the algorithm behaves poorly. Our analysis is nearly optimal according to the traditional metrics, with a linear dependence on the state space for the PAC setting and square root dependence for the regret. Therefore UBEV is a Uniform PAC algorithm with PAC bounds and high probability regret bounds that are near optimal in the dependence on the length of the episodes (horizon) and optimal in the state and action spaces cardinality as well as the number of episodes. To our knowledge UBEV is the first algorithm with both near-optimal PAC and regret guarantees.

Notation and setup. We consider episodic fixed-horizon MDPs as introduced in Chapter 2 but with one minor difference. In this chapter, we allow dynamics to be time-dependent. This can be formalized as a tuple $M = (\mathcal{S}, \mathcal{A}, p_R, P, p_0, H)$. The state space $\mathcal{S}$ and the action space $\mathcal{A}$ are finite sets with cardinality S and A. The agent interacts with the MDP in episodes of H time steps each. At the beginning of each time-step $t \in [H]$ the agent observes a state s_t and chooses an action a_t based on a policy π that may depend on the within-episode time step ($a_t = \pi(s_t, t)$). The next state is sampled from the tth transition kernel $s_{t+1} \sim P(\cdot|s_t, a_t, t)$ and the initial state from $s_1 \sim p_0$. The agent then receives a reward drawn from a distribution $p_R(s_t, a_t, t)$ which can depend on s_t, a_t and t with mean $r(s_t, a_t, t)$ determined by the reward function. The reward distribution p_R is supported on $[0, 1]$.[1] The value function from time step t for policy π is defined as

$$V_t^\pi(s) := \mathbb{E}\left[\sum_{i=t}^{H} r(s_i, a_i, i)\middle| s_t = s\right] = \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s,t), t) V_{t+1}^\pi(s') + r(s, \pi(s,t), t)\,.$$

and the optimal value function is denoted by $V_t^\star$. As a reminder, in any fixed episode, the quality of a policy π is evaluated by the *total expected return*

$$\rho(\pi) := \mathbb{E}\left[\sum_{i=t}^{H} r(s_i, a_i, i)\middle|\pi\right] = p_0^\top V_1^\pi\,,$$

which is compared to the *optimal return* $\rho^\star = p_0^\top V_1^\star$. For this notation p_0 and the value functions $V_t^\star$, V_1^π are interpreted as vectors of length S. If an algorithm follows policy π_k in episode k, then the optimality gap in episode k is $\Delta_k := \rho^\star - \rho(\pi_k)$ which is bounded by $\Delta_{\max} = \max_\pi \rho^\star - \rho(\pi) \le H$. We let $N_\epsilon := \sum_{k=1}^{\infty} \mathbb{I}\{\Delta_k > \epsilon\}$ be the number of ϵ-errors and $R(T)$ be the regret after T episodes: $R(T) := \sum_{k=1}^{T} \Delta_k$. Note that T is the number of episodes and not total time steps (which is HT after T episodes) and k is an episode index while t usually denotes time indices within an episode. The $\tilde{O}$ notation is similar to the usual O-notation but suppresses additional polylog-factors, that is $g(x) = \tilde{O}(f(x))$ iff there is a polynomial p such that $g(x) = O(f(x)p(\log(x)))$.

4.2 Uniform PAC and Existing Learning Frameworks

We briefly summarize the most common performance measures used in the literature.

[1] The reward may be allowed to depend on the next-state with no further effort in the proofs. The boundedness assumption could be replaced by the assumption of subgaussian noise with known subgaussian parameter.

- (ϵ, δ)-*PAC (mistake-style):* There exists a polynomial function $F_{\text{PAC}}(S, A, H, 1/\epsilon, \log(1/\delta))$ such that

$$\mathbb{P}\left(N_\epsilon > F_{\text{PAC}}(S, A, H, 1/\epsilon, \log(1/\delta))\right) \leq \delta\,.$$

- *Expected Regret:* There exists a function $F_{\text{ER}}(S, A, H, T)$ such that $\mathbb{E}[R(T)] \leq F_{\text{ER}}(S, A, H, T)$.
- *High Probability Regret:* There exists a function $F_{\text{HPR}}(S, A, H, T, \log(1/\delta))$ such that

$$\mathbb{P}\left(R(T) > F_{\text{HPR}}(S, A, H, T, \log(1/\delta))\right) \leq \delta\,.$$

- *Uniform High Probability Regret:* There exists a function $F_{\text{UHPR}}(S, A, H, T, \log(1/\delta))$ such that

$$\mathbb{P}\left(\text{exists } T : R(T) > F_{\text{UHPR}}(S, A, H, T, \log(1/\delta))\right) \leq \delta\,.$$

In all definitions the function F should be polynomial in all arguments. For notational conciseness we often omit some of the parameters of F where the context is clear. The different performance guarantees are widely used (e.g. PAC: (Lattimore and Hutter, 2012; Dann and Brunskill, 2015; Jiang, Krishnamurthy, et al., 2017; Strehl and Littman, 2008), (uniform) high-probability regret: (Jaksch, Ortner, and Auer, 2010; Agarwal, Hsu, et al., 2014; Srinivas et al., 2010); expected regret: (Audibert, Munos, and Szepesvári, 2009; Auer, 2000; Bubeck and Cesa-Bianchi, 2012; Auer and Ortner, 2005)). Due to space constraints, we will not discuss Bayesian-style performance guarantees that only hold in expectation with respect to a distribution over problem instances. We will shortly discuss the limitations of the frameworks listed above, but first formally define the Uniform-PAC criteria

Definition 1 (Uniform-PAC)**.** *An algorithm is Uniform-PAC for $\delta > 0$ if*

$$\mathbb{P}\left(\textit{exists } \epsilon > 0 :\ N_\epsilon > F_{\textit{UPAC}}\left(S, A, H, 1/\epsilon, \log(1/\delta)\right)\right) \leq \delta\,,$$

where $F_{\textit{UPAC}}$ is polynomial in all arguments.

All the performance metrics are functions of the distribution of the sequence of errors over the episodes $(\Delta_k)_{k\in\mathbb{N}}$. Regret bounds are the integral of this sequence up to time T, which is a random variable. The expected regret is just the expectation of the integral, while the high-probability regret is a quantile. PAC bounds are the quantile of the size of the superlevel set for a fixed level ϵ. Uniform-PAC bounds are like PAC bounds, but hold for all ϵ simultaneously.

Limitations of regret. Since regret guarantees only bound the integral of Δ_k over k, it does not distinguish between making a few severe mistakes and many small mistakes. In fact, since regret bounds provably grow with the number of episodes T, an algorithm that achieves optimal regret may still make infinitely many mistakes (of arbitrary quality, see proof of Theorem 22 below). This is highly undesirable in high-stakes scenarios. For example in drug treatment optimization in healthcare, we would like to distinguish between infrequent severe complications (few large Δ_k) and frequent minor side effects (many small Δ_k). In fact, even with an optimal regret bound, we could still serve infinitely patients with the worst possible treatment.

Limitations of PAC. PAC bounds limit the number of mistakes for a given accuracy level ϵ, but is otherwise non-restrictive. That means an algorithm with $\Delta_k > \epsilon/2$ for all k almost surely might still be (ϵ, δ)-PAC. Worse, many algorithms designed to be (ϵ, δ)-PAC actually exhibit this behavior because they explicitly halt learning once an ϵ-optimal policy has been found. The less widely used TCE (total cost of exploration) bounds (Pazis and Parr, 2016) and KWIK guarantees (Li, Littman, and Walsh, 2008) suffer from the same issueand for conciseness are not discussed in detail.

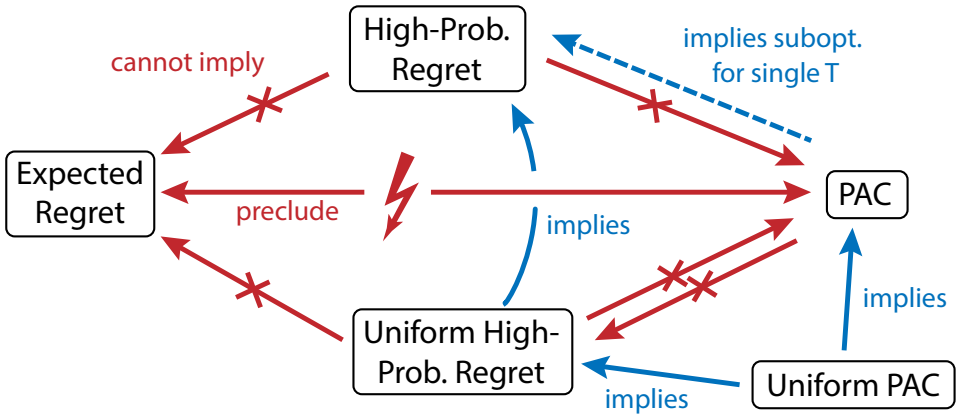

Figure 4.1: Visual summary of relationship among the different learning frameworks: Expected regret (ER) and PAC preclude each other while the other crossed arrows represent only a *does-not-implies* relationship. Blue arrows represent *imply* relationships. For details see the theorem statements.

Advantages of Uniform-PAC. The new criterion overcomes the limitations of PAC and regret guarantees by measuring the number of ϵ-errors at every level simultaneously. By definition, algorithms that are Uniform-PAC for a δ are (ϵ, δ)-PAC for all $\epsilon > 0$. We will soon see that an algorithm with a non-trivial Uniform-PAC guarantee also has small regret with high probability. Furthermore, there is no loss in the reduction so that an algorithm with optimal Uniform-PAC guarantees also has optimal regret, at least in the episodic RL setting. In this sense Uniform-PAC is the missing bridge between regret and PAC. Finally, for algorithms based on confidence bounds, Uniform-PAC guarantees are usually obtained without much additional work by replacing standard concentration bounds with versions that hold uniformly over episodes (e.g. using the law of the iterated logarithms). In this sense we think Uniform-PAC is the new 'gold-standard' of theoretical guarantees for RL algorithms.

4.2.1 Relationships between Performance Guarantees

Existing theoretical analyses usually focus exclusively on either the regret or PAC framework. Besides occasional heuristic translations, Proposition 4 in (Strehl and Littman, 2008) and Corollary 3 in (Jaksch, Ortner, and Auer, 2010) are the only results relating a notion of PAC and regret, we are aware of. Yet the guarantees there are not widely used[2] unlike the definitions given above which we now formally relate to each other. A simplified overview of the relations discussed below is shown in Figure 4.1.

Theorem 21. *No algorithm can achieve*

- *a sub-linear expected regret bound for all T and*
- *a finite (ϵ, δ)-PAC bound for a small enough ϵ*

simultaneously for all two-armed multi-armed bandits with Bernoulli reward distributions. This implies that such guarantees also cannot be satisfied simultaneously for all episodic MDPs.

A full proof is in Section 4.6.1, but the intuition is simple. Suppose a two-armed Bernoulli bandit has mean rewards $1/2 + \epsilon$ and $1/2$ respectively and the second arm is chosen at most $F < \infty$ times with probability at least $1 - \delta$, then one can easily show that in an alternative bandit with mean rewards $1/2 + \epsilon$

[2]The average per-step regret in (Jaksch, Ortner, and Auer, 2010) is superficially a PAC bound, but does not hold over infinitely many time-steps and exhibits the limitations of a conventional regret bound. The translation to average loss in (Strehl and Littman, 2008) comes at additional costs due to the discounted infinite horizon setting.

and $1/2 + 2\epsilon$ there is a non-zero probability that the second arm is played finitely often and in this bandit the expected regret will be linear. Therefore, sub-linear expected regret is only possible if each arm is pulled infinitely often almost surely.

Theorem 22. *The following statements hold for performance guarantees in episodic MDPs:*

(a) If an algorithm satisfies a (ϵ, δ)-PAC bound with $F_{PAC} = \Theta(1/\epsilon^2)$ then it satisfies for a specific $T = \Theta(\epsilon^{-3})$ a $F_{HPR} = \Theta(T^{2/3})$ bound. Further, there is an MDP and algorithm that satisfies the (ϵ, δ)-PAC bound $F_{PAC} = \Theta(1/\epsilon^2)$ on that MDP and has regret $R(T) = \Omega(T^{2/3})$ on that MDP for any T. That means a (ϵ, δ)-PAC bound with $F_{PAC} = \Theta(1/\epsilon^2)$ can only be converted to a high-probability regret bound with $F_{HPR} = \Omega(T^{2/3})$.

(b) For any chosen $\epsilon, \delta > 0$ and F_{PAC}, there is an MDP and algorithm that satisfies the (ϵ, δ)-PAC bound F_{PAC} on that MDP and has regret $R(T) = \Omega(T)$ on that MDP. That means a (ϵ, δ)-PAC bound cannot be converted to a sub-linear uniform high-probability regret bound.

(c) For any $F_{UHPR}(T, \delta)$ with $F_{UHPR}(T, \delta) \to \infty$ as $T \to \infty$, there is an algorithm that satisfies that uniform high-probability regret bound on some MDP but makes infinitely many mistakes for any sufficiently small accuracy level $\epsilon > 0$ for that MDP. Therefore, a high-probability regret bound (uniform or not) cannot be converted to a finite (ϵ, δ)-PAC bound.

(d) For any $F_{UHPR}(T, \delta)$ there is an algorithm that satisfies that uniform high-probability regret bound on some MDP but suffers expected regret $\mathbb{E}R(T) = \Omega(T)$ on that MDP.

For most interesting RL problems including episodic MDPs the worst-case expected regret grows with $O(\sqrt{T})$. The theorem shows that establishing an optimal high probability regret bound does not imply any finite PAC bound. While PAC bounds may be converted to regret bounds, the resulting bounds are necessarily severely suboptimal with a rate of $T^{2/3}$. The next theorem formalises the claim that Uniform-PAC is stronger than both the PAC and high-probability regret criteria.

Theorem 23. *Suppose an algorithm is Uniform-PAC for some δ with $F_{UPAC} = \tilde{O}(C_1/\epsilon + C_2/\epsilon^2)$ where $C_1, C_2 > 0$ are constant in ϵ, but may depend on other quantities such as S, A, H, $\log(1/\delta)$, then the algorithm*

(a) converges to optimal policies with high probability: $\mathbb{P}(\lim_{k\to\infty} \Delta_k = 0) \geq 1 - \delta$.

(b) is (ϵ, δ)-PAC with bound $F_{PAC} = F_{UPAC}$ for all ϵ.

(c) enjoys a high-probability regret at level δ with $F_{UHPR} = \tilde{O}(\sqrt{C_2 T} + \max\{C_1, C_2\})$.

Observe that stronger uniform PAC bounds lead to stronger regret bounds and for RL in episodic MDPs, an optimal uniform-PAC bound implies a uniform regret bound. To our knowledge, there are no existing approaches with PAC or regret guarantees that are Uniform-PAC. PAC methods such as MBIE, MoRMax, UCRL-γ, UCFH, Delayed Q-Learning or Median-PAC all depend on advance knowledge of ϵ and eventually stop improving their policies. Even when disabling the stopping condition, these methods are not uniform-PAC as their confidence bounds only hold for finitely many episodes and are eventually violated according to the law of iterated logarithms. Existing algorithms with uniform high-probability regret bounds such as UCRL2 or UCBVI (Azar, Osband, and Munos, 2017) also do not satisfy uniform-PAC bounds since they use upper confidence bounds with width $\sqrt{\log(T)/n}$ where T is the number of observed episodes and n is the number of observations for a specific state and action. The presence of $\log(T)$ causes the algorithm to try each action in each state infinitely often. One might begin to wonder if uniform-PAC is too good to be true. Can *any* algorithm meet the requirements? We demonstrate in Section 4.4 that the answer is yes by showing that `UBEV` has meaningful Uniform-PAC bounds. A key technique that allows us to prove these bounds is the use of finite-time law of iterated logarithm confidence bounds which decrease at rate $\sqrt{(\log\log n)/n}$.

4.3 The UBEV Algorithm

The pseudo-code for the proposed UBEV algorithm is given in Algorithm 3. In each episode it follows an optimistic policy π_k that is computed by backwards induction using a carefully chosen confidence interval on the transition probabilities in each state. In line 8 an optimistic estimate of the Q-function for the current state-action-time triple is computed using the empirical estimates of the expected next state value $\hat{V}_{\text{next}} \in \mathbb{R}$ (given that the values at the next time are $\tilde{V}_{t+1}$) and expected immediate reward $\hat{r}$ plus confidence bounds $(H-t)\phi$ and ϕ. We show in Lemma 27 later in this chapter that the policy update in Lines 3–9 finds an optimal solution to $\max_{P',r',V',\pi'} \mathbb{E}_{s\sim p_0}[V'_1(s)]$ subject to the constraints that for all $s \in \mathcal{S}, a \in \mathcal{A}, t \in [H]$,

$$V'_t(s) = r(s,\pi'(s,t),t) + P'(s,\pi'(s,t),t)^\top V'_{t+1} \qquad \text{(Bellman Equation)} \tag{4.1}$$

$$V'_{H+1} = 0, \quad P'(s,a,t) \in \Delta_S, \quad r'(s,a,t) \in [0,1]$$

$$|[(P' - \hat{P}_k)(s,a,t)]^\top V'_{t+1}| \le \phi(s,a,t)(H-t)$$

$$|r'(s,a,t) - \hat{r}_k(s,a,t)| \le \phi(s,a,t) \tag{4.2}$$

where $(P' - \hat{P}_k)(s,a,t)$ is short for $P'(s,a,t) - \hat{P}_k(s,a,t) = P'(\cdot|s,a,t) - \hat{P}_k(\cdot|s,a,t)$ and

$$\phi(s,a,t) = \sqrt{\frac{2\ln\ln\max\{e, n(s,a,t)\} + \ln(18SAH/\delta)}{n(s,a,t)}} = O\left(\sqrt{\frac{\ln(SAH\ln(n(s,a,t))/\delta)}{n(s,a,t)}}\right)$$

is the width of a confidence bound with $e = \exp(1)$ and $\hat{P}_k(s'|s,a,t) = \frac{m(s',s,a,t)}{n(s,a,t)}$ are the empirical transition probabilities and $\hat{r}_k(s,a,t) = l(s,a,t)/n(s,a,t)$ the empirical immediate rewards (both at the beginning of the kth episode). Our algorithm is conceptually similar to other algorithms based on the optimism principle such as MBIE (Strehl, Li, and Littman, 2009), UCFH (Dann and Brunskill, 2015), UCRL2 (Jaksch, Ortner, and Auer, 2010) or UCRL-γ (Lattimore and Hutter, 2012) but there are several key differences:

- Instead of using confidence intervals over the transition kernel by itself, we incorporate the value function directly into the concentration analysis. Ultimately this saves a factor of S in the sample complexity, but the price is a more difficult analysis. Previously MoRMax (Szita and Szepesvári, 2010) also used the idea of directly bounding the transition and value function, but in a very different algorithm that required discarding data and had a less tight bound. A similar technique has been used by Azar, Osband, and Munos (2017).
- Many algorithms update their policy less and less frequently (usually when the number of samples doubles), and only finitely often in total. Instead, we update the policy after every episode, which means that UBEV immediately leverages new observations.
- Confidence bounds in existing algorithms that keep improving the policy (e.g. Jaksch, Ortner, and Auer (2010) and Azar, Osband, and Munos (2017)) scale at a rate $\sqrt{\log(k)/n}$ where k is the number of episodes played so far and n is the number of times the specific (s,a,t) has been observed. As the results of a brief empirical comparison in Figure 4.2 indicate, this leads to slow learning (compare UCBVI_1 and UBEV's performance which differ essentially only by their use of different rate bounds). Instead the width of UBEV's confidence bounds ϕ scales at rate $\sqrt{\ln\ln(\max\{e,n\})/n} \approx \sqrt{(\log\log n)/n}$ which is the best achievable rate and results in significantly faster learning.

Algorithm 3: UBEV (**U**pper **B**ounding the **E**xpected Next State **V**alue) Algorithm

Input : failure tolerance $\delta \in (0, 1]$

$n(s,a,t) = l(s,a,t) = m(s',s,a,t) = 0; \quad \tilde{V}_{H+1}(s') := 0 \quad \forall s, s' \in \mathcal{S}, a \in \mathcal{A}, t \in [H]$

for $k = 1, 2, 3, \ldots$ **do**

/* Optimistic planning */

  for $t = H$ **to** 1 **do**

    for $s \in \mathcal{S}$ **do**

      for $a \in \mathcal{A}$ **do**

        $\phi := \sqrt{\frac{2 \ln \ln(\max\{e, n(s,a,t)\}) + \ln(18SAH/\delta)}{n(s,a,t)}}$ // confidence bound

        $\hat{r} := \frac{l(s,a,t)}{n(s,a,t)}; \quad \hat{V}_{\text{next}} := \frac{m(\cdot,s,a,t)^\top \tilde{V}_{t+1}}{n(s,a,t)}$ // empirical estimates

        $Q(a) := \min\{1, \hat{r} + \phi\} + \min\left\{\max \tilde{V}_{t+1}, \hat{V}_{\text{next}} + (H-t)\phi\right\}$

      $\pi_k(s,t) := \operatorname{argmax}_a Q(a), \quad \tilde{V}_t(s) := Q(\pi_k(s,t))$

/* Execute policy for one episode */

  $s_1 \sim p_0$;

  for $t = 1$ **to** H **do**

    $a_t := \pi_k(s_t, t)$, $r_t \sim p_R(s_t, a_t, t)$ and $s_{t+1} \sim P(s_t, a_t, t)$

    $n(s_t, a_t, t)$++; $\quad m(s_{t+1}, s_t, a_t, t)$++; $\quad l(s_t, a_t, t)$+= r_t // update statistics

4.4 Uniform PAC Analysis

We now discuss the Uniform-PAC analysis of UBEV which results in the following Uniform-PAC and regret guarantee.

Theorem 24. *Let π_k be the policy of UBEV in the kth episode. Then with probability at least $1 - \delta$ for all $\epsilon > 0$ jointly the number of episodes k where the expected return from the start state is not ϵ-optimal (that is $\Delta_k > \epsilon$) is at most*

$$O\left(\frac{SAH^4}{\epsilon^2} \min\{1 + \epsilon S/H, S\} \operatorname{polylog}\left(A, S, H, \frac{1}{\epsilon}, \frac{1}{\delta}\right)\right).$$

Therefore, with probability at least $1 - \delta$ UBEV converges to optimal policies and for all episodes T has regret

$$R(T) = O\left(H^2(\sqrt{SAT} + S^2AH^3) \operatorname{polylog}(S, A, H, T)\right).$$

Here $\operatorname{polylog}(x \ldots)$ is a function that can be bounded by a polynomial of logarithm, that is, $\exists k, C$: $\operatorname{polylog}(x \ldots) \leq \ln(x \ldots)^k + C$. In Section 4.8 we provide a lower bound on the sample complexity that shows that if $\epsilon < H/S$, the Uniform-PAC bound is tight up to log-factors and a factor of H. To our knowledge, UBEV is the first algorithm with both near-tight (up to H factors) high probability regret and (ϵ, δ) PAC bounds as well as the first algorithm with any nontrivial uniform-PAC bound.

Using Theorem 23 the convergence and regret bound follows immediately from the uniform PAC bound. After a discussion of the different confidence bounds allowing us to prove uniform-PAC bounds, we will provide a short proof sketch of the uniform PAC bound.

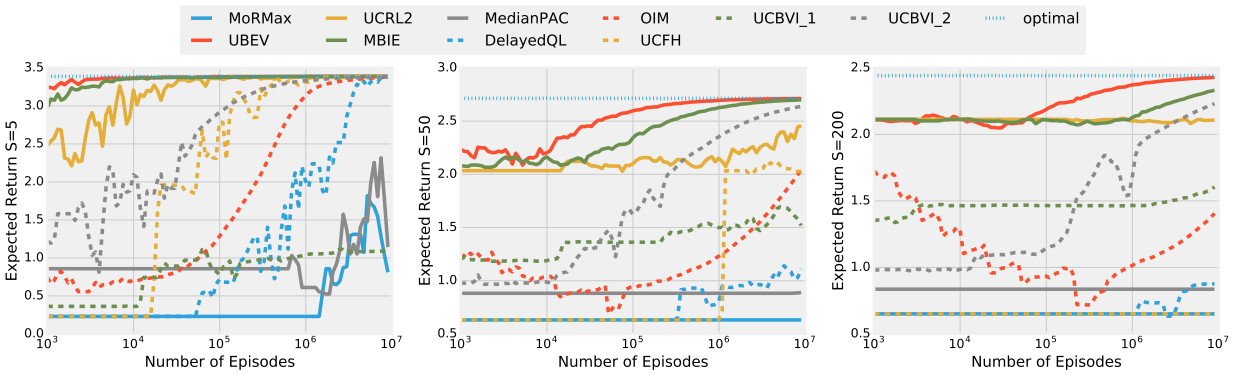

Figure 4.2: Empirical comparison of optimism-based algorithms with frequentist regret or PAC bounds on a randomly generated MDP with 3 actions, time horizon 10 and $S = 5, 50, 200$ states. All algorithms are run with parameters that satisfy their bound requirements. A detailed description of the experimental setup including a link to the source code can be found in Section 4.7.

4.4.1 Enabling Uniform PAC With Law-of-Iterated-Logarithm Confidence Bounds

To have a PAC bound for all ϵ jointly, it is critical that UBEV continually make use of new experience. If UBEV stopped leveraging new observations after some fixed number, it would not be able to distinguish with high probability among which of the remaining possible MDPs do or do not have optimal policies that are sufficiently optimal in the other MDPs. The algorithm therefore could potentially follow a policy that is not at least ϵ-optimal for infinitely many episodes for a sufficiently small ϵ. To enable UBEV to incorporate all new observations, the confidence bounds in UBEV must hold for an infinite number of updates. We therefore require a proof that the total probability of all possible failure events (of the high confidence bounds not holding) is bounded by δ, in order to obtain high probability guarantees. In contrast to prior (ϵ, δ)-PAC proofs that only consider a finite number of failure events (which is enabled by requiring an RL algorithm to stop using additional data), we must bound the probability of an infinite set of possible failure events.

Some choices of confidence bounds will hold uniformly across all sample sizes but are not sufficiently tight for uniform PAC results. For example, the recent work by Azar, Osband, and Munos (2017) uses confidence intervals that shrink at a rate of $\sqrt{\frac{\ln T}{n}}$, where T is the number of episodes, and n is the number of samples of a (s, a) pair at a particular time step. This confidence interval will hold for all episodes, but these intervals do not shrink sufficiently quickly and can even increase. One simple approach for constructing confidence intervals that is sufficient for uniform PAC guarantees is to combine bounds for fixed number of samples with a union bound allocating failure probability δ/n^2 to the failure case with n samples. This results in confidence intervals that shrink at rate $\sqrt{1/n \ln n}$. Interestingly we know of no algorithms that do such in our setting.

We follow a similarly simple but much stronger approach of using law-of-iterated logarithm (LIL) bounds that shrink at the better rate of $\sqrt{1/n \ln \ln n}$. Such bounds have sparked recent interest in sequential decision making (Jamieson et al., 2014; Balsubramani and Ramdas, 2016; Garivier, Lattimore, and Kaufmann, 2016; Massart, 2007; Garivier and Cappé, 2011) but to the best of our knowledge we are the first to leverage them for RL. We prove several general LIL bounds Section 4.11 and explain how we use these results in our analysis in Section 4.10.2. These LIL bounds are both sufficient to ensure uniform PAC bounds, and much tighter (and therefore will lead to much better performance) than $\sqrt{1/n \ln T}$ bounds. Indeed, LIL have the tightest possible rate dependence on the number of samples n for a bound that holds

for all timesteps (though they are not tight with respect to constants).

4.4.2 Proof Sketch

We now provide a short overview of our uniform PAC bound in Theorem 24. It follows the typical scheme for optimism based algorithms: we show that in each episode UBEV follows a policy that is optimal with respect to the MDP $\tilde{M}_k$ that yields highest expected return in a set of MDPs $\mathcal{M}_k$ given by the constraints in Eqs. (4.1)–(4.2) (Lemma 27 in the later sections). We then define a failure event F (more details see below) such that on the complement F^C, the true MDP is in $\mathcal{M}_k$ for all k.

Under the event that the true MDP is in the desired set, the $V_1^{\pi} \leq V_1^{\star} \leq \tilde{V}_1^{\pi_k}$, i.e., the value $\tilde{V}_1^{\pi_k}$ of π_k in MDP $\tilde{M}_k$ is higher than the optimal value function of the true MDP M (Lemma 45). Therefore, the optimality gap is bounded by $\Delta_k \leq p_0^\top(\tilde{V}_1^{\pi_k} - V_1^{\pi_k})$. The right hand side this expression is then decomposed via a standard identity (Lemma 44) as

$$\sum_{t=1}^{H} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} w_{tk}(s,a)((\tilde{P}_k - P)(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k} + \sum_{t=1}^{H} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} w_{tk}(s,a)(\tilde{r}_k(s,a,t) - r(s,a,t)),$$

where $w_{tk}(s,a)$ is the probability that when following policy π_k in the true MDP we encounter $s_t = s$ and $a_t = a$. The quantities $\tilde{P}_k$, $\tilde{r}_k$ are the model parameters of the optimistic MDP $\tilde{M}_k$ For the sake of conciseness, we ignore the second term above in the following which can be bounded by $\epsilon/3$ in the same way as the first. We further decompose the first term as

$$\sum_{\substack{t\in[H]\\(s,a)\in L_{tk}^c}} w_{tk}(s,a)((\tilde{P}_k - P)(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k} \tag{4.3}$$

$$+\sum_{\substack{t\in[H]\\(s,a)\in L_{tk}}} w_{tk}(s,a)((\tilde{P}_k - \hat{P}_k)(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k} + \sum_{\substack{t\in[H]\\(s,a)\in L_{tk}}} w_{tk}(s,a)((\hat{P}_k - P)(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k} \tag{4.4}$$

where $L_{tk} = \left\{(s,a) \in \mathcal{S}\times\mathcal{A} : w_{tk}(s,a) \geq w_{\min} = \frac{\epsilon}{3HS^2}\right\}$ is the set of state-action pairs with non-negligible visitation probability. The value of $w_{\min}$ is chosen so that (4.3) is bounded by $\epsilon/3$. Since $\tilde{V}^{\pi_k}$ is the optimal solution of the optimization problem in Eq. (4.1), we can bound

$$|((\tilde{P}_k - \hat{P}_k)(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| \leq \phi_k(s,a,t)(H-t) = O\left(\sqrt{\frac{H^2 \ln\left(\ln(n_{tk}(s,a))/\delta\right)}{n_{tk}(s,a)}}\right), \tag{4.5}$$

where $\phi_k(s,a,t)$ is the value of $\phi(s,a,t)$ and $n_{tk}(s,a)$ the value of $n(s,a,t)$ right before episode k. Further we decompose

$$|((\hat{P}_k - P)(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| \leq \|(\hat{P}_k - P)(s,a,t)\|_1 \|\tilde{V}_{t+1}^{\pi_k}\|_\infty \leq O\left(\sqrt{\frac{SH^2 \ln \frac{\ln n_{tk}(s,a)}{\delta}}{n_{tk}(s,a)}}\right), \tag{4.6}$$

where the second inequality follows from a standard concentration bound used in the definition of the failure event F (see below). Substituting this and (4.5) into (4.4) leads to

$$(4.4) \leq O\left(\sum_{t=1}^{H} \sum_{s,a\in L_{tk}} w_{tk}(s,a)\sqrt{\frac{SH^2 \ln(\ln(n_{tk}(s,a))/\delta)}{n_{tk}(s,a)}}\right). \tag{4.7}$$

On F^C it also holds that $n_{tk}(s,a) \geq \frac{1}{2}\sum_{i<k} w_{ti}(s,a) - \ln\frac{9SAH}{\delta}$ and so on *nice episodes* where each $(s,a) \in L_{tk}$ with significant probability $w_{tk}(s,a)$ also had significant probability in the past, i.e., $\sum_{i<k} w_{ti}(s,a) \geq 4\ln\frac{9SA}{\delta}$, it holds that $n_{tk}(s,a) \geq \frac{1}{4}\sum_{i<k} w_{ti}(s,a)$. Substituting this into (4.7), we can use a careful pidgeon-hole argument laid out it Lemma 35 to show that this term is bounded by $\epsilon/3$ on all but $O(AS^2H^4/\epsilon^2 \operatorname{polylog}(A,S,H,1/\epsilon,1/\delta))$ nice episodes. Again using a pidgeon-hole argument, one can show that all but at most $O(S^2AH^3/\epsilon \ln(SAH/\delta))$ episodes are nice. Combining both bounds, we get that on F^C the optimality gap Δ_k is at most ϵ except for at most $O(AS^2H^4/\epsilon^2 \operatorname{polylog}(A,S,H,1/\epsilon,1/\delta))$ episodes.

We decompose the failure event into multiple components. In addition to the events F_k^N that a (s,a,t) triple has been observed few times compared to its visitation probabilities in the past, i.e., $n_{tk}(s,a) < \frac{1}{2}\sum_{i<k} w_{ti}(s,a) - \ln\frac{9SAH}{\delta}$ as well as a conditional version of this statement, the failure event F contains events where empirical estimates of the immediate rewards, the expected optimal value of the successor states and the individual transition probabilites are far from their true expectations. For the full definition of F see Section 4.10.2. F also contains event F^{L1} we used in Eq. (4.6) defined as

$$\left\{\exists k,s,a,t \,:\, \|\hat{P}_k(s,a,t) - P(s,a,t)\|_1 \geq \sqrt{\tfrac{4}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\tfrac{18SAH(2^S-2)}{\delta}\right)}\right\}.$$

It states that the L1-distance of the empirical transition probabilities to the true probabilities for any (s,a,t) in any episode k is too large and we show that $\mathbb{P}(F^{L1}) \leq 1-\delta/9$ using a uniform version of the popular bound by Weissman et al. (2003) which we prove in Section 4.11. We show in similar manner that the other events in F have small probability uniformly for all episodes k so that $\mathbb{P}(F) \leq \delta$. Together this yields the uniform PAC bound in Thm. 24 using the second term in the $\min$.

With a more refined analysis that avoids the use of Hölder's inequality in (4.6) we obtain the bound with the first term in the $\min$. However, since a similar analysis has been recently released (Azar, Osband, and Munos, 2017), we defer this discussion to the later sections.

4.4.3 Discussion of UBEV Bound

The (Uniform-)PAC bound for UBEV in Theorem 24 is never worse than $\tilde{O}(S^2AH^4/\epsilon^2)$, which improves on the similar MBIE algorithm by a factor of H^2 (after adapting the discounted setting for which MBIE was analysed to our setting). For $\epsilon < H/S^2$ our bound has a linear dependence on the size of the state-space and depends on H^4, which is a tighter dependence on the horizon than MoRMax's $\tilde{O}(SAH^6/\epsilon^2)$, the best sample-complexity bound with linear dependency S so far.

Comparing UBEV's regret bound to the ones of UCRL2 (Jaksch, Ortner, and Auer, 2010) and REGAL (Bartlett and Tewari, 2009) requires care because (a) we measure the regret over entire episodes and (b) our transition dynamics are time-dependent within each episode, which effectively increases the state-space by a factor of H. Converting the bounds for UCRL2/REGAL to our setting yields a regret bound of order $SH^2\sqrt{AHT}$. Here, the diameter is H, the state space increases by H due to time-dependent transition dynamics and an additional $\sqrt{H}$ is gained by stating the regret in terms of episodes T instead of time steps. Hence, UBEV's bounds are better by a factor of $\sqrt{SH}$. Our bound matches the recent regret bound for episodic RL developed in parallel by Azar, Osband, and Munos (2017) in the S, A and T terms but not in H. Azar, Osband, and Munos (2017) has regret bounds that close to optimal in H (up to a spurious $\sqrt{H^3T}$ term and lower-order terms) but their algorithm is not uniform PAC, due to the characteristics we outlined in Section 4.2.

4.5 Summary

The Uniform-PAC framework strengthens and unifies the mistake-style PAC and high-probability regret performance criteria for reinforcement learning in episodic MDPs. The newly proposed algorithm is Uniform-PAC, which as a side-effect means it is the first algorithm that is both PAC and has sub-linear (and nearly optimal) regret. Besides this, the use of law-of-the-iterated-logarithm confidence bounds in RL algorithms for MDPs provides a practical and theoretical boost at no cost in terms of computation or implementation complexity.

This work opens up several immediate research questions for future work. The definition of Uniform-PAC and the relations to other PAC and regret notions directly apply to multi-armed bandits and contextual bandits as special cases of episodic RL, but not to infinite horizon reinforcement learning. An extension to these non-episodic RL settings is highly desirable. Similarly, a version of the `UBEV` algorithm for infinite-horizon RL with linear state-space sample complexity would be of interest. More broadly, if theory is ever to say something useful about practical algorithms for large-scale reinforcement learning, then it will have to deal with the unrealizable function approximation setup (unlike the tabular function representation setting considered here), which is a major long-standing open challenge.

4.6 Framework Relation Proofs

4.6.1 Proof of Theorem 21

Proof. We will use two episodic MDPs, M_1 and M_2, which are essentially 2-armed bandits and hard to distinguish to prove this statement. Both MDPs have one state, horizon $H = 1$, and two actions $\mathcal{A} = \{1, 2\}$. For a fixed $\alpha > 0$, the rewards are Bernoulli$(1/2 + \alpha/2)$ distributed for actions 1 in both MDPs. Playing action 2 in M_1 gives Bernoulli$(1/2)$ rewards and action 2 in M_2 gives Bernoulli$(1/2 + \alpha)$ rewards.

Assume now that an algorithm in MDP M_1 with nonzero probability plays the suboptimal action only at most N times in total, i.e., $\mathbb{P}_{M_1}(n_2 \leq N) \geq \beta$ where n_2 is the number of times action 2 is played and $\infty > N > 0, \beta > 0$. Then

$$\mathbb{P}_{M_1}(n_2 \leq N) = \mathbb{E}_{M_1}\left[\mathbb{I}\{n_2 \leq N\}\right] = \mathbb{E}_{M_2}\left[\frac{\mathbb{P}_{M_1}(Y_\infty)}{\mathbb{P}_{M_2}(Y_\infty)}\mathbb{I}\{n_2 \leq N\}\right]$$

where $Y_k = (A_1, R_1, A_2, R_2, \dots A_k, R_k)$ denotes the entire sequence of observed rewards R_i and action indices A_i after k episodes. Since $\mathbb{P}_{M_1}(A_k|Y_{k-1}) = \mathbb{P}_{M_2}(A_k|Y_{k-1})$ and $\mathbb{P}_{M_1}(R_k|A_k = 1, Y_{k-1}) = \mathbb{P}_{M_2}(R_k|A_k = 1, Y_{k-1})$ and

$$\frac{\mathbb{P}_{M_1}(R_k|A_k = 2, Y_{k-1})}{\mathbb{P}_{M_2}(R_k|A_k = 2, Y_{k-1})} \leq \max\left\{\frac{1/2}{1/2 + \alpha}, \frac{1/2}{1/2 - \alpha}\right\} = \frac{1}{1 - 2\alpha}$$

the likelihood ratio of Y_∞ is upper bounded by $(1 + 2\alpha)^N$ if the second action has been chosen at most N times. Hence

$$\begin{aligned}\mathbb{P}_{M_2}\left[n_2 \leq N\right] = \frac{(1 - 2\alpha)^N}{(1 - 2\alpha)^N}\mathbb{E}_{M_2}\left[\mathbb{I}\{n_2 \leq N\}\right] &\geq (1 - 2\alpha)^N \mathbb{E}_{M_2}\left[\frac{\mathbb{P}_{M_1}(Y_\infty)}{\mathbb{P}_{M_2}(Y_\infty)}\mathbb{I}\{n_2 \leq N\}\right] \\ &\geq (1 - 2\alpha)^N \beta > 0\end{aligned}$$

Therefore, the regret for M_2 is for T large enough $\mathbb{E}_{M_2} R(T) \geq (T - N)\beta(1 - 2\alpha)^N \alpha/2 = O(T)$. Hence, for the algorithm to ensure sublinear regret for M_2, it has to play the suboptimal action for M_1 infinitely often with probability 1. This however implies that the algorithm cannot satisfy any finite PAC bound for accuracy $\epsilon < \alpha/2$. □

4.6.2 Proof of Theorem 22

Proof. **PAC Bound to high-probability regret bound:** Consider a fixed $\delta > 0$ and PAC bound with $F_{\text{PAC}} = \Theta(1/\epsilon^2)$. Then there is a $C > 0$ such that the following algorithm satisfies the PAC bound. The algorithm uses the worst possible policy with optimality gap H in all episodes on some event E and in the first C/ϵ^2 episodes on the complimentary event E^C. For the remaining episodes on E^C it follows a policy with optimality gap ϵ. The probability of E is δ. The regret of the algorithm on E is $R(T) = TH$ and on E^C it is $R(T) = \min\{T, C/\epsilon^2\}H + \min\{T - C/\epsilon^2, 0\}\epsilon$. For $T \geq C/\epsilon^2$, on any event the regret of this algorithm is at least

$$R(T) = \frac{CH}{\epsilon^2} + \left(T - \frac{C}{\epsilon^2}\right)\epsilon = T\epsilon + \frac{C(H-\epsilon)}{\epsilon^2}. \tag{4.8}$$

The quantity

$$\frac{R(T)}{T^{2/3}} = \frac{C(H-\epsilon)}{T^{2/3}\epsilon^2} + \epsilon T^{1/3}$$

takes its minimum at $T = \frac{C(H-\epsilon)}{\epsilon^3}$ with a positive value and hence $R(T) = \Omega(T^{2/3})$. Therefore a PAC bound with rate $1/\epsilon^2$ implies at best a high-probability regret bound of order $O(T^{2/3})$ and is only tight at $T = \Theta(1/\epsilon^3)$. Furthermore, by looking at Equation (4.8), we see that for any fixed ϵ, there is an algorithm that has uniform high-probability regret that is $\Omega(T)$.

PAC Bound to uniform high-probability regret bound: Consider a fixed $\delta > 0$ and $\epsilon > 0$ and a PAC bound F_{PAC} that evaluates to some value N for parameter ϵ. The algorithm uses the worst possible policy with optimality gap H in all episodes on some event E and in the first N episodes on the complimentary event E^C. For the remaining episodes on E^C it follows a policy with optimality gap ϵ. The probability of E is δ. The regret of the algorithm on E is $R(T) = TH$ and on E^C it is $R(T) = \min\{T, N\}H + \min\{T - N, 0\}\epsilon$. For $T \geq N$, on any event the regret of this algorithm is at least

$$R(T) = NH + (T - N)\,\epsilon = T\epsilon + H(T - N) = \Omega(T).$$

Uniform high-probability regret bound to PAC bound: Consider an MDP such that at least one suboptimal policy exists with optimality gap $\epsilon > 0$. Further let $L(T)$ be a nondecreasing function with $F_{\text{UHPR}}(T) \geq L(T)$ and $L(T) \to \infty$ as $T \to \infty$. Then the algorithm plays the optimal policy except for episodes k where $\lfloor L(k-1)/\epsilon \rfloor \neq \lfloor L(k)/\epsilon \rfloor$. This algorithm satisfies the regret bound but makes infinitely many $\epsilon/2$-mistakes with probability 1.

Uniform high-probability regret bound to expected regret bound: Consider an MDP such that at least one suboptimal policy exists with optimality gap $\epsilon > 0$. Consider an algorithm that with probability δ always plays the suboptimal policy and with probability $1 - \delta$ always plays the optimal policy. This algorithm satisfies the uniform high-probability regret bound but suffers regret $\mathbb{E}R(T) = \delta\epsilon T = \Omega(T)$. □

4.6.3 Proof of Theorem 23

Proof. **Convergence to optimal policies:** The convergence to the set of optimal policies follows directly by using the definition of limits on the Δ_k sequence for each outcome in the high-probability event where the bound holds.

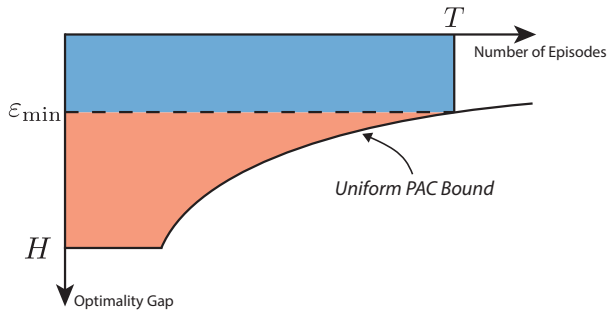

Figure 4.3: Relation of PAC-bound and Regret; The area of the shaded regions are a bound on the regret after T episodes.

(ϵ, δ)**-PAC:** Due to sub-additivity of probabilities, we have

$$\mathbb{P}\left(N_\epsilon > F_{\text{PAC}}\left(\frac{1}{\epsilon}, \log\frac{1}{\delta}\right)\right) \leq \mathbb{P}\left(\bigcup_{\epsilon'}\left\{N_{\epsilon'} > F_{\text{PAC}}\left(\frac{1}{\epsilon'}, \log\frac{1}{\delta}\right)\right\}\right)$$
$$= \mathbb{P}\left(\exists \epsilon' \,:\, N_{\epsilon'} > F_{\text{PAC}}\left(\frac{1}{\epsilon'}, \log\frac{1}{\delta}\right)\right) \leq \delta.$$

High-Probability Regret Bound: This part is proved separately in Theorem 25 below. □

Theorem 25 (Uniform-PAC to Regret Conversion Theorem)**.** *Assume on some event E an algorithm follows for all ϵ an ϵ-optimal policy π_k, i.e., $\Delta_k \leq \epsilon$, on all but at most*

$$\frac{C_1}{\epsilon}\left(\ln\frac{C_3}{\epsilon}\right)^k + \frac{C_2}{\epsilon^2}\left(\ln\frac{C_3}{\epsilon}\right)^{2k}$$

episodes where $C_1 \geq C_2 \geq 2$ and $C_3 \geq \max\{H, e\}$ and C_1, C_2, C_3 do not depend on ϵ . Then this algorithm has on this event a regret of

$$R(T) \leq (\sqrt{C_2 T} + C_1)\,\text{polylog}(T, C_3, C_1) = O(\sqrt{C_2 T}\,\text{polylog}(T, C_3, C_1, H))$$

for all number of episodes T.

Proof. The mistake bound $g(\epsilon) = \frac{C_1}{\epsilon}\left(\ln\frac{C_3}{\epsilon}\right)^k + \frac{C_2}{\epsilon^2}\left(\ln\frac{C_3}{\epsilon}\right)^{2k} \leq T$ is monotonically decreasing for $\epsilon \in (0, H]$. For a given T large enough, we can therefore find an $\epsilon_{\min} \in (0, H]$ such that $g(\epsilon) \leq T$ for all $\epsilon \in (\epsilon_{\min}, H]$. The regret $R(T)$ of the algorithm can then be bounded as follows

$$R(T) \leq T\epsilon_{\min} + \int_{\epsilon_{\min}}^{H} g(\epsilon)d\epsilon.$$

This bound assumes the worst case where first the algorithm makes the worst mistakes possible with regret H and subsequently less and less severe mistakes controlled by the mistake bound. For a better intuition, see Figure 4.3.

We first find a suitable $\epsilon_{\min}$. Define $y = \frac{1}{\epsilon}\left(\ln\frac{C_3}{\epsilon}\right)^k$ then since g is monotonically decreasing, it is sufficient to find a ϵ with $g(\epsilon) \leq T$. That is equivalent to $C_1 y + C_2 y^2 \leq T$ for which

$$\frac{1}{\epsilon}\left(\ln\frac{C_3}{\epsilon}\right)^k = y \leq \frac{C_1}{2C_2} + \frac{\sqrt{C_1^2 + 4TC_2}}{2C_2} =: a$$

is sufficient. We set now

$$\epsilon_{\min} = \frac{\ln(C_3 a)^k}{a} = \frac{2C_2}{C_1 + \sqrt{C_1^2 + 4TC_2}} \left(\ln \frac{(C_1 + \sqrt{C_1^2 + 4TC_2})C_3}{2C_2} \right)^k$$

which is a valid choice as

$$\begin{aligned}\frac{1}{\epsilon_{\min}} \left(\ln \frac{C_3}{\epsilon_{\min}} \right)^k =& \frac{a}{\ln(C_3 a)^k} \left(\ln \frac{C_3 a}{\ln(C_3 a)^k} \right)^k = \frac{a}{\ln(C_3 a)^k} \left(\ln(C_3 a) - k \ln \ln(C_3 a) \right)^k \\ \leq& \frac{a}{\ln(C_3 a)^k} \left(\ln(C_3 a) \right)^k = a.\end{aligned}$$

We now first bound the regret further as

$$\begin{aligned}R(T) \leq& T\epsilon_{\min} + \int_{\epsilon_{\min}}^{H} g(\epsilon) d\epsilon \leq T\epsilon_{\min} + C_1 \left(\ln \frac{C_3}{\epsilon_{\min}} \right)^k \int_{\epsilon_{\min}}^{H} \frac{1}{\epsilon} d\epsilon + C_2 \left(\ln \frac{C_3}{\epsilon_{\min}} \right)^{2k} \int_{\epsilon_{\min}}^{H} \frac{1}{\epsilon^2} d\epsilon \\ =& T\epsilon_{\min} + C_1 \left(\ln \frac{C_3}{\epsilon_{\min}} \right)^k \ln \frac{H}{\epsilon_{\min}} + C_2 \left(\ln \frac{C_3}{\epsilon_{\min}} \right)^{2k} \left[\frac{1}{\epsilon_{\min}} - \frac{1}{H} \right]\end{aligned}$$

and then use the choice of $\epsilon_{\min}$ from above to look at each of the terms in this bound individually. In the following bounds we extensively use the fact $\ln(a+b) \leq \ln(a) + \ln(b) = \ln(ab)$ for all $a, b \geq 2$ and that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ which holds for all $a, b \geq 0$.

$$\begin{aligned}T\epsilon_{\min} &= \frac{2TC_2}{C_1 + \sqrt{C_1^2 + 4TC_2}} \left(\ln \frac{C_3(C_1 + \sqrt{C_1^2 + 4TC_2})}{2C_2} \right)^k \\ &\leq \frac{2TC_2}{\sqrt{4TC_2}} \left(\ln C_3 + \ln C_1 + \ln C_1 + \ln \frac{2\sqrt{TC_2}}{2C_2} \right)^k \\ &\leq \sqrt{TC_2} \left(\ln(C_3 C_1^2 \sqrt{T}) \right)^k\end{aligned}$$

Now for a $C \geq 0$ we first look at

$$\begin{aligned}\ln \frac{C}{\epsilon_{\min}} =& \ln C + \ln \frac{C_1 + \sqrt{C_1^2 + 4TC_2}}{2C_2} - k \ln \ln \frac{C_3(C_1 + \sqrt{C_1^2 + 4TC_2})}{2C_2} \\ \leq& \ln C + \ln \frac{C_1 + \sqrt{C_1^2 + 4TC_2}}{2C_2} \\ \leq& \ln C + \ln C_1 + \ln C_1 + \ln \frac{\sqrt{4TC_2}}{2C_2} \\ \leq& \ln(C C_1^2 \sqrt{T})\end{aligned}$$

where the first inequality follows from the fact that $\frac{C_3(C_1 + \sqrt{C_1^2 + 4TC_2})}{2C_2} \geq \frac{C_3 2C_1}{2C_2} \geq e$. Hence, we can bound

$$C_1 \left(\ln \frac{C_3}{\epsilon_{\min}} \right)^k \ln \frac{H}{\epsilon_{\min}} \leq C_1 \left(\ln(C_3 C_1^2 \sqrt{T}) \right)^k \ln(H C_1^2 \sqrt{T}).$$

Now since

$$\frac{1}{\epsilon_{\min}} = \frac{C_1 + \sqrt{C_1^2 + 4TC_2}}{2C_2} \left(\ln \frac{C_3(C_1 + \sqrt{C_1^2 + 4TC_2})}{2C_2} \right)^{-k} \leq \frac{C_1}{C_2} + \sqrt{\frac{T}{C_2}}$$

we get

$$C_2 \left(\ln \frac{C_3}{\epsilon_{\min}}\right)^{2k} \left[\frac{1}{\epsilon_{\min}} - \frac{1}{H}\right] \leq C_2 \left(\ln(C_3 C_1^2 \sqrt{T})\right)^{2k} \left[\frac{C_1}{C_2} + \sqrt{\frac{T}{C_2}}\right]$$
$$\leq \left(\ln(C_3 C_1^2 \sqrt{T})\right)^{2k} \left[C_1 + \sqrt{T C_2}\right].$$

As a result we can conclude that

$$R(T) \leq (\sqrt{C_2 T} + C_1) \operatorname{polylog}(T, C_3, C_1, H) = O(\sqrt{C_2 T} \operatorname{polylog}(T, C_3, C_1, H))$$

□

4.7 Experimental Details

We generated the MDPs with $S = 5, 50, 200$ states, $A = 3$ actions and $H = 10$ timesteps as follows: The transition probabilities $P(s, a, t)$ were sampled independently from Dirichlet $\left(\frac{1}{10}, \ldots \frac{1}{10}\right)$ and the rewards were all deterministic with their value $r(s, a, t)$ set to 0 with probability 85% and set uniformly at random in $[0, 1]$ otherwise. This construction results in MDPs that have concentrated but non-deterministic transition probabilities and sparse rewards.

Since some algorithms have been proposed assuming the rewards $r(s, a, t)$ are known and we aim for a fair comparison, we assumed for all algorithms that the immediate rewards $r(s, a, t)$ are known and adapted the algorithms accordingly. For example, in `UBEV`, the $\min\left\{1, \frac{l(s,a,t)}{\max\{1, n(s,a,t)\}} + \phi\right\}$ term was replaced by the true known rewards $r(s, a, t)$ and the δ parameter in ϕ was scaled by $9/7$ accordingly since the concentration result for immediate rewards is not necessary in this case. We used $\delta = \frac{1}{10}$ for all algorithms and $\epsilon = \frac{1}{10}$ if they require to know ϵ beforehand.

We adapted MoRMax, UCRL2, UCFH, MBIE, MedianPAC, Delayed Q-Learning and OIM to the episodic MDP setting with time-dependent transition dynamics by using allowing them to learn time-dependent dynamics and use finite-horizon planning. We did adapt the confidence intervals and but did not re-derive the constants for each algorithm. When in doubt we opted for smaller constants typically resulting better performance of the competitors. We further replaced the range of the value function $O(H)$ by the observed range of the optimistic next state values in the confidence bounds. We also reduced the number of episodes used in the delays by a factor of $\frac{1}{1000}$ for MoRMax and Delayed Q-Learning and by 10^{-6} for UCFH because they would otherwise not have performed a single policy update even for $S = 5$ within the 10 million episodes we considered. This scaling violates their theoretical guarantees but at least shows that the methods work in principle.

The performance reported in Figure 4.2 are the expected return of the current policy of each algorithm averaged over 1000 episodes. The figure shows a single run of the same randomly generated MDP but the results are representative. We reran this experiments with different random seeds and consistently obtained qualitatively similar results.

Source code for the experiments including concise but efficient implementations of the algorithms is available at `https://github.com/chrodan/FiniteEpisodicRL.jl`.

4.8 PAC Lower Bound

Theorem 26. *There exist positive constants c, $\delta_0 > 0$, $\epsilon_0 > 0$ such that for every $\epsilon \in (0, \epsilon_0)$, $S \geq 4, A \geq 2$ and for every algorithm A that and $n \leq \frac{cASH^3}{\epsilon^2}$ there is a fixed-horizon episodic MDP M_{hard} with time-dependent transition probabilities and S states and A actions so that returning an ϵ-optimal policy after n*

episodes is at most $1-\delta_0$. *That implies that no algorithm can have a PAC guarantee better than* $\Omega\left(\frac{ASH^3}{\epsilon^2}\right)$ *for sufficiently small* ϵ.

Note that this lower bound on the sample complexity of any method in episodic MDPs with time-dependent dynamics applies to the arbitrary but fixed ϵ PAC bound and therefore immediately to the stronger uniform-PAC bounds. This theorem can be proved in the same way as Theorem 5 by Jiang, Krishnamurthy, et al. (2017), which itself is a standard construction involving a careful layering of difficult instances of the multi-armed bandit problem.[3] For simplicity, we omitted the dependency on the failure probability δ, but using the techniques in the proof of Theorem 26 by Strehl, Li, and Littman (2009), a lower bound of order $\Omega\left(\frac{ASH^3}{\epsilon^2}\log(SA/\delta)\right)$ can be obtained. The lower bound shows for small ϵ the sample complexity of UBEV given in Theorem 24 is optimal except for a factor of H and logarithmic terms.

4.9 Planning Problem of UBEV

Lemma 27 (Planning Problem)**.** *The policy update in Lines 3–9 of Algorithm 3 finds an optimal solution to the optimization problem*

$$
\begin{aligned}
&\max_{P',V',\pi',r'} \mathbb{E}_{s\sim p_0}[V'_1(s)] \\
&\forall s\in\mathcal{S}, a\in\mathcal{A}, t\in[H]: \\
&\quad V'_{H+1} = 0, \qquad P'(s,a,t)\in\Delta_S, \qquad r'(s,a,t)\in[0,1] \\
&\quad V'_t(s) = r'(s,\pi'(s,t),t) + \mathbb{E}_{s'\sim P'(s,\pi'(s,t),t)}[V'_{t+1}] \\
&\quad |(P'(s,a,t)-\hat{P}_k(s,a,t))^\top V'_{t+1}| \le \phi(s,a,t)(H-t) \\
&\quad |r'(s,a,t)-\hat{r}_k(s,a,t)| \le \phi(s,a,t)
\end{aligned}
$$

where $\phi(s,a,t)=\sqrt{\frac{2\,\mathrm{llnp}(n(s,a,t))+\ln(18SAH/\delta)}{n(s,a,t)}}$ *is a confidence bound and* $\hat{P}_k(s'|s,a,t)=\frac{m(s',s,a,t)}{n(s,a,t)}$ *are the empirical transition probabilities and* $\hat{r}_k(s,a,t)=l(s,a,t)/n(s,a,t)$ *the empirical average rewards.*

Proof. Since $\tilde{V}_{H+1}(\cdot)$ is initialized with 0 and never changed, we immediately get that it is an optimal value for $V'_{H+1}(\cdot)$ which is constrained to be 0. Consider now a single time step t and assume V'_{t+1} are fixed to the optimal values $\tilde{V}_{t+1}$. Plugging in the computation of $Q(a)$ into the computation of $\tilde{V}_t(s)$, we get

$$
\begin{aligned}
\tilde{V}_t(s) = \max_a Q(a) = \max_{a\in\mathcal{A}} \Big[&\min\{1, \hat{r}(s,a,t)+\phi(s,a,t)\} \\
&+ \min\Big\{\max \tilde{V}_{t+1}, \mathbb{I}\{n(s,a,t)>0\}(\hat{P}(s,a,t)^\top \tilde{V}_{t+1}) + \phi(s,a,t)(H-t)\Big\}\Big]
\end{aligned}
$$

using the convention that $\hat{r}(s,a,t)=0$ if $n(s,a,t)=0$. Assuming that $V'_{t+1}=\tilde{V}_{t+1}$, and that our goal for

[3] We here only use $H/2$ timesteps for bandits and the remaining $H/2$ time steps to accumulate a reward of $O(H)$ for each bandit

now is to maximize $\tilde{V}_t(s)$, this can be rewritten as

$$\begin{aligned}\max_{P'(s,a,t),r'(s,a,t)} \tilde{V}_t(s) = \max_{P'(s,a,t),r'(s,a,t),\pi'(s,t)} & \left[r'(s,\pi'(s,t),t) + P'(s,\pi'(s,t),t)^\top \tilde{V}_{t+1}\right] \\ \text{s.t.} \quad & \forall a \in \mathcal{A} : r'(s,a,t) \in [0,1], \qquad P'(s,a,t) \in \Delta_S \\ |(P'(s,a,t) - \hat{P}_k(s,a,t))^\top V'_{t+1}| \leq & \phi(s,a,t)(H-t) \\ |r'(s,a,t) - \hat{r}_k(s,a,t)| \leq & \phi(s,a,t)\end{aligned}$$

since in this problem either $P'(s,\pi'(s,t),t)^\top \tilde{V}_{t+1} = \hat{P}(s,\pi'(s,t),t)^\top \tilde{V}_{t+1} + \phi(s,a,t)(H-t)$ if that does not violate $P'(s,\pi'(s,t),t)^\top \tilde{V}_{t+1} \leq \max \tilde{V}_{t+1}$ and otherwise $P'(s',s,\pi'(s,t),t) = 1$ for one state s' with $\tilde{V}_{t+1}(s') = \max \tilde{V}_{t+1}$. Similarly, either $r'(s,\pi'(s,t),t) = \hat{r}(s,\pi'(s,t),t) + \phi(s,\pi'(s,t),t)$ if that does not violate $r'(s,\pi'(s,t),t) \leq 1$ or $r'(s,\pi'(s,t),t) = 1$ otherwise. Using induction for $t = H, H-1 \ldots 1$, we see that `UBEV` computes an optimal solution to

$$\begin{aligned}&\max_{P',V',\pi',r'} V'_1(\tilde{s}) \\ &\forall s \in \mathcal{S}, a \in \mathcal{A}, t \in [H] : \\ &\quad V'_{H+1} = 0, \qquad P'(s,a,t) \in \Delta_S, \qquad r'(s,a,t) \in [0,1] \\ &\quad V'_t(s) = r'(s,\pi'(s,t),t) + \mathbb{E}_{s' \sim P'(s,\pi'(s,t),t)}[V'_{t+1}] \\ &\quad |(P'(s,a,t) - \hat{P}_k(s,a,t))^\top V'_{t+1}| \leq \phi(s,a,t)(H-t) \\ &\quad |r'(s,a,t) - \hat{r}_k(s,a,t)| \leq \phi(s,a,t)\end{aligned}$$

for any fixed $\tilde{s}$. The intersection of all optimal solutions to this problem for all $\tilde{s} \in \mathcal{S}$ are also an optimal solution to

$$\begin{aligned}&\max_{P',V',\pi',r'} p_0^\top V'_1 \\ &\forall s \in \mathcal{S}, a \in \mathcal{A}, t \in [H] : \\ &\quad V'_{H+1} = 0, \qquad P'(s,a,t) \in \Delta_S, \qquad r'(s,a,t) \in [0,1] \\ &\quad V'_t(s) = r'(s,\pi'(s,t),t) + \mathbb{E}_{s' \sim P'(s,\pi'(s,t),t)}[V'_{t+1}] \\ &\quad |(P'(s,a,t) - \hat{P}_k(s,a,t))^\top V'_{t+1}| \leq \phi(s,a,t)(H-t) \\ &\quad |r'(s,a,t) - \hat{r}_k(s,a,t)| \leq \phi(s,a,t).\end{aligned}$$

Hence, `UBEV` computes an optimal solution to this problem. □

4.10 Details of PAC Analysis

In the analysis, we denote the value of $n(\cdot,t)$ after the planning in iteration k as $n_{tk}(\cdot)$. We further denote by $P(s'|s,a,t)$ the probability of sampling state s' as s_{t+1} when $s_t = s, a_t = a$. With slight abuse of notation, $P(s,a,t) \in [0,1]^S$ denotes the probability vector of $P(\cdot|s,a,t)$. We further use $\tilde{P}_k(s'|s,a,t)$ as conditional probability of $s_{t+1} = s'$ given $s_t = s, a_t = a$ but in the optimistic MDP $\tilde{M}$ computed in the

optimistic planning steps in iteration k. We also use the following definitions:

$$\begin{aligned} w_{\min} =& w'_{\min} = \frac{\epsilon c_\epsilon}{H^2 S} \\ c_\epsilon =& \frac{1}{21e} \\ L_{tk} =& \{(s,a) \in \mathcal{S} \times \mathcal{A} : w_{tk}(s,a) \geq w_{\min}\} \\ \operatorname{llnp}(x) =& \ln(\ln(\max\{x, e\})) \\ \operatorname{rng}(x) =& \max(x) - \min(x) \\ \delta' =& \frac{\delta}{8} \end{aligned}$$

In the following, we provide the formal proof for Theorem 24 and then present all necessary lemmas:

4.10.1 Proof of Theorem 24

Proof of Theorem 24. Corollary 32 ensures that the failure event has probability at most δ. Outside the failure event Lemma 34 ensures that all but at most $\frac{S^2AH^3}{\epsilon} \operatorname{polylog}(A, S, H, 1/\epsilon, 1/\delta)$ episodes are nice. Finally, Lemma 39 shows that all nice episodes except at most $\left(S + \frac{H}{\epsilon}\right) \frac{SAH^3}{\epsilon} \operatorname{polylog}(A, S, H, 1/\epsilon, 1/\delta)$ are ϵ-optimal. Furthermore, Lemma 38 shows that all nice episodes except at most $\frac{S^2AH^4}{\epsilon^2} \operatorname{polylog}(A, S, H, 1/\epsilon, 1/\delta)$ are ϵ-optimal.

□

4.10.2 Failure Events and Their Probabilities

In this section, we define a failure event F in which we cannot guarantee the performance of UBEV. We then show that this event F only occurs with low probability. All our arguments are based on general uniform concentration of measure statements that we prove in Section 4.11. In the following we argue how the apply in our setting and finally combine all concentration results to get $\mathbb{P}(F) \leq \delta$. The failure event is defined as

$$F = \bigcup_k \left[F_k^N \cup F_k^P \cup F_k^V \cup F_k^{L1} \cup F_k^R\right]$$

where

$$F_k^N = \left\{\exists s, a, t : n_{tk}(s,a) < \frac{1}{2}\sum_{i<k} w_{ti}(s,a) - \ln\frac{SAH}{\delta'}\right\}$$

$$F_k^V = \left\{\exists s, a, t : |(\hat{P}_k(s,a,t) - P(s,a,t))^\top V_{t+1}^\star| \geq \sqrt{\frac{\operatorname{rng}(V_{t+1}^\star)^2}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3SAH}{\delta'}\right)}\right\}$$

$$F_k^P = \left\{\exists s, s', a, t : |\hat{P}_k(s'|s,a,t) - P(s'|s,a,t)| \geq \sqrt{\frac{2P(s'|s,a,t)}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3S^2AH}{\delta'}\right)}\right.$$

$$\left. + \frac{1}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3S^2AH}{\delta'}\right)\right\}$$

$$F_k^{L1} = \left\{ \exists s,a,t : \|\hat{P}_k(s,a,t) - P(s,a,t)\|_1 \geq \sqrt{\frac{4}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3SAH(2^S-2)}{\delta'}\right)} \right\}$$

$$F_k^{R} = \left\{ \exists s,a,t : |\hat{r}_k(s,a,t) - r(s,a,t)| \geq \sqrt{\frac{1}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3SAH}{\delta'}\right)} \right\}.$$

We now bound the probability of each type of failure event individually:

Corollary 28. *For any $\delta' > 0$, it holds that $\mathbb{P}\left(\bigcup_{k=1}^{\infty} F_k^V\right) \leq 2\delta'$ and $\mathbb{P}\left(\bigcup_{k=1}^{\infty} F_k^R\right) \leq 2\delta'$*

Proof. Consider a fix $s \in \mathcal{S}, a \in \mathcal{A}, t \in [H]$ and denote $\mathcal{F}_k$ the sigma-field induced by the first $k-1$ episodes and the k-th episode up to s_t and a_t but not s_{t+1}. Define τ_i to be the index of the episode where (s,a) was observed at time t the ith time. Note that τ_i are stopping times with respect to $\mathcal{F}_i$. Define now the filtration $\mathcal{G}_i = \mathcal{F}_{\tau_i} = \{A \in \mathcal{F}_\infty : A \cap \{\tau_i \leq t\} \in \mathcal{F}_t \; \forall t \geq 0\}$ and $X_k = (V_{t+1}^\star(s'_k) - P(s,a,t)^\top V_{t+1}^\star)\mathbb{I}\{\tau_k < \infty\}$ where s'_i is the value of s_{t+1} in episode τ_i (or arbitrary, if $\tau_i = \infty$).

By the Markov property of the MDP, we have that X_i is a martingale difference sequence with respect to the filtration $\mathcal{G}_i$. Further, since $\mathbb{E}[X_i|\mathcal{G}_{i-1}] = 0$ and $|X_i| \in [0, \operatorname{rng}(V_{t+1}^\star)]$, X_i conditionally $\operatorname{rng}(V_{t+1}^\star)/2$-subgaussian due to Hoeffding's Lemma, i.e., satisfies $\mathbb{E}[\exp(\lambda X_i)|\mathcal{G}_{i-1}] \leq \exp(\lambda^2 \operatorname{rng}(V_{t+1}^\star)^2/2)$.

We can therefore apply Lemma 46 and conclude that

$$\mathbb{P}\left(\exists k : |(\hat{P}_k(s,a,t) - P(s,a,t))^\top V_{t+1}^\star| \geq \sqrt{\frac{\operatorname{rng}(V_{t+1}^\star)^2}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3}{\delta'}\right)}\right) \leq 2\delta'.$$

Analogously

$$\mathbb{P}\left(\exists k : |\hat{r}_k(s,a,t) - r(s,a,t)| \geq \sqrt{\frac{1}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3}{\delta'}\right)}\right) \leq 2\delta'.$$

Applying the union bound over all $s \in \mathcal{S}, a \in \mathcal{A}$ and $t \in [H]$, we obtain the desired statement for F^V. In complete analogy using the same filtration, we can show the statement for F^R. □

Corollary 29. *For any $\delta' > 0$, it holds that $\mathbb{P}\left(\bigcup_{k=1}^{\infty} F_k^P\right) \leq 2\delta'$.*

Proof. Consider first a fix $s', s \in \mathcal{S}, t \in [H]$ and $a \in \mathcal{A}$. Let K denote the number of times the triple s,a,t was encountered in total during the run of the algorithm. Define the random sequence X_i as follows. For $i \leq K$, let X_i be the indicator of whether s' was the next state when s,a,t was encountered the ith time and for $i > K$, let $X_i \sim \text{Bernoulli}(P(s'|s,a,t))$ be drawn i.i.d. By construction this is a sequence of i.i.d. Bernoulli random variables with mean $P(s'|s,a,t)$. Further the event

$$\bigcup_k \left\{ \left|\hat{P}_k(s'|s,a,t) - P(s'|s,a,t)\right| \geq \sqrt{\frac{2P(s'|s,a,t)}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n(s,a,t)) + \ln\frac{3S^2AH}{\delta'}\right)} + \frac{1}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3S^2AH}{\delta'}\right) \right\}$$

is contained in the event

$$\bigcup_i \left\{ |\hat{\mu}_i - \mu| \geq \sqrt{\frac{2\mu}{i}\left(2\operatorname{llnp}(i) + \ln\frac{3}{\delta'}\right)} + \frac{1}{i}\left(2\operatorname{llnp}(i) + \ln\frac{3S^2AH}{\delta'}\right) \right\}$$

whose probability can be bounded by $2\delta'/S^2/A/H$ using Lemma 47. The statement now follows by applying the union bound. □

Corollary 30. *For any $\delta' > 0$, it holds that $\mathbb{P}\left(\bigcup_{k=1}^{\infty} F_k^{L1}\right) \leq \delta'$*

Proof. Using the same argument as in the proof of Corollary 29 the statement follows from Lemma 48. □

Corollary 31. *It holds that*

$$\mathbb{P}\left(\bigcup_k F_k^N\right) \leq \delta'$$

Proof. Consider a fix $s \in \mathcal{S}, a \in \mathcal{A}, t \in [H]$. We define $\mathcal{F}_k$ to be the sigma-field induced by the first $k-1$ episodes and X_k as the indicator whether s, a, t was observed in episode k. The probability $w_{tk}(s,a)$ of whether $X_k = 1$ is $\mathcal{F}_k$ measurable and hence we can apply Lemma 49 with $W = \ln \frac{SAH}{\delta'}$ and obtain that $\mathbb{P}\left(\bigcup_k F_k^N\right) \leq \delta'$ after applying the union bound.

□

Corollary 32. *The total failure probability of the algorithm is bounded by $\mathbb{P}(F) \leq 8\delta' = \delta$.*

Proof. Statement follows directly from Corollary 28, Corollary 29, Corollary 30, Corollary 31 and the union bound. □

4.10.3 Nice Episodes

We now define the notion of *nice* . In nice episodes, all states either have low probability of occuring or the sum of probability of occuring in the previous episodes is large enough so that outside the failure event we can guarantee that

$$n_{tk}(s,a) \geq \frac{1}{4}\sum_{i<k} w_{ti}(s,a).$$

This allows us to then bound the number of nice episodes by the number of times terms of the form

$$\sum_{t=1}^{H}\sum_{s,a\in L_{tk}} w_{tk}(s,a)\sqrt{\frac{\text{llnp}(n_{tk}(s,a))+D}{n_{tk}(s,a)}}$$

can exceed a chosen threshold (see Lemma 35 below). In the next section, we will bound the optimality gap of an episode by terms of such form and use the results derived here to bound the number of nice episodes where the algorithm can follow a ϵ-suboptimal policy. Together with a bound on the number of non-nice episodes, we obtain the sample complexity of UBEV shown in Theorem 24.

Definition 2 (Nice Episodes)**.** *An episode k is* nice *if and only if for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $t \in [H]$ the following condition holds:*

$$w_{tk}(s,a) \leq w_{\min} \quad \vee \quad \frac{1}{4}\sum_{i<k} w_{ti}(s,a) \geq \ln\frac{SAH}{\delta'}.$$

We denote the set of all nice episodes by $N \subseteq \mathbb{N}$.

Lemma 33 (Properties of nice episodes)**.** *If an episode k is nice, i.e., $k \in N$, then on F^c (outside the failure event) for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $t \in [H]$ the following statement holds:*

$$w_{tk}(s,a) \leq w_{\min} \quad \vee \quad n_{tk}(s,a) \geq \frac{1}{4}\sum_{i<k} w_{ti}(s,a).$$

Proof. Since we consider the event $(F_k^N)^c$, it holds for all s, a, t triples with $w_{tk}(s,a) > w_{\min}$

$$n_{tk}(s,a) \geq \frac{1}{2}\sum_{i<k} w_{ti}(s,a) - \ln\frac{SAH}{\delta'} \geq \frac{1}{4}\sum_{i<k} w_{ti}(s,a)$$

for $k \in N$. □

Lemma 34 (Number of non-nice episodes)**.** *On the good event F^c, the number episodes that are not nice is at most*

$$\frac{84eS^2AH^3}{\epsilon}\ln\frac{SAH}{\delta'}.$$

Proof. If an episode k is not nice, then there is a (s,a,t) with $w_{tk}(s,a) > w_{\min}$ and $\sum_{i<k} w_{ti}(s,a) < 4\ln\frac{SAH}{\delta'}$. Since the sum on the left-hand side of this inequality increases by at least $w_{\min}$ when this happens and the right hand side stays constant, this situation can occur at most

$$\frac{4SAH}{w_{\min}}\ln\frac{SAH}{\delta'} = \frac{84eS^2AH^3}{\epsilon}\ln\frac{SAH}{\delta'}$$

times in total. □

Lemma 35 (Main Rate Lemma)**.** *Let $r \geq 1$ fix and $C > 0$ which can depend polynomially on the relevant quantities and $\epsilon' > 0$ and let $D \geq 1$ which can depend poly-logarithmically on the relevant quantities. Then*

$$\sum_t \sum_{s,a \in L_{tk}} w_{tk}(s,a)\left(\frac{C(\mathrm{llnp}(n_{tk}(s,a)) + D)}{n_{tk}(s,a)}\right)^{1/r} \leq \epsilon'$$

on all but at most

$$\frac{8CASH^r}{\epsilon'^r}\,\mathrm{polylog}(S, A, H, \delta^{-1}, \epsilon'^{-1}).$$

nice episodes.

Proof. Define

$$\begin{aligned}\Delta_k =& \sum_t \sum_{s,a \in L_{tk}} w_{tk}(s,a)\left(\frac{C(\mathrm{llnp}(n_{tk}(s,a)) + D)}{n_{tk}(s,a)}\right)^{1/r} \\ =& \sum_t \sum_{s,a \in L_{tk}} w_{tk}(s,a)^{1-\frac{1}{r}}\left(w_{tk}(s,a)\frac{C(\mathrm{llnp}(n_{tk}(s,a)) + D)}{n_{tk}(s,a)}\right)^{1/r}.\end{aligned}$$

We first bound using Hölder's inequality

$$\Delta_k \leq \left(\sum_t \sum_{s,a \in L_{tk}} \frac{CH^{r-1} w_{tk}(s,a)(\mathrm{llnp}(n_{tk}(s,a)) + D)}{n_{tk}(s,a)} \right)^{\frac{1}{r}}.$$

Using the property in Lemma 33 of nice episodes as well as the fact that $w_{tk}(s,a) \leq 1$ and $\sum_{i<k} w_{ti}(s,a) \geq 4 \ln \frac{SAH}{\delta'} \geq 4 \ln(2) \geq 2$, we bound

$$n_{tk}(s,a) \geq \frac{1}{4} \sum_{i<k} w_{ti}(s,a) \geq \frac{1}{8} \sum_{i \leq k} w_{ti}(s,a).$$

The function $\frac{\mathrm{llnp}(x)+D}{x}$ is monotonically decreasing in $x \geq 0$ since $D \geq 1$ (see Lemma 37). This allows us to bound

$$\begin{aligned}
\Delta_k^r \leq & \sum_t \sum_{s,a \in L_{tk}} \frac{CH^{r-1} w_{tk}(s,a)(\mathrm{llnp}(n_{tk}(s,a)) + D)}{n_{tk}(s,a)} \\
\leq & 8CH^{r-1} \sum_t \sum_{s,a \in L_{tk}} \frac{w_{tk}(s,a) \left(\mathrm{llnp} \left(\frac{1}{8} \sum_{i \leq k} w_{ti}(s,a) \right) + D \right)}{\sum_{i \leq k} w_{ti}(s,a)} \\
\leq & 8CH^{r-1} \sum_t \sum_{s,a \in L_{tk}} \frac{w_{tk}(s,a) \left(\mathrm{llnp} \left(\sum_{i \leq k} w_{ti}(s,a) \right) + D \right)}{\sum_{i \leq k} w_{ti}(s,a)}.
\end{aligned}$$

Assume now $\Delta_k > \epsilon'$. In this case the right-hand side of the inequality above is also larger than ϵ'^r and there is at least one (s,a,t) with $w_{tk}(s,a) > w_{\min}$ and

$$\begin{aligned}
& \frac{8CSAH^r \left(\mathrm{llnp} \left(\sum_{i \leq k} w_{ti}(s,a) \right) + D \right)}{\sum_{i \leq k} w_{ti}(s,a)} > \epsilon'^r \\
\Leftrightarrow & \frac{\mathrm{llnp} \left(\sum_{i \leq k} w_{ti}(s,a) \right) + D}{\sum_{i \leq k} w_{ti}(s,a)} > \frac{\epsilon'^r}{8CSAH^r}.
\end{aligned}$$

Let us denote $C' = \frac{8CASH^r}{\epsilon'^r}$. Since $\frac{\mathrm{llnp}(x)+D}{x}$ is monotonically decreasing and $x = C'^2 + 3C'D$ satisfies $\frac{\mathrm{llnp}(x)+D}{x} \leq \frac{\sqrt{x}+D}{x} \leq \frac{1}{C'}$, we know that if $\sum_{i \leq k} w_{ti}(s,a) \geq C'^2 + 3C'D$ then the above condition cannot be satisfied for s,a,t. Since each time the condition is satisfied, it holds that $w_{tk}(s,a) > w_{\min}$ and so $\sum_{i \leq k} w_{ti}(s,a)$ increases by at least $w_{\min}$, it can happen at most

$$m \leq \frac{ASH(C'^2 + 3C'D)}{w_{\min}}$$

times that $\Delta_k > \epsilon'$. Define $K = \{k : \Delta_k > \epsilon'\} \cap N$ and we know that $|K| \leq m$. Now we consider the sum

$$\begin{aligned}
\sum_{k \in K} \Delta_k^r \leq & \sum_{k \in K} 8CH^{r-1} \sum_t \sum_{s,a \in L_{tk}} \frac{w_{tk}(s,a) \left(\mathrm{llnp} \left(\sum_{i \leq k} w_{ti}(s,a) \right) + D \right)}{\sum_{i \leq k} w_{ti}(s,a)} \\
\leq & 8CH^{r-1} \left(\mathrm{llnp} \left(C'^2 + 3C'D \right) + D \right) \sum_t \sum_{s,a \in L_{tk}} \sum_{k \in K} \frac{w_{tk}(s,a)}{\sum_{i \leq k} w_{ti}(s,a) \mathbb{I}\{w_{ti}(s,a) \geq w_{\min}\}}
\end{aligned}$$

For every (s,a,t), we consider the sequence of $w_{ti}(s,a) \in [w_{\min}, 1]$ with $i \in I = \{i \in \mathbb{N} : w_{ti}(s,a) \geq w_{\min}\}$ and apply Lemma 36. This yields that

$$\sum_{k \in K} \frac{w_{tk}(s,a)}{\sum_{i \leq k} w_{ti}(s,a) \mathbb{I}\{w_{ti}(s,a) \geq w_{\min}\}} \leq 1 + \ln(m/w_{\min}) = \ln\left(\frac{me}{w_{\min}}\right)$$

and hence

$$\sum_{k \in K} \Delta_k^r \leq 8CASH^r \ln\left(\frac{me}{w_{\min}}\right) \left(\operatorname{llnp}\left(C'^2 + 3C'D\right) + D\right)$$

Since each element in K has to contribute at least ϵ'^r to this bound, we can conclude that

$$\sum_{k \in N} \mathbb{I}\{\Delta_k \geq \epsilon'\} \leq \sum_{k \in K} \mathbb{I}\{\Delta_k \geq \epsilon'\} \leq |K| \leq \frac{8CASH^r}{\epsilon'^r} \ln\left(\frac{me}{w_{\min}}\right) \left(\operatorname{llnp}\left(C'^2 + 3C'D\right) + D\right).$$

Since $\ln\left(\frac{me}{w_{\min}}\right) \left(\operatorname{llnp}\left(C'^2 + 3C'D\right) + D\right)$ is $\operatorname{polylog}(S, A, H, \delta^{-1}, \epsilon'^{-1})$, the proof is complete. □

Lemma 36. *Let a_i be a sequence taking values in $[a_{\min}, a_{\max}]$ with $a_{\max} \geq a_{\min} > 0$ and $m > 0$, then*

$$\sum_{k=1}^{m} \frac{a_k}{\sum_{i=1}^{k} a_i} \leq \ln\left(\frac{mea_{\max}}{a_{\min}}\right).$$

Proof. Let f be a step-function taking value a_i on $[i-1, i)$ for all i. We have $F(t) := \int_0^t f(x)dx = \sum_{i=1}^{t} a_i$. By the fundamental theorem of Calculus, we can bound

$$\begin{aligned}\sum_{k=1}^{m} \frac{a_k}{\sum_{i=1}^{k} a_i} =& \frac{a_1}{a_1} + \int_1^m \frac{f(x)}{F(x) - F(0)} dx = 1 + \ln F(m) - \ln F(1) \\ \leq& 1 + \ln(ma_{\max}) - \ln a_{\min} = \ln\left(\frac{mea_{\max}}{a_{\min}}\right),\end{aligned}$$

where the inequality follows from $a_1 \geq a_{\min}$ and $\sum_{i=1}^{m} a_i \leq m$. □

Lemma 37 (Properties of llnp)**.** *The following properties hold:*

1. llnp *is continuous and nondecreasing.*
2. $f(x) = \frac{\operatorname{llnp}(nx)+D}{x}$ *with $n \geq 0$ and $D \geq 1$ is monotonically decreasing on $\mathbb{R}_+$.*
3. $\operatorname{llnp}(xy) \leq \operatorname{llnp}(x) + \operatorname{llnp}(y) + 1$ *for all $x, y \geq 0$.*

Proof.

1. For $x \leq e$ we have $\operatorname{llnp}(x) = 0$ and for $x \geq e$ we have $\operatorname{llnp}(x) = \ln(\ln(x))$ which is continuous and monotonically increasing and $\lim_{x \searrow e} \ln(\ln(x)) = 0$.
2. The function llnp is continuous as well as $1/x$ on $\mathbb{R}_+$ and therefore so it f. Further, f is differentiable except at $x = e/n$. For $x \in [0, e/n)$, we have $f(x) = D/x$ with derivative $-D/x^2 < 0$. Hence f is monotonically decreasing on $x \in [0, e/n)$. For $x > e/n$, we have $f(x) = \frac{\ln(\ln(nx))+D}{x}$ with derivative

$$-\frac{D + \ln(\ln(nx))}{x^2} + \frac{1}{x^2 \ln(nx)} = \frac{1 - \ln(nx)(D + \ln(\ln(nx)))}{x^2 \ln(nx)}.$$

The denominator is always positive in this range so f is monotonically decreasing if and only if $\ln(nx)(D - \ln(\ln(nx))) \geq 1$. Using $D \geq 1$, we have $\ln(nx)(D + \ln(\ln(nx))) \geq 1(1 + 0) = 1$.

3. First note that for $xy \leq e^e$ we have $\operatorname{llnp}(xy) \leq 1 \leq \operatorname{llnp}(x) + \operatorname{llnp}(y) + 1$ and therfore the statement holds for $x, y \leq e$.
Then consider the case that $x, y \geq e$ and $\operatorname{llnp}(x) + \operatorname{llnp}(y) + 1 - \operatorname{llnp}(xy) = \ln\ln x + \ln\ln y + 1 - \ln(\ln(x) + \ln(y)) = -\ln(a+b) + 1 + \ln(a) + \ln(b)$ where $a = \ln x \geq 1$ and $b = \ln y \geq 1$. The function $g(a,b) = -\ln(a+b) + 1 + \ln(a) + \ln(b)$ is continuous and differentiable with $\frac{\partial g}{\partial a} = \frac{b}{a(a+b)} > 0$ and $\frac{\partial g}{\partial b} = \frac{a}{b(a+b)} > 0$. Therefore, g attains its minimum on $[1,\infty) \times [1,\infty)$ at $a = 1, b = 1$. Since $g(1,1) = 1 - \ln(2) \geq 0$, the statement also holds for $x, y \geq e$.
Finally consider the case where $x \leq e \leq y$. Then $\operatorname{llnp}(xy) \leq \operatorname{llnp}(ey) = \ln(1 + \ln y) \leq \ln\ln y + 1 \leq \operatorname{llnp}(x) + \operatorname{llnp}(y) + 1$. Due to symmetry this also holds for $y \leq e \leq x$.

□

4.10.4 Decomposition of Optimality Gap

In this section we decompose the optimality gap and then bound each term individually. Finally, the rate lemma presented in the previous section is used to determine a bound on the number of nice episodes where the optimality gap can be larger than ϵ. The decomposition in the following lemma is a simple approach to bound the optimality gap and eventually lead to the second term in the min in Theorem 24.

Lemma 38 (Optimality Gap Bound On Nice Episodes)**.** *On the good event F^c it holds that $V_1^\star(s_0) - V_1^{\pi_k}(s_0) \leq \epsilon$ on all nice episodes $k \in N$ except at most*

$$\frac{S^2AH^4}{\epsilon^2} \operatorname{polylog}(A, S, H, 1/\epsilon, 1/\delta)$$

episodes.

Proof. Using optimism of the algorithm shown in Lemma 45, we can bound

$$\begin{aligned}
&V_1^\star(s_0) - V_1^{\pi_k}(s_0) \\
\leq &|\tilde{V}_1^{\pi_k}(s_0) - V_1^{\pi_k}(s_0)| \\
\leq &\sum_{t=1}^{H} \sum_{s,a} w_{tk}(s,a) |(\tilde{P}_k(s,a,t) - P(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| + \sum_{t=1}^{H} \sum_{s,a} w_{tk}(s,a) |\tilde{r}_k(s,a,t) - r(s,a,t)| \\
\leq &\sum_{t=1}^{H} \sum_{s,a \in L_{tk}} w_{tk}(s,a) |(\tilde{P}_k(s,a,t) - P(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| + \sum_{t=1}^{H} \sum_{s,a \in L_{tk}} w_{tk}(s,a) |\tilde{r}_k(s,a,t) - r(s,a,t)| \\
&+ \sum_{t=1}^{H} \sum_{s,a \notin L_{tk}} w_{tk}(s,a) |(\tilde{P}_k(s,a,t) - P(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| + \sum_{t=1}^{H} \sum_{s,a \notin L_{tk}} w_{tk}(s,a) |\tilde{r}_k(s,a,t) - r(s,a,t)| \\
\leq &\sum_{t=1}^{H} \sum_{s,a \notin L_{tk}} H w_{\min} + \sum_{t=1}^{H} \sum_{s,a \in L_{tk}} w_{tk}(s,a) \Big[|(\tilde{P}_k(s,a,t) - \hat{P}_k(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| \\
&+ |(\hat{P}_k(s,a,t) - P(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| + |\tilde{r}_k(s,a,t) - r(s,a,t)| \Big]
\end{aligned}$$

The first term is bounded by $c_\epsilon \epsilon \leq \frac{\epsilon}{3}$. We now can use Lemma 40, Lemma 41 to bound the other terms by

$$\sum_{t=1}^{H} \sum_{s,a \in L_{tk}} w_{tk}(s,a) \sqrt{\frac{8(H + H\sqrt{S} + 2)^2}{n_{tk}(s,a)} \left(\operatorname{llnp}(n_{tk}(s,a)) + \frac{1}{2} \ln \frac{6SAH'}{\delta} \right)}.$$

We can then apply Lemma 35 with $r=2$, $C=8(H+H\sqrt{S}+2)^2$, $D=\frac{1}{2}\ln\frac{6SAH}{\delta'}$ (≥ 1 for any nontrivial setting) and $\epsilon'=2\epsilon/3$ to bound this term by $\frac{2\epsilon}{3}$ on all nice episodes except at most

$$\begin{aligned}&\frac{64(H+\sqrt{S}H+2)^2ASH^2 3^2}{4\epsilon^2}\operatorname{polylog}(A,S,H,1/\epsilon,1/\delta)\\ \leq&\frac{144(4+3H^2+4SH^2)ASH^2}{\epsilon^2}\operatorname{polylog}(A,S,H,1/\epsilon,1/\delta)=\frac{S^2AH^4}{\epsilon^2}\operatorname{polylog}(A,S,H,1/\epsilon,1/\delta)\end{aligned}$$

Hence $V_1^\star(s_0)-V_1^{\pi_k}(s_0)\leq\epsilon$ holds on all nice episodes except those. □

The lemma below is a refined version of the bound above which eventually leads to the first bound in $\min$ in Theorem 24.

Lemma 39 (Optimality Gap Bound Using Recursion)**.** *On the good event F^c it holds that $V_1^\star(s_0)-V_1^{\pi_k}(s_0)\leq\epsilon$ on all nice episodes $k\in N$ except at most*

$$\left(\frac{S^2AH^3}{\epsilon}+\frac{SAH^4}{\epsilon^2}\right)\operatorname{polylog}(A,S,H,1/\epsilon,1/\delta)$$

episodes.

Proof. We first consider the suboptimality in an arbitrary state $s\in\mathcal{S}$ at time $t\in[H]$. With $a=\pi_k(s,t)$ and shorthand $\psi_{tk}(s,a)=\sqrt{\frac{1}{n_{tk}(s,a)}\left(2\operatorname{llnp}((n_{tk}(s,a))+\ln\frac{18S^2AH}{\delta'}\right)}$ and $\psi'_{tk}(s,a)=\min\{\psi_{tk}(s,a),1\}$, we can bound

$$V_t^\star(s)-V_t^{\pi_k}(s)$$

Using optimism (Lemma 45):

$$\begin{aligned}\leq&\tilde{V}_t^{\pi_k}(s)-V_t^{\pi_k}(s)=\tilde{Q}_t^{\pi_k}(s,a)-Q_t^{\pi_k}(s,a)\\ =&\tilde{Q}_t^{\pi_k}(s,a)-\hat{P}_k(s,a,t)^\top\tilde{V}_{t+1}^{\pi_k}-\hat{r}(s,a,t)\\ &+\hat{r}(s,a,t)+\hat{P}_k(s,a,t)^\top\tilde{V}_{t+1}^{\pi_k}-P(s,a,t)^\top V_{t+1}^{\pi_k}-r(s,a,t)\end{aligned}$$

The expression $\tilde{Q}_t^{\pi_k}(s,a)-\hat{P}_k(s,a,t)^\top\tilde{V}_{t+1}^{\pi_k}-\hat{r}(s,a,t)$ is the amount of optimism and by Lemma 27 upper-bounded by $(H+1-t)\phi_k(s,a,t)\leq(H+1-t)\psi_{tk}(s,a)$ and $(H+1-t)$:

$$\leq(H+1-t)\psi'_{tk}(s,a)+\hat{r}(s,a,t)-r(s,a,t)+\hat{P}_k(s,a,t)^\top\tilde{V}_{t+1}^{\pi_k}-P(s,a,t)^\top V_{t+1}^{\pi_k}$$

Using now Lemma 41 and we can bound $\hat{r}(s,a,t)-r(s,a,t)\leq\min\{\sqrt{2}\psi_{tk}(s,a),1\}$ and get

$$\begin{aligned}\leq&(H+1-t+\sqrt{2})\psi'_{tk}(s,a)+\hat{P}_k(s,a,t)^\top\tilde{V}_{t+1}^{\pi_k}-P(s,a,t)^\top V_{t+1}^{\pi_k}\\ =&(H+1-t+\sqrt{2})\psi'_{tk}(s,a)+(\hat{P}_k-P)(s,a,t)^\top\tilde{V}_{t+1}^{\pi_k}+P(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k}-V_{t+1}^{\pi_k})\\ =&(H+1-t+\sqrt{2})\psi'_{tk}(s,a)+(\hat{P}_k-P)(s,a,t)^\top V_{t+1}^\star\\ &+(\hat{P}_k-P)(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k}-V_{t+1}^\star)+P(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k}-V_{t+1}^{\pi_k})\end{aligned}$$

Applying Lemma 42:

$$\begin{aligned}\leq&(3H+1-t+\sqrt{2})\psi'_{tk}(s,a)+(\hat{P}_k-P)(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k}-V_{t+1}^\star)+P(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k}-V_{t+1}^{\pi_k})\\ \leq&6H\psi'_{tk}(s,a)+(\hat{P}_k-P)(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k}-V_{t+1}^\star)+P(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k}-V_{t+1}^{\pi_k})\end{aligned}$$

Applying the recursive lemma for the lower-order term (Lemma 43), the term $(\hat{P}_k - P)(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k} - V_{t+1}^\star)$ can be bounded recursively. Alternatively, it can be bounded by H using the fact that $0 \le (\tilde{V}_{t+1}^{\pi_k} - V_{t+1}^\star) \le H$ and that $(\hat{P}_k - P)$ is a difference of probability vectors. This gives :

$$\le 6H\psi'_{tk}(s,a) + \min\{3H^2S\psi_{tk}(s,a)^2, H\} + \left(1 + \frac{1}{H}\right) P(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k} - V_{t+1}^{\pi_k})$$

We therefore have obtained a recursive bound that we can now roll out for $t = 1, \dots H$ until $\tilde{V}_{H+1}^{\pi_k}(s) - V_{H+1}^{\pi_k}(s) = 0$ and get that

$$\begin{aligned}
&V_1^\star(s_0) - V_1^{\pi_k}(s_0) \\
\le &\sum_{t=1}^{H} \left(1 + \frac{1}{H}\right)^{t-1} \sum_{s,a \in \mathcal{S}\times\mathcal{A}} w_{tk}(s,a) \left[6H\psi'_{tk}(s,a) + \min\{3H^2S\psi_{tk}(s,a)^2, H\}\right] \\
\le &6He \sum_{t=1}^{H} \sum_{s,a\in\mathcal{S}\times\mathcal{A}} w_{tk}(s,a)\psi'_{tk}(s,a) + e\sum_{t=1}^{H}\sum_{s,a\in\mathcal{S}\times\mathcal{A}} w_{tk}(s,a)\min\{H, 3H^2S\psi_{tk}(s,a)^2\}
\end{aligned}$$

Splitting now in (s,a,t) that are likely to be visited or not:

$$\begin{aligned}
&\le 7He \sum_{t=1}^{H} \sum_{s,a\notin L_{tk}} w_{tk}(s,a) + \sum_{t=1}^{H}\sum_{s,a\in L_{tk}} w_{tk}(s,a)[6He\psi_{tk}(s,a) + 3eH^2S\psi_{tk}(s,a)^2] \\
&\le 7eSH^2 w_{\min} + 6He\sum_{t=1}^{H}\sum_{s,a\in L_{tk}} w_{tk}(s,a)\psi_{tk}(s,a) + 3e\sum_{t=1}^{H}\sum_{s,a\in L_{tk}} w_{tk}(s,a)H^2S\psi_{tk}(s,a)^2
\end{aligned}$$

We now apply Lemma 35 with $r = 2$, $C = 2(6He)^2$, an approproate D and $\epsilon' = \epsilon/3$ to bound the second term by $\frac{\epsilon}{3}$ on all nice episodes except at most

$$\begin{aligned}
&8\frac{18(6He)^2ASH^2}{\epsilon^2}\operatorname{polylog}(A,S,H,1/\epsilon,1/\delta) \\
\le &\frac{72^2e^2ASH^4}{\epsilon^2}\operatorname{polylog}(A,S,H,1/\epsilon,1/\delta).
\end{aligned}$$

Similarly, we apply Lemma 35 with $r = 1$, $C = 6eH^2S$, an approproate D and $\epsilon' = \epsilon/3$ to bound the third term by $\frac{\epsilon}{3}$ on all nice episodes except at most

$$\begin{aligned}
&8\frac{3(6eH^2S)ASH}{\epsilon}\operatorname{polylog}(A,S,H,1/\epsilon,1/\delta) \\
\le &\frac{144eAS^2H^3}{\epsilon}\operatorname{polylog}(A,S,H,1/\epsilon,1/\delta).
\end{aligned}$$

Finally, the first term is bounded by $7eSH^2w_{\min} = 7ec_\epsilon\epsilon \le \frac{\epsilon}{3}$ since $c_\epsilon \le \frac{1}{21e}$. □

Lemma 40 (Algorithm Learns Fast Enough)**.** *It holds for all $s \in \mathcal{S}, a \in \mathcal{A}$ and $t \in [H]$*

$$|(\hat{P}_k(s,a,t) - \tilde{P}_k(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| \le \sqrt{\frac{2H^2}{n_{tk}(s,a)}\left(\operatorname{llnp}(n_{tk}(s,a)) + \frac{1}{2}\ln\frac{3SAH}{\delta'}\right)}.$$

Proof. Using the definition of the constraint in the planning step of the algorithm shown in Lemma 27 we can bound

$$\begin{aligned}|(\hat{P}_k(s,a,t) - \tilde{P}_k(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| \leq& \sqrt{\frac{H^2}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3SAH}{\delta'}\right)}.\\ \leq& \sqrt{\frac{2H^2}{n_{tk}(s,a)}\left(\operatorname{llnp}(n_{tk}(s,a)) + \frac{1}{2}\ln\frac{3SAH}{\delta'}\right)}.\end{aligned}$$

□

Lemma 41 (Basic Decompsition Bound)**.** *On the good event F^c it holds for all $s \in \mathcal{S}, a \in \mathcal{A}$ and $t \in [H]$*

$$\begin{aligned}|(\hat{P}_k(s,a,t) - P(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| \leq& \sqrt{\frac{8H^2S}{n_{tk}(s,a)}\left(\operatorname{llnp}(n_{tk}(s,a)) + \frac{1}{2}\ln\frac{6SAH}{\delta'}\right)}\\ |\tilde{r}_k(s,a,t) - r(s,a,t)| \leq& \sqrt{\frac{4}{n_{tk}(s,a)}\left(\operatorname{llnp}(n_{tk}(s,a)) + \frac{1}{2}\ln\frac{3SAH}{\delta'}\right)}.\end{aligned}$$

Proof. On the good event $(F_k^{L1})c$ we have using Hölder's inequality

$$\begin{aligned}|(\hat{P}_k(s,a,t) - P(s,a,t))^\top \tilde{V}_{t+1}^{\pi_k}| \leq& \|\hat{P}_k(s,a,t) - P(s,a,t))\|_1 \|\tilde{V}_{t+1}^{\pi_k}\|_\infty\\ \leq& H\sqrt{\frac{4}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3SAH(2^S-2)}{\delta'}\right)}\\ \leq& \sqrt{\frac{8H^2S}{n_{tk}(s,a)}\left(\operatorname{llnp}(n_{tk}(s,a)) + \frac{1}{2}\ln\frac{6SAH}{\delta'}\right)}.\end{aligned}$$

Further, on $(F_k^R)^c$ we have

$$\begin{aligned}|\tilde{r}_k(s,a,t) - r(s,a,t)| \leq& |\tilde{r}_k(s,a,t) - r(s,a,t)| + |\tilde{r}_k(s,a,t) - \hat{r}(s,a,t)|\\ \leq& 2\sqrt{\frac{1}{n_{tk}(s,a)}\left(2\operatorname{llnp}(n_{tk}(s,a)) + \ln\frac{3SAH}{\delta'}\right)}\end{aligned}$$

□

Lemma 42 (Fixed V Term Confidence Bound)**.** *On the good event F^c it holds for all $s \in \mathcal{S}, a \in \mathcal{A}$ and $t \in [H]$*

$$|(\hat{P}_k(s,a,t) - P(s,a,t))^\top V_{t+1}^\star| \leq \sqrt{\frac{2H^2}{n_{tk}(s,a)}\left(\operatorname{llnp} n_{tk}(s,a) + \frac{1}{2}\ln\frac{3SAH}{\delta'}\right)}$$

Proof. Since we consider the event $(F_k^V)^c$, we can bound

$$|(\hat{P}_k(s,a,t) - P(s,a,t))^\top V_{t+1}^\star| \leq \sqrt{\frac{2H^2}{n_{tk}(s,a)}\left(\operatorname{llnp} n_{tk}(s,a) + \frac{1}{2}\ln\frac{3SAH}{\delta'}\right)}$$

□

Lemma 43 (Lower Order Term Recursion)**.** *On the good event F^c it holds for all $s \in \mathcal{S}, a \in \mathcal{A}$ and $t \in [H]$*

$$(\hat{P}_k(s,a,t) - P(s,a,t))^\top(\tilde{V}_{t+1}^{\pi_k} - V_{t+1}^\star) \leq \frac{1}{H} P(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k} - V_{t+1}^{\pi_k}) + 3H^2 S\psi_{tk}(s,a)^2$$

where

$$\psi_{tk}(s,a) = \sqrt{\frac{1}{n_{tk}(s,a)}\left(2\,\mathrm{llnp}((n_{tk}(s,a)) + \ln\frac{18S^2AH}{\delta'}\right)}.$$

Proof. We use the short-hand notation

$$\psi_{tk}(s,a) = \sqrt{\frac{1}{n_{tk}(s,a)}\left(2\,\mathrm{llnp}((n_{tk}(s,a)) + \ln\frac{18S^2AH}{\delta'}\right)}$$

and apply the definition of the event F^c to bound

$$\begin{aligned}
&(\hat{P}_k(s,a,t) - P(s,a,t))^\top(\tilde{V}_{t+1}^{\pi_k} - V_{t+1}^\star) \\
\leq &\sum_{s'\in\mathcal{S}}\left[\sqrt{2P(s'|s,a,t)}\psi_{tk}(s,a)(\tilde{V}_{t+1}^{\pi_k}(s') - V_{t+1}^\star(s')) + \psi_{tk}(s,a)^2(\tilde{V}_{t+1}^{\pi_k}(s') - V_{t+1}^\star(s'))\right]
\end{aligned}$$

Using that the range of value functions is H:

$$\begin{aligned}
\leq &SH\psi_{tk}(s,a)^2 + \sum_{s'\in\mathcal{S}}\psi_{tk}(s,a)\sqrt{2P(s'|s,a,t)}(\tilde{V}_{t+1}^{\pi_k}(s') - V_{t+1}^\star(s')) \\
= &SH\psi_{tk}(s,a)^2 + \sum_{s'\in\mathcal{S}}\psi_{tk}(s,a)\sqrt{\frac{2}{P(s'|s,a,t)}}P(s'|s,a,t)(\tilde{V}_{t+1}^{\pi_k}(s') - V_{t+1}^\star(s'))
\end{aligned}$$

Using the optimality of $V^\star$:

$$\leq SH\psi_{tk}(s,a)^2 + \sum_{s'\in\mathcal{S}}\psi_{tk}(s,a)\sqrt{\frac{2}{P(s'|s,a,t)}}P(s'|s,a,t)(\tilde{V}_{t+1}^{\pi_k}(s') - V_{t+1}^{\pi_k}(s'))$$

Splitting the sum based on magnitude of individual probabilities:

$$\begin{aligned}
\leq &SH\psi_{tk}(s,a)^2 + \frac{1}{H}P(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k} - V_{t+1}^{\pi_k}) \\
&+ \sum_{s'\in\mathcal{S}}\psi_{tk}(s,a)\sqrt{\frac{2}{P(s'|s,a,t)}}P(s'|s,a,t)(\tilde{V}_{t+1}^{\pi_k}(s') - V_{t+1}^{\pi_k}(s'))\mathbb{I}\{\sqrt{P(s'|s,a,t)} < \sqrt{2}H\psi_{tk}(s,a)\}
\end{aligned}$$

$$\leq SH\psi_{tk}(s,a)^2 + \frac{1}{H}P(s,a,t)^\top(\tilde{V}_{t+1}^{\pi_k} - V_{t+1}^{\pi_k}) + 2H^2S\psi_{tk}(s,a)^2$$

□

4.10.5 Useful Lemmas

Lemma 44 (Value Difference Lemma)**.** *For any two MDPs M' and M'' with rewards r' and r'' and transition probabilities P' and P'', the difference in values with respect to the same policy π can be written as*

$$V_i'(s) - V_i''(s) = \mathbb{E}''\left[\sum_{t=i}^{H}(r'(s_t,a_t,t) - r''(s_t,a_t,t))\middle| s_i = s\right] + \mathbb{E}''\left[\sum_{t=i}^{H}(P'(s_t,a_t,t) - P''(s_t,a_t,t))^\top V_{t+1}'\middle| s_i = s\right]$$

where $V_{H+1}' = V_{H+1}'' = \vec{0}$ and the expectation $\mathbb{E}'$ is taken w.r.t to P' and π and $\mathbb{E}''$ w.r.t. P'' and π.

Proof. For $i = H+1$ the statement is trivially true. We assume now it holds for $i+1$ and show it holds also for i. Using only this induction hypothesis and basic algebra, we can write

$$\begin{aligned}
&V_i'(s) - V_i''(s)\\
=&\mathbb{E}_\pi[r'(s_i,a_i,i) + {V_{i+1}'}^\top P'(s_i,a_i,i) - r''(s_i,a_i,i) - {V_{i+1}''}^\top P''(s_i,a_i,i)|s_i = s]\\
=&\mathbb{E}_\pi[r'(s_i,a_i,i) - r''(s_i,a_i,i)|s_i = s] + \mathbb{E}_\pi\left[\sum_{s'\in\mathcal{S}} V_{i+1}'(s')(P'(s'|s_i,a_i,i) - P''(s'|s_i,a_i,i))\middle| s_i = s\right]\\
&+ \mathbb{E}_\pi\left[\sum_{s'\in\mathcal{S}} P''(s'|s_i,a_i,i)(V_{i+1}'(s') - V_{i+1}''(s'))\middle| s_i = s\right]\\
=&\mathbb{E}_\pi[r'(s_i,a_i,i) - r''(s_i,a_i,i)|s_i = s] + \mathbb{E}_\pi\left[\sum_{s'\in\mathcal{S}} V_{i+1}'(s')(P'(s'|s_i,a_i,i) - P''(s'|s_i,a_i,i))\middle| s_i = s\right]\\
&+ \mathbb{E}''\left[V_{i+1}'(s_{i+1}) - V_{i+1}''(s_{i+1}))\middle| s_i = s\right]\\
=&\mathbb{E}_\pi[r'(s_i,a_i,i) - r''(s_i,a_i,i)|s_i = s] + \mathbb{E}_\pi\left[\sum_{s'\in\mathcal{S}} V_{i+1}'(s')(P'(s'|s_i,a_i,i) - P''(s'|s_i,a_i,i))\middle| s_i = s\right]\\
&+ \mathbb{E}''\left[\mathbb{E}''\left[\sum_{t=i+1}^{H}(r'(s_t,a_t,t) - r''(s_t,a_t,t))\middle| s_{i+1}\right] + \mathbb{E}''\left[\sum_{t=i+1}^{H}(P'(s_t,a_t,t) - P''(s_t,a_t,t))^\top V_{t+1}'\middle| s_{i+1}\right]\middle| s_i = s\right]\\
=&\mathbb{E}''\left[\sum_{t=i}^{H}(r'(s_t,a_t,t) - r''(s_t,a_t,t))\middle| s_i = s\right] + \mathbb{E}''\left[\sum_{t=i}^{H}(P'(s_t,a_t,t) - P''(s_t,a_t,t))^\top V_{t+1}'\middle| s_i = s\right]
\end{aligned}$$

where the last equality follows from law of total expectation □

Lemma 45 (Algorithm ensures optimism)**.** *On the good event F^c it holds that for all episodes k, $t \in [H]$, $s \in \mathcal{S}$ that*

$$V_t^{\pi_k}(s) \le V_t^\star(s) \le \tilde{V}_t^{\pi_k}(s).$$

Proof. The first inequality follows simply from the definition of the optimal value function $V^\star$.

Since all outcome we consider are in the event $(F_k^V)^c$, we know that the true transition probabilities P, the optimal policy $\pi^\star$ and optimal policy $V^\star$ are a feasible solution for the optimistic planning problem in Lemma 27 that UBEV solves. It therefore follows immediately that $p_0^\top \tilde{V}_1^{\pi_k} \geq p_0^\top V_1^\star$. □

4.11 General Concentration Bounds

Lemma 46. *Let $X_1, X_2, \ldots$ be a martingale difference sequence adapted to filtration $\{\mathcal{F}_t\}_{t=1}^\infty$ with X_t conditionally σ^2-subgaussian so that $\mathbb{E}[\exp(\lambda(X_t - \mu))|\mathcal{F}_{t-1}] \leq \exp(\lambda^2\sigma^2/2)$ almost surely for all $\lambda \in \mathbb{R}$. Then with $\hat{\mu}_t = \frac{1}{t}\sum_{i=1}^t X_i$ we have for all $\delta \in (0, 1]$*

$$\mathbb{P}\left(\exists t : |\hat{\mu}_t - \mu| \geq \sqrt{\frac{4\sigma^2}{t}\left(2\operatorname{llnp}(t) + \ln\frac{3}{\delta}\right)}\right) \leq 2\delta\,.$$

Proof. Let $S_t = \sum_{s=1}^t (X_s - \mu)$. Then

$$\begin{aligned}
&\mathbb{P}\left(\exists t : \hat{\mu}_t - \mu \geq \sqrt{\frac{4\sigma^2}{t}\left(2\operatorname{llnp}(t) + \ln\frac{3}{\delta}\right)}\right) \\
\leq &\mathbb{P}\left(\exists t : S_t \geq \sqrt{4\sigma^2 t\left(2\operatorname{llnp}(t) + \ln\frac{3}{\delta}\right)}\right) \\
\leq &\sum_{k=0}^\infty \mathbb{P}\left(\exists t \in [2^k, 2^{k+1}] : S_t \geq \sqrt{4\sigma^2 t\left(2\operatorname{llnp}(t) + \ln\frac{3}{\delta}\right)}\right) \\
\leq &\sum_{k=0}^\infty \mathbb{P}\left(\exists t \leq 2^{k+1} : S_t \geq \sqrt{2\sigma^2 2^{k+1}\left(2\operatorname{llnp}(2^k) + \ln\frac{3}{\delta}\right)}\right)
\end{aligned}$$

We now consider $M_t = \exp(\lambda S_t)$ for $\lambda > 0$ which is a nonnegative sub-martingale and use the short-hand $f = \sqrt{2\sigma^2 2^{k+1}\left(2\operatorname{llnp}(2^k) + \ln\frac{3}{\delta}\right)}$. Then by Doob's maximal inequality for nonnegative submartingales

$$\mathbb{P}\left(\exists t \leq 2^{k+1} : S_t \geq f\right) = \mathbb{P}\left(\max_{t \leq 2^{k+1}} M_t \geq \exp(\lambda f)\right) \leq \frac{\mathbb{E}[M_{2^{k+1}}]}{\exp(\lambda f)} \leq \exp\left(2^{k+1}\frac{\lambda^2\sigma^2}{2} - \lambda f\right).$$

Choosing the optimal $\lambda = \frac{f}{\sigma^2 2^{k+1}}$ we obtain the bound

$$\begin{aligned}
\mathbb{P}\left(\exists t \leq 2^{k+1} : S_t \geq f\right) \leq &\exp\left(-\frac{f^2}{2^{k+2}\sigma^2}\right) = \exp\left(-2\operatorname{llnp}(2^k) - \ln\frac{3}{\delta}\right) = \frac{\delta}{3}\exp\left(-2\operatorname{llnp}(2^k)\right) \\
= &\frac{\delta}{3}\exp\left(-\max\{0, 2\ln\max\{0, \ln 2^k\}\}\right) = \frac{\delta}{3}\min\left\{1, (k\ln 2)^{-2}\right\} \\
\leq &\frac{\delta}{3}\min\left\{1, \frac{1}{k^2\ln 2}\right\}.
\end{aligned} \tag{4.9}$$

Plugging this back in the bound from above, we get

$$\begin{aligned}\mathbb{P}\left(\exists t : \hat{\mu}_t - \mu \geq \sqrt{\frac{4\sigma^2}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right)}\right) \leq& \frac{\delta}{3}\sum_{k=0}^{\infty}\min\left\{1, \frac{1}{k^2\ln(2)}\right\} \\ =& \delta\,\frac{1}{3}\left(\frac{\pi^2}{6\ln 2} + 2 - 1/\ln(2)\right) \leq \delta\,. \end{aligned} \tag{4.10}$$

For the other side, the argument follows completely analogously with

$$\begin{aligned}\mathbb{P}\left(\exists t \leq 2^{k+1} : S_t \leq -f\right) =& \mathbb{P}\left(\exists t \leq 2^{k+1} : -S_t \geq f\right) \\ =& \mathbb{P}\left(\max_{t\leq 2^{k+1}} \exp(-\lambda S_t) \geq \exp(\lambda f)\right) \\ \leq& \frac{\mathbb{E}[\exp(-\lambda S_{2^{k+1}})]}{\exp(\lambda f)} \leq \exp\left(2^{k+1}\frac{\lambda^2\sigma^2}{2} - \lambda f\right).\end{aligned}$$

□

Lemma 47. *Let $X_1, X_2, \ldots$ be a sequence of Bernoulli random variables with bias $\mu \in [0,1]$. Then for all $\delta \in (0,1]$*

$$\mathbb{P}\left(\exists t : |\hat{\mu}_t - \mu| \geq \sqrt{\frac{2\mu}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right)} + \frac{1}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right)\right) \leq 2\delta$$

Proof.

$$\begin{aligned}&\mathbb{P}\left(\exists t : \hat{\mu}_t - \mu \geq \sqrt{\frac{2\mu}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right)} + \frac{1}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right)\right) \\ =&\mathbb{P}\left(\exists t : S_t \geq \sqrt{2\mu t\left(2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right)} + 2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right) \\ \leq& \sum_{k=0}^{\infty}\mathbb{P}\left(\exists t \leq 2^{k+1} : S_t \geq \sqrt{2\mu 2^k\left(2\,\mathrm{llnp}(2^k) + \ln\frac{3}{\delta}\right)} + 2\,\mathrm{llnp}(2^k) + \ln\frac{3}{\delta}\right)\end{aligned}$$

Let $g = 2\,\mathrm{llnp}(2^k) + \ln\frac{3}{\delta}$ and $f = \sqrt{2^{k+1}\mu g} + g$. Further define $S_t = \sum_{i=1}^{t} X_i - t\mu$ and $M_t = \exp(\lambda S_t)$ which is by construction a nonnegative submartingale. Applying Doob's maximal inequality for nonnegative submartingales, we bound

$$\mathbb{P}\left(\exists t \leq 2^{k+1} : S_t \geq f\right) = \mathbb{P}\left(\max_{i\leq 2^{k+1}} M_i \geq \exp(\lambda f)\right) \leq \frac{\mathbb{E}[M_{2^{k+1}}]}{\exp(\lambda f)} = \exp\left(\ln\mathbb{E}[M_{2^{k+1}}] - \lambda f\right).$$

Since this holds for all $\lambda \in \mathbb{R}$, we can bound

$$\mathbb{P}\left(\exists t \leq 2^{k+1} : S_t \geq f\right) \leq \exp\left(-\sup_{\lambda\in\mathbb{R}}\left(\lambda f - \ln\mathbb{E}[M_{2^{k+1}}]\right)\right)$$

and using Corollary 2.11 by Boucheron, Lugosi, and Massart (2013) (see also note below proof of Corollary 2.11) bound that by

$$\exp\left(-\frac{f^2}{2(2^{k+1}\mu + f/3)}\right)$$

We now argue that this quantity can be upper-bounded by $\exp(-g)$. This is equivalent to

$$\begin{aligned}
-\frac{f^2}{2(2^{k+1}\mu + f/3)} &\leq -g \\
f^2 &\geq 2g(2^{k+1}\mu + f/3) = \frac{2}{3}gf + \frac{2^{k+2}}{3}\mu g \\
g^2 + 2\sqrt{2^{k+1}\mu g}g + 2^{k+1}\mu g &\geq \frac{2}{3}g^2 + \frac{2}{3}\sqrt{2^{k+1}\mu g}g + \frac{2^{k+2}}{3}\mu g \\
\frac{1}{3}g^2 + \frac{4}{3}\sqrt{2^{k+1}\mu g}g + \frac{1}{3}2^{k+1}\mu g &\geq 0.
\end{aligned}$$

Each line is an equivalent inequality since $g, f \geq 0$ and each term on the left in the final inequality is nonnegative. Hence, we get $\mathbb{P}\left(\exists t \leq 2^{k+1} : S_t \geq f\right) \leq \exp(-g)$. Following now the arguments from the proof of Lemma 46 in Equations (4.9)–(4.10), we obtain that

$$\mathbb{P}\left(\exists t : \hat{\mu}_t - \mu \geq \sqrt{\frac{2\mu}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right)} + \frac{1}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3}{\delta}\right)\right) \leq \delta.$$

For the other direction, we proceed analogously to above and arrive at

$$\mathbb{P}\left(\exists t \leq 2^{k+1} : -S_t \geq f\right) \leq \exp\left(-\sup_{\lambda \in \mathbb{R}}\left(-\lambda f - \ln \mathbb{E}[M_{2^{k+1}}]\right)\right)$$

which we bound similarly to above by

$$\exp\left(-\frac{f^2}{2(2^{k+1}\mu - f/3)}\right) \leq \exp\left(-\frac{f^2}{2(2^{k+1}\mu + f/3)}\right) \leq \exp(-g).$$

□

Lemma 48 (Uniform L1-Deviation Bound for Empirical Distribution)**.** *Let $X_1, X_2, \ldots$ be a sequence of i.i.d. categorical variables on $[U]$ with distribution P. Then for all $\delta \in (0, 1]$*

$$\mathbb{P}\left(\exists t \,:\, \|\hat{P}_t - P\|_1 \geq \sqrt{\frac{4}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3(2^U - 2)}{\delta}\right)}\right) \leq \delta$$

where $\hat{P}_t$ is the empirical distribution based on samples $X_1 \ldots X_t$.

Proof. We use the identity $\|Q - P\|_1 = 2\max_{B \subseteq \mathcal{B}} Q(B) - P(B)$ which holds for all distributions P, Q

defined on the finite set $\mathcal{B}$ to bound

$$\begin{aligned}
&\mathbb{P}\left(\exists t \,:\, \|\hat{P}_t - P\|_1 \geq \sqrt{\frac{4}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3(2^U-2)}{\delta}\right)}\right)\\
=&\mathbb{P}\left(\max_{t,B\subseteq[U]} \hat{P}_t(B) - P(B) \geq \frac{1}{2}\sqrt{\frac{4}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3(2^U-2)}{\delta}\right)}\right)\\
\leq& \sum_{B\subseteq[U]} \mathbb{P}\left(\max_t \hat{P}_t(B) - P(B) \geq \sqrt{\frac{1}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3(2^U-2)}{\delta}\right)}\right).
\end{aligned}$$

Define now $S_t = \sum_{i=1}^t \mathbb{I}\{X_1 \in B\} - tP(B)$ which is a martingale sequence. Then the last line above is equivalent to

$$\begin{aligned}
&\sum_{B\subseteq[U]} \mathbb{P}\left(\max_t S_t \geq \sqrt{t\left(2\,\mathrm{llnp}(t) + \ln\frac{3(2^U-2)}{\delta}\right)}\right)\\
\leq& \sum_{B\subseteq[U]} \mathbb{P}\left(\max_{k\in\mathbb{N},t\in[2^k,2^{k+1}]} S_t \geq \sqrt{t\left(2\,\mathrm{llnp}(t) + \ln\frac{3(2^U-2)}{\delta}\right)}\right)\\
\leq& \sum_{B\subseteq[U]} \sum_{k=0}^{\infty} \mathbb{P}\left(\max_{t\in[2^k,2^{k+1}]} S_t \geq \sqrt{t\left(2\,\mathrm{llnp}(t) + \ln\frac{3(2^U-2)}{\delta}\right)}\right)\\
\leq& \sum_{B\subseteq[U]} \sum_{k=0}^{\infty} \mathbb{P}\left(\max_{t\leq 2^{k+1}} S_t \geq \sqrt{2^k\left(2\,\mathrm{llnp}(2^k) + \ln\frac{3(2^U-2)}{\delta}\right)}\right)\\
=& \sum_{B\subseteq[U]} \sum_{k=0}^{\infty} \mathbb{P}\left(\max_{t\leq 2^{k+1}} \exp(\lambda S_t) \geq \exp(\lambda f)\right)\\
=& \sum_{B\subseteq[U],B\neq\emptyset,B\neq[U]} \sum_{k=0}^{\infty} \mathbb{P}\left(\max_{t\leq 2^{k+1}} \exp(\lambda S_t) \geq \exp(\lambda f)\right)
\end{aligned}$$

where $f = \sqrt{2^k\left(2\,\mathrm{llnp}(2^k) + \ln\frac{3(2^U-2)}{\delta}\right)}$ and $\lambda \in \mathbb{R}$ and the last equality follows from the fact that for $B = \emptyset$ and $B = [U]$ the difference between the distributions has to be 0. Since $\mathbb{I}\{X_1 \in B\} - tP(B)$ is a centered Bernoulli variable it is $1/2$-subgaussian and so S_t satisfies $\mathbb{E}[\exp(\lambda S_t)] \leq \exp(\lambda^2 t/8)]$. Since S_t is a martingale, $\exp(\lambda S_t)$ is a nonnegative sub-martingale and we can apply the maximal inequality to bound

$$\mathbb{P}\left(\max_{t\leq 2^{k+1}} \exp(\lambda S_t) \geq \exp(\lambda f)\right) \leq \exp\left(\frac{1}{8}\lambda^2 2^{k+1} - \lambda f\right).$$

Choosing $\lambda = \frac{4f}{2^{k+1}}$, we get $\mathbb{P}\left(\max_{t\leq 2^{k+1}} \exp(\lambda S_t) \geq \exp(\lambda f)\right) \leq \exp\left(-\frac{f^2}{2^k}\right)$. Hence, using the same steps as in the proof of Lemma 46, we get $\mathbb{P}\left(\max_{t\leq 2^{k+1}} \exp(\lambda S_t) \geq \exp(\lambda f)\right) \leq \frac{\delta}{3(2^{[U]}-2)} \min\left\{1, \frac{1}{k^2 \ln 2}\right\}$

and then

$$\mathbb{P}\left(\exists t \,:\, \|\hat{P}_t - P\|_1 \geq \sqrt{\frac{4}{t}\left(2\,\mathrm{llnp}(t) + \ln\frac{3(2^U-2)}{\delta}\right)}\right)$$
$$\leq \sum_{B\subseteq[U],B\neq\emptyset,B\neq[U]} \frac{\delta}{3(2^{[U]}-2)} \sum_{k=0}^{\infty} \min\left\{1, \frac{1}{k^2 \ln 2}\right\} \leq \sum_{B\subseteq[U],B\neq\emptyset,B\neq[U]} \frac{\delta}{2^{[U]}-2} = \delta.$$

□

Lemma 49. *Let $\mathcal{F}_i$ for $i = 1\ldots$ be a filtration and $X_1, \ldots X_n$ be a sequence of Bernoulli random variables with $\mathbb{P}(X_i = 1|\mathcal{F}_{i-1}) = P_i$ with P_i being $\mathcal{F}_{i-1}$-measurable and X_i being $\mathcal{F}_i$ measurable. It holds that*

$$\mathbb{P}\left(\exists n : \sum_{t=1}^{n} X_t < \sum_{t=1}^{n} P_t/2 - W\right) \leq e^{-W}$$

Proof. $P_t - X_t$ is a Martingale difference sequence with respect to the filtration $\mathcal{F}_t$. Since X_t is nonnegative and has finite second moment, we have for any $\lambda > 0$ that $\mathbb{E}\left[e^{-\lambda(X_t-P_t)}|\mathcal{F}_{t-1}\right] \leq e^{\lambda^2 P_t/2}$ (Exercise 2.9, Boucheron, Lugosi, and Massart (2013)). Hence, we have

$$\mathbb{E}\left[e^{\lambda(P_t-X_t)-\lambda^2 P_t/2}|\mathcal{F}_{t-1}\right] \leq 1$$

and by setting $\lambda = 1$, we see that

$$M_n = e^{\sum_{t=1}^{n}(-X_t+P_t/2)}$$

is a supermartingale. It hence holds by Markov's inequality

$$\mathbb{P}\left(\sum_{t=1}^{n}(-X_t + P_t/2) \geq W\right) = \mathbb{P}\left(M_n \geq e^W\right) \leq e^{-W}\mathbb{E}[M_n] \leq e^{-W}$$

wich gives us the derised result

$$\mathbb{P}\left(\sum_{t=1}^{n} X_t \leq \sum_{t=1}^{n} P_t/2 - W\right) \leq e^{-W}$$

for a fixed n. We define now the stopping time $\tau = \min\{t \in \mathbb{N} \,:\, M_t > e^W\}$ and the sequence $\tau_n = \min\{t \in \mathbb{N} \,:\, M_t > e^W \vee t \geq n\}$. Applying the convergence theorem for nonnegative supermartingales (Theorem 5.2.9 in Durrett (2010)), we get that $\lim_{t\to\infty} M_t$ is well-defined almost surely. Therefore, M_τ is well-defined even when $\tau = \infty$. By the optional stopping theorem for nonnegative supermartingales (Theorem 5.7.6 by Durrett (2010)), we have $\mathbb{E}[M_{\tau_n}] \leq \mathbb{E}[M_0] \leq 1$ for all n and applying Fatou's lemma, we obtain $\mathbb{E}[M_\tau] = \mathbb{E}[\lim_{n\to\infty} M_{\tau_n}] \leq \liminf_{n\to\infty} \mathbb{E}[M_{\tau_n}] \leq 1$. Using Markov's inequality, we can finally bound

$$\mathbb{P}\left(\exists n : \sum_{t=1}^{n} X_t < \frac{1}{2}\sum_{t=1}^{n} P_t - W\right) \leq \mathbb{P}(\tau < \infty) \leq \mathbb{P}(M_\tau > e^W) \leq e^{-W}\mathbb{E}[M_\tau] \leq e^{-W}.$$

□

Chapter 5

Policy Certificates: Towards Accountable and Minimax-Optimal Reinforcement Learning

This chapter is based on the work which I started during an internship at Google Cloud AI. It was published as:
Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. "Policy Certificates: Towards Accountable Reinforcement Learning". In: *International Conference on Machine Learning* (2019)

5.1 Introduction

There is increasing excitement around applications of machine learning, but also growing awareness and concerns about fairness, accountability and transparency. Recent research aims to address these concerns but most work focuses on supervised learning and only few results (Jabbari et al., 2016; Joseph et al., 2016; Kannan et al., 2017; Raghavan et al., 2018) exist on reinforcement learning (RL).

One challenge when applying RL in practice is that, unlike in supervised learning, the performance of an RL algorithm is typically not monotonically increasing with more data due to the trial-and-error nature of RL that necessitates exploration. Even sharp drops in policy performance during learning are common, e.g., when the agent starts to explore a new part of the state space. Such unpredictable performance fluctuation has limited the use of RL in high-stakes applications like healthcare, and calls for more *accountable* algorithms that can quantify and reveal their performance online during learning.

To address this lack of accountability, we propose that RL algorithms output *policy certificates* in episodic RL. Policy certificates consist of (1) a confidence interval of the algorithm's expected sum of rewards (expected return) in the next episode (policy return certificates) and (2) a bound on how far from the optimal return the performance can be (policy optimality certificates). Certificates make the policy's performance more transparent and accountable, and allow designers to intervene if necessary. For example, in medical applications, one would need to intervene unless the policy achieves a certain minimum treatment outcome; in financial applications, policy optimality certificates can be used to assess the potential loss when learning a trading strategy. In addition to accountability, we also want RL algorithms to be sample-efficient and quickly achieve good performance. To formally quantify accountability and sample-efficiency of an algorithm, we introduce a new framework for theoretical analysis called IPOC. IPOC bounds guarantee that certificates indeed bound the algorithm's expected performance in an episode, and prescribe the rate at which the algorithm's policy and certificates improve with more data. IPOC is

stronger than other frameworks like regret (Jaksch, Ortner, and Auer, 2010), PAC (Kakade, 2003) and Uniform-PAC (Dann, Lattimore, and Brunskill, 2017), that only guarantee the cumulative performance of the algorithm, but do not provide bounds for *individual* episodes during learning. IPOC also provides stronger bounds and more nuanced guarantees on per episode performance than KWIK Li, Littman, and Walsh, 2008.

A natural way to create accountable and sample-efficient RL algorithms is to combine existing sample-efficient algorithms with off-policy policy evaluation approaches to estimate the expected return (expected sum of rewards) of the algorithm's policy before each episode. Existing policy evaluation approaches estimate the return of a fixed policy from a batch of data (e.g., Thomas, Theocharous, and Ghavamzadeh, 2015a; Jiang and Li, 2016; Thomas and Brunskill, 2016). They provide little to no guarantees when the policy is not fixed but computed from that same batch of data, as is here the case. They also do not reason about the expected return of the unknown optimal policy which is necessary for providing policy optimality certificates. We found that by focusing on optimism-in-the-face-of-uncertainty (OFU) based RL algorithms for updating the policy and model-based policy evaluation techniques for estimating the policy returns, we can create sample-efficient algorithms that compute policy certificates on both the current policy's expected return and its difference to the optimal return. The main insight is that OFU algorithms compute an upper confidence bound on the optimal return from an empirical model when updating the policy. Model-based policy evaluation can leverage the same empirical model to compute a confidence interval on the policy's expected return, even when the policy depends on the data. We illustrate this approach with new algorithms for two different episodic settings.

Perhaps surprisingly, we show that in tabular Markov decision processes (MDPs) it can be beneficial to explicitly leverage the combination of OFU-based policy optimization and model-based policy evaluation to improve either component. Specifically, computing the certificates can directly improve the underlying OFU approach and knowing that the policy converges to the optimal policy at a certain rate improves the accuracy of policy return certificates. As a result, the guarantees for our new algorithm improve state-of-the-art regret and PAC bounds in problems with large horizons and are minimax-optimal up to lower-order terms.

The second setting we consider are finite MDPs with linear side information (context) (Abbasi-Yadkori and Neu, 2014; Hallak, Di Castro, and Mannor, 2015; Modi et al., 2018), which is of particular interest in practice. For example, in a drug treatment optimization task where each patient is one episode, context is the background information of the patient which influences the treatment outcome. While one expects the algorithm to learn a good policy quickly for frequent contexts, the performance for unusual patients may be significantly more variable due to the limited prior experience of the algorithm. Policy certificates allow humans to detect when the current policy is good for the current patient and intervene if a certified performance is deemed inadequate. For example, for this health monitoring application, a human expert could intervene to either directly specify the policy for that episode, or in the context of automated customer service, the service could be provided at reduced cost to the customer.

To summarize, we make the following main contributions in this chapter:

1. We introduce policy certificates and the IPOC framework for evaluating RL algorithms with certificates. Similar to existing frameworks like PAC, it provides formal requirements to be satisfied by the algorithm, here requiring the algorithm to be an efficient learner and to quantify its performance online through policy certificates.

2. We provide a new RL algorithm for finite, episodic MDPs that satisfies this definition, and show that it has stronger, minimax regret and PAC guarantees than prior work. Formally, our sample complexity bound is $\tilde{O}(SAH^2/\epsilon^2 + S^2AH^3/\epsilon)$ vs. prior $\tilde{O}(SAH^4/\epsilon^2 + S^2AH^3/\epsilon)$ (Dann, Lattimore, and Brunskill, 2017), and our regret bound $\tilde{O}(\sqrt{SAH^2T} + S^2AH^2)$ improves prior work (Azar, Osband,

and Munos, 2017) since it has minimax rate up to log-terms in the dominant term even for long horizons $H > SA$.

3. We introduce a new RL algorithm for finite, episodic MDPs with linear side information that has a cumulative IPOC bound, which is tighter than past results Abbasi-Yadkori and Neu, 2014 by a factor of $\sqrt{SAH}$.

5.2 Setting and Notation

We consider episodic RL problems where the agent interacts with the environment in episodes of a certain length. While the framework for policy certificates applies more bredly, we focus on finite MDPs with linear side information (Modi et al., 2018; Hallak, Di Castro, and Mannor, 2015; Abbasi-Yadkori and Neu, 2014) for concreteness. This setting includes tabular MDPs as a special case but is more general and can model variations in the environment across episodes, e.g., because different episodes correspond to treating different patients in a healthcare application. Unlike the tabular special case, function approximation is necessary for efficient learning.

Tabular MDPs The agent interacts with the MDP in episodes indexed by k. Each episode is a sequence $(s_{k,1}, a_{k,1}, r_{k,1}, \dots, s_{k,H}, a_{k,H}, r_{k,H})$ of H states $s_{k,h} \in \mathcal{S}$, actions $a_{k,h} \in \mathcal{A}$ and scalar rewards $r_{k,h} \in [0,1]$. For notational simplicity, we assume that the initial state $s_{k,1}$ is deterministic. The actions are taken as prescribed by the agent's policy π_k and we here focus on deterministic time-dependent policies, i.e., $a_{k,h} = \pi_k(s_{k,h}, h)$ for all time steps $h \in [H] := \{1, 2, \dots H\}$. The successor states and rewards are sampled from the MDP as $s_{k,h+1} \sim P(s_{k,h}, a_{k,h})$ and $r_{k,h} \sim P_R(s_{k,h}, a_{k,h})$. In tabular MDPs the size of the state space $S = |\mathcal{S}|$ and action space $A = |\mathcal{A}|$ are finite.

Finite MDPs with linear side information. We assume that state- and action-space are finite as in tabular MDPs, but here the agent essentially interacts with a family of infinitely many tabular MDPs that is parameterized by linear contexts. At the beginning of episode k, two contexts, $x_k^{(r)} \in \mathbb{R}^{d^{(r)}}$ and $x_k^{(p)} \in \mathbb{R}^{d^{(p)}}$, are observed and the agent interacts in this episode with a tabular MDP, whose dynamics and reward function depend on the contexts in a linear fashion. Specifically, it is assumed that the rewards are sampled from $P_R(s,a)$ with means $r_k(s,a) = (x_k^{(r)})^\top \theta_{s,a}^{(r)}$ and transition probabilities are $P_k(s'|s,a) = (x_k^{(p)})^\top \theta_{s',s,a}^{(p)}$ where $\theta_{s,a}^{(r)} \in \mathbb{R}^{d^{(r)}}$ and $\theta_{s',s,a}^{(p)} \in \mathbb{R}^{d^{(p)}}$ are unknown parameter vectors for each $s, s' \in \mathcal{S}, a \in \mathcal{A}$. As a regularity condition, we assume bounded parameters, i.e., $\|\theta_{s,a}^{(r)}\|_2 \le \xi_{\theta^{(r)}}$ and $\|\theta_{s',s,a}^{(p)}\|_2 \le \xi_{\theta^{(p)}}$ as well as bounded contexts $\|x_k^{(r)}\|_2 \le \xi_{x^{(r)}}$ and $\|x_k^{(p)}\|_2 \le \xi_{x^{(p)}}$. We allow $x_k^{(r)}$ and $x_k^{(p)}$ to be different, and use x_k to denote $(x_k^{(r)}, x_k^{(p)})$ in the following. Note that our framework and algorithms can handle adversarially chosen contexts.

Return and optimality gap. The quality of a policy π in any episode k is evaluated by the *total expected reward* or *expected return*: $\rho_k(\pi) := \mathbb{E}\left[\sum_{h=1}^H r_{k,h} \middle| a_{k,1:H} \sim \pi\right]$, where this notation means that all actions in the episode are taken as prescribed by a policy π. Optimal policy and return $\rho_k^\star = \max_\pi \rho_k(\pi)$ may depend on the episode's contexts. The difference of achieved and optimal return is called *optimality gap* $\Delta_k = \rho_k^\star - \rho_k(\pi_k)$ for each episode k where π_k is the algorithm's policy in that episode.

Additional notation. We denote the largest possible optimality gap by $\Delta_{\max} = H$, and the value functions of π in episode k by $Q_h^{\pi_k}(s,a) = \mathbb{E}[\sum_{t=h}^H r_{k,t} | a_{k,h} = a, a_{k,h+1:H} \sim \pi]$ and $V_h^{\pi_k}(s) = Q_h^{\pi_k}(s, \pi(s,h))$. Optimal versions are marked by superscript $\star$ and subscripts are omitted when unambiguous. We treat $P(s,a)$ as a linear operator, that is, $P(s,a)f = \sum_{s' \in \mathcal{S}} P(s'|s,a) f(s')$ for any

$f : \mathcal{S} \to \mathbb{R}$. We also use $\sigma_q(f) = \sqrt{q(f - qf)^2}$ for the standard deviation of f with respect to a state distribution q and $V_h^{\max} = (H - h + 1)$ for all $h \in [H]$. We also use the common short hand notation $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$ as well as $\tilde{O}(f) = O(f \cdot \text{poly}(\log(f)))$.

5.3 The IPOC Framework

During execution, the optimality gaps Δ_k are hidden and the algorithm only observes the sum of rewards which is a sample of $\rho_k(\pi_k)$. This causes risk as one does not know whether the algorithm is playing a good or potentially bad policy. We introduce a new learning framework that mitigates this limitation. This framework forces the algorithm to output its current policy π_k as well as certificates $\epsilon_k \in \mathbb{R}_+$ and $\mathcal{I}_k \subseteq \mathbb{R}$ before each episode k. The *return certificate* $\mathcal{I}_k$ is a confidence interval on the return of the policy, while the *optimality certificate* ϵ_k informs the user how sub-optimal the policy can be for the current context, i.e., $\epsilon_k \geq \Delta_k$. Certificates allow one to intervene if needed. For example, in automated customer services, one might reduce the service price in episode k if certificate ϵ_k is above a certain threshold, since the quality of the provided service cannot be guaranteed. When there is no context, an optimality certificate upper bounds the sub-optimality of the current policy in any episode which makes algorithms anytime interruptable (Zilberstein and Russell, 1996): one is guaranteed to always know a policy with improving performance. Our learning framework is formalized as follows:

Definition 50 (Individual Policy Certificates (IPOC) Bounds)**.** *An algorithm satisfies an individual policy certificate (IPOC) bound F if for a given $\delta \in (0, 1)$ it outputs the current policy π_k, a* return certificate *$\mathcal{I}_k \subseteq \mathbb{R}$ and an* optimality certificate *ϵ_k with $\epsilon_k \geq |\mathcal{I}_k|$ before each episode k (after observing the contexts) so that with probability at least $1 - \delta$:*

1. *all return certificates contain the expected return of policy π_k played in episode k and all optimality certificates are upper bounds on the sub-optimality of π_k, i.e., $\forall\, k \in \mathbb{N} : \quad \epsilon_k \geq \Delta_k$ and $\rho_k(\pi_k) \in \mathcal{I}_k$; and either*

2a. *for all number of episodes T the cumulative sum of certificates is bounded $\sum_{k=1}^{T} \epsilon_k \leq F(W, T, \delta)$ (Cumulative Version), or*

2b. *for any threshold ϵ, the number of times certificates can exceed the threshold is bounded as $\sum_{k=1}^{\infty} \mathbf{1}\{\epsilon_k > \epsilon\} \leq F(W, \epsilon, \delta)$ (Mistake Version).*

Here, W can be (known or unknown) properties of the environment. If conditions 1 and 2a hold, we say the algorithm has a cumulative IPOC bound and if conditions 1 and 2b hold, we say the algorithm has a mistake IPOC bound.

Condition 1 alone would be trivial to satisfy with $\epsilon_k = \Delta_{\max}$ and $\mathcal{I}_k = [0, \Delta_{\max}]$, but condition 2 prohibits this by controlling the size of ϵ_k (and therefore the size of $|\mathcal{I}_k| \leq \epsilon_k$). Condition 2a bounds the cumulative sum of optimality certificates (similar to regret bounds), and condition 2b bounds the size of the superlevel sets of ϵ_k (similar to PAC bounds). We allow both alternatives as condition 2b is stronger but one sometimes can only prove condition 2a (see Section 5.11). An IPOC bound controls simultaneously the quality of certificates (how big $\epsilon_k - \Delta_k$ and $|\mathcal{I}_k|$ are) as well as the optimality gaps Δ_k themselves and, hence, an IPOC bound not only guarantees that the algorithm improves its policy but also becomes better at telling us how well the policy performs. Note that the condition $\epsilon_k \geq |\mathcal{I}_k|$ in Definition 50 is natural as any upper bound on $\rho_k^\star$ is also an upper bound on $\rho_k(\pi_k)$ and is made for notational convenience.

We would like to emphasize that we provide certificates on the expected return, the *expected* sum of rewards given the algorithm's policy, in the next episode. Due to the stochasticity in the environment, one in general cannot hope to accurately predict the sum of rewards directly. Since expected return is the default optimization criteria in RL, certificates for it are a natural starting point and relevant in many scenarios. Nonetheless, certificates for other properties of the sum-of-reward distribution of a policy are an interesting

direction for future work. For example, one might want certificates on properties that take into account the variability of the sum of rewards (e.g., conditional value at risk) in high-stakes applications which are often the objective in risk-sensitive RL.

5.3.1 Relation to Existing Frameworks

Unlike IPOC, existing frameworks for RL only guarantee sample-efficiency of the algorithm over multiple episodes and do not provide performance bounds for single episodes during learning. The common existing frameworks are:

- *Mistake-style PAC bounds* (Strehl, Li, Wiewiora, et al., 2006; Strehl, Li, and Littman, 2009; Szita and Szepesvári, 2010; Lattimore and Hutter, 2012; Dann and Brunskill, 2015) bound the number of ϵ-mistakes, that is, the size of the set $\{k \in \mathbb{N} \,:\, \Delta_k > \epsilon\}$ with high probability, but do not tell us when mistakes happen. The same is true for the stronger Uniform-PAC bounds (Dann, Lattimore, and Brunskill, 2017) which hold for all ϵ jointly.
- *Supervised-learning style PAC bounds* (Kearns and Singh, 2002; Jiang, Krishnamurthy, et al., 2017; Dann, Jiang, et al., 2018) ensure that the algorithm outputs an ϵ-optimal policy for a given ϵ, i.e., they ensure $\Delta_k \leq \epsilon$ for k greater than the bound. Yet, they need to know ϵ ahead of time and tell us nothing about Δ_k during learning (for k smaller than the bound).
- *Regret bounds* (Osband, Russo, and Van Roy, 2013; Osband, Van Roy, and Wen, 2016; Azar, Osband, and Munos, 2017; Jin et al., 2018) control the cumulative sum of optimality gaps $\sum_{k=1}^{T} \Delta_k$ (regret) which does not yield any nontrivial guarantee for individual Δ_k because it does not reveal which optimality gaps are small.

We show that mistake IPOC bounds are stronger than any of the above guarantees, i.e., they imply Uniform PAC, PAC, and regret bounds. Cumulative IPOC bounds are slightly weaker but still imply regret bounds. Both versions of IPOC also ensure that the algorithm is anytime interruptable, i.e., it can be used to find better and better policies that have small Δ_k with high probability $1-\delta$. That means IPOC bounds imply supervised-learning style PAC bounds for all ϵ jointly. These claims are formalized as follows:

Proposition 51. *Assume an algorithm has a cumulative IPOC bound $F(W, T, \delta)$.*

1. *Then it has a regret bound of same order, i.e., with probability at least $1-\delta$, for all T the regret $R(T) := \sum_{k=1}^{T} \Delta_k$ is bounded by $F(W, T, \delta)$.*
2. *If F has the form $\sum_{p=0}^{N} (C_p(W, \delta)T)^{\frac{p}{p+1}}$ for appropriate functions C_p, then with probability at least $1-\delta$ for any ϵ, it outputs a certificate $\epsilon_k \leq \epsilon$ within*

$$\sum_{p=0}^{N} \frac{C_p(W, \delta)^p (N+1)^{p+1}}{\epsilon^{p+1}}$$

episodes. Hence, for settings without context, the algorithm outputs an ϵ-optimal policy within that number of episodes (supervised learning-style PAC bound).

Proposition 52. *If an algorithm has a mistake IPOC bound $F(W, \epsilon, \delta)$, then*

1. *it has a uniform PAC bound $F(W, \epsilon, \delta)$, i.e., with probability at least $1-\delta$, the number of episodes with $\Delta_k \geq \epsilon$ is at most $F(W, \epsilon, \delta)$ for all $\epsilon > 0$;*
2. *with probability $\geq 1-\delta$ for all ϵ, it outputs a certificate $\epsilon_k \leq \epsilon$ within $F(W, \epsilon, \delta) + 1$ episodes. For settings without context, that means the algorithm outputs an ϵ-optimal policy within that many episodes (supervised learning-style PAC).*
3. *if F has the form $\sum_{p=1}^{N} \frac{C_p(W,\delta)}{\epsilon^p} \left(\ln \frac{\tilde{C}(W,\delta)}{\epsilon}\right)^{np}$ with $C_p(W, \delta) \geq 1$ and constants $N, n \in \mathbb{N}$, it also has a cumulative IPOC bound of order*

$$\tilde{O}\left(\sum_{p=1}^{N} C_p(W,\delta)^{1/p} T^{\frac{p-1}{p}} \operatorname{polylog}(\Delta_{\max}, \tilde{C}(W,\delta), T)\right).$$

The functional form in part 2 of Proposition 51 includes common polynomial bounds like $O(\sqrt{T})$ or $O(T^{2/3})$ with appropriate factors and similarly for part 3 of Proposition 52 which covers for example $\tilde{O}(1/\epsilon^2)$.

Our IPOC framework is similar to KWIK (Li, Littman, and Walsh, 2008), in that the algorithm is required to declare how well it will perform. Hower, KWIK only requires an algorithm to declare whether the output will perform better than a single pre-specified input threshold. Existing KWIK for RL methods only provide such a binary classification, and have less strong learning guarantees. In a sense IPOC is a generalization of KWIK.

5.4 Algorithms with Policy Certificates

A natural path to obtain RL algorithms with IPOC bounds is to combine existing provably efficient online RL algorithms with an off-policy policy evaluation method to compute a confidence interval on the online RL algorithm's policy for the current episode. This yields policy return certificates, but not necessarily policy optimality certificates – bounds on the difference of the optimal and current policy's expected return. Estimating the optimal return using off-policy evaluation algorithms in order to compute optimality certificates would require a significant computational burden, e.g. evaluating all (exponentially many) policies.

However optimism in the face of uncertainty (OFU) algorithms can be modified to provide both policy return certificates and optimality certificates without the need for a separate off-policy policy optimization step. Specifically, we here consider OFU algorithms that maintain an upper confidence bound (for a potentially changing confidence level) on the optimal value function $Q^\star_{k,h}$ and therefore optimal return $\rho^\star_k$. This bound is also an upper bound on the expected return of the current policy which is chosen to maximize this bound. Many OFU methods explicitly maintain a confidence set of the MDP model to compute the upper confidence bound on $Q^\star_{k,h}$. These same confidence sets of the model can be used to compute a lower bound on the value function of the current policy. In doing so, OFU algorithms can be modified with little computational overhead to provide policy return and optimality certificates.

For these reasons, we focus on OFU methods, introducing two new algorithms with policy certificates, one for tabular MDPs and and one for the more general MDPs with linear side information setting. Both approaches have a similar structure, but leverage different confidence sets and model estimators. In the first case, we show that maintaining lower bounds on the current policy's value has significant benefits beyond enabling policy certificates: lower bounds help us to derive a tighter bound on our uncertainty over the range of future values. Thus we are able to provide the strongest, to our knowledge, PAC and regret bounds for tabular MDPs. It remains an intriguing but non-trivial question if we can create confidence sets that leverage explicit upper and lower bounds for the linear side information setting.

5.4.1 Tabular MDPs

We present the `ORLC` (optimistic RL with certificates) Algorithm shown in Algorithm 4 (see Algorithm 6 in Section 5.9 for a version with empirically tighter confidence bounds but same theoretical guarantees). It shares similar structure with recent OFU algorithms like `UBEV` (Dann, Lattimore, and Brunskill, 2017) and `UCBVI-BF` (Azar, Osband, and Munos, 2017) but has some significant differences highlighted in red. Before each episode k, Algorithm 4 computes an optimistic estimate $\tilde{Q}_{k,h}$ of $Q^\star_h$ in Line 10 by

Algorithm 4: ORLC (**O**ptimistic **R**einforcement **L**earning with **C**ertificates)

Input : failure tolerance $\delta \in (0,1]$

1 $\phi(n) = 1 \wedge \sqrt{\frac{0.52}{n}\left(1.4 \ln\ln(e \vee n) + \ln\frac{26SA(H+1+S)}{\delta}\right)}$; $\quad \tilde{V}_{k,H+1}(s) = 0$; $\quad \underset{\sim}{V}_{k,H+1}(s) = 0 \quad \forall s \in \mathcal{S}, k \in \mathbb{N}$;

2 **for** $k = 1, 2, 3, \ldots$ **do**

3 **for** $s', s \in \mathcal{S}, a \in \mathcal{A}$ **do** // update empirical model and number of observations

4 $n_k(s,a) = \sum_{i=1}^{k-1} \sum_{h=1}^{H} \mathbf{1}\{s_{i,h} = s, a_{i,h} = a\}$; // number of times (s,a) was observed

5 $\hat{r}_k(s,a) = \frac{1}{n_k(s,a)} \sum_{i=1}^{k-1} \sum_{h=1}^{H} r_{i,h} \mathbf{1}\{s_{i,h} = s, a_{i,h} = a\}$; // avg. reward observed for (s,a)

6 $\hat{P}_k(s'|s,a) = \frac{1}{n_k(s,a)} \sum_{i=1}^{k-1} \sum_{h=1}^{H} \mathbf{1}\{s_{i,h} = s, a_{i,h} = a, s_{i,h+1} = s'\}$

7 **for** $h = H$ **to** 1 ***and*** $s \in \mathcal{S}$ **do** // optimistic planning with upper and lower confidence bounds

8 **for** $a \in \mathcal{A}$ **do**

9 $\psi_{k,h}(s,a) = (1+\sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}))\phi(n_k(s,a)) + 45SH^2\phi(n_k(s,a))^2 + \frac{1}{H}\hat{P}(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})$;

10 $\tilde{Q}_{k,h}(s,a) = 0 \vee \left(\hat{r}_k(s,a) + \hat{P}_k(s,a)\tilde{V}_{k,h+1} + \psi_{k,h}(s,a)\right) \wedge V_h^{\max}$; // UCB of $Q^\star_{h+1}$

11 $\underset{\sim}{Q}_{k,h}(s,a) = 0 \vee \left(\hat{r}_k(s,a) + \hat{P}_k(s,a)\underset{\sim}{V}_{k,h+1} - \psi_{k,h}(s,a)\right) \wedge V_h^{\max}$; // LCB of $Q^{\pi_k}_{h+1}$

12 $\pi_k(s,h) = \operatorname{argmax}_a \tilde{Q}_{k,h}(s,a)$; $\quad \tilde{V}_{k,h}(s) = \tilde{Q}_{k,h}(s,\pi_k(s,h))$; $\quad \underset{\sim}{V}_{k,h}(s) = \underset{\sim}{Q}_{k,h}(s,\pi_k(s,h))$;

13 **output** policy π_k with certificates $\mathcal{I}_k = [\underset{\sim}{V}_{k,1}(s_{1,1}), \tilde{V}_{k,1}(s_{1,1})]$ and $\epsilon_k = |\mathcal{I}_k|$;

14 **sample episode** k with policy π_k; // Observe $s_{k,1}, a_{k,1}, r_{k,1}, s_{k,2}, \ldots, s_{k,H}, a_{k,H}, r_{k,H}$

dynamic programming on the empirical model $(\hat{P}_k, \hat{r}_k)$ with confidence intervals $\psi_{k,h}$. Importantly, it also computes $\underset{\sim}{Q}_{k,h}$, a pessimistic estimate of $Q_h^{\pi_k}$ in similar fashion in Line 11. The optimistic and pessimistic estimates $\underset{\sim}{Q}_{k,h}, \tilde{Q}_{k,h}$ (resp. $\underset{\sim}{V}_{k,h}, \tilde{V}_{k,h}$) allow us to compute the certificates ϵ_k and $\mathcal{I}_k$ and enables more sample-efficient learning. Specifically, Algorithm 4 uses a novel form of confidence intervals ψ that explicitly depends on this difference. These confidence intervals are key for proving the following IPOC bound:

Theorem 53 (Mistake IPOC Bound of Alg. 4)**.** *For any given $\delta \in (0,1)$, Alg. 4 satisfies in any tabular MDP with S states, A actions and horizon H, the following Mistake IPOC bound: For all $\epsilon > 0$, the number of episodes where Alg. 4 outputs a certificate $|\mathcal{I}_k| = \epsilon_k > \epsilon$ is*

$$\tilde{O}\left(\left(\frac{SAH^2}{\epsilon^2} + \frac{S^2AH^3}{\epsilon}\right)\ln\frac{1}{\delta}\right).$$

By Proposition 52, this implies a Uniform-PAC bound of same order as well as the regret and PAC bounds listed in Table 5.1. This table also contains previous state of the art bounds of each type[1] as well as

[1]These model-free and model-based methods have the best known bounds in our problem class. Q-learning with UCB and UBEV allow time-dependent dynamics. One might be able to improve their regret bound by $\sqrt{H}$ when adapting them to our setting. Note that by augmenting our state space with a time index, our algorithm also achieves minimax optimality with $\tilde{O}(\sqrt{SAH^3T})$ regret up to lower order terms in their setting.

Algorithm	Regret	PAC	Mistake IPOC
UCBVI-BF	$\tilde{O}(\sqrt{SAH^2T} + \sqrt{H^3T} + S^2AH^2)$	-	-
UCBQ [1]	$\tilde{O}(\sqrt{SAH^4T} + S^{1.5}A^{1.5}H^{4.5})$	-	-
UCFH	-	$\tilde{O}\left(\frac{S^2AH^2}{\epsilon^2}\right)$	-
UBEV [1]	$\tilde{O}(\sqrt{SAH^4T} + S^2AH^3)$	$\tilde{O}\left(\frac{SAH^4}{\epsilon^2} + \frac{S^2AH^3}{\epsilon}\right)$	-
EULER	$\tilde{O}(\sqrt{SAH^2T} + S^{1.5}AH^2(\sqrt{S}+H))$	-	-
ORLC (this work)	$\tilde{O}(\sqrt{SAH^2T} + S^2AH^2)$	$\tilde{O}\left(\frac{SAH^2}{\epsilon^2} + \frac{S^2AH^3}{\epsilon}\right)$	$\tilde{O}\left(\frac{SAH^2}{\epsilon^2} + \frac{S^2AH^3}{\epsilon}\right)$
Lower bounds	$\Omega\left(\sqrt{SAH^2T}\right)$	$\Omega\left(\frac{SAH^2}{\epsilon^2}\right)$	$\Omega\left(\frac{SAH^2}{\epsilon^2}\right)$

Table 5.1: Comparison of the state of the art problem-independent bounds for episodic RL in tabular MDPs. This includes UCBVI-BF (Azar, Osband, and Munos, 2017), UCBQ (Jin et al., 2018), UCFH (Dann and Brunskill, 2015), UBEV (Dann, Lattimore, and Brunskill, 2017), EULER (Zanette and Brunskill, 2019) and our ORLC algorithm. A dash means that the algorithm does not satisfy a non-trivial bound without modifications. T is the number of episodes and $\ln(1/\delta)$ factors are omitted for readability. For an empirical comparison of the sample-complexity of these approaches, see Section 5.12.2.

lower bounds. The IPOC lower bound follows from the PAC lower bound by Dann and Brunskill (2015) and Proposition 52. For ϵ small enough ($\leq O(1/(SH))$ specifically), our IPOC bound is minimax, i.e., the best achievable, up to log-factors. This is also true for the Uniform-PAC and PAC bounds implied by Theorem 53 as well as the implied regret bound when the number of episodes $T = \Omega(S^3AH^4)$ is large enough. ORLC is the first algorithm to achieve this minimax rate for PAC and Uniform-PAC. While UCBVI-BF achieves minimax regret for problems with small horizon, their bound is suboptimal when $H > SA$.

While the lower-order term in our IPOC mistake bound is $\tilde{O}(S^2AH^3)$, we can achieve a tighter $\tilde{O}(S^2AH^2)$ dependency in our cumulative IPOC and regret bound with a slightly refined analysis:

Theorem 54. *For any given $\delta \in (0,1)$, Alg. 4 satisfies in any tabular MDP with S states, A actions and horizon H, the following cumulative IPOC bound*

$$\tilde{O}\left(\sqrt{SAH^2T}\ln\frac{1}{\delta} + S^2AH^2\ln\frac{T}{\delta}\right),$$

where T is the number of episodes.

This results is to the best of our knowledge the tightest problem-independent regret bound for this setting, including a slight improvement over the recent problem-independent bound by Zanette and Brunskill (2019).

We defer details of our IPOC analysis to Section 5.9 but the main advances leverage that $[Q_{k,h}(s,a), \tilde{Q}_{k,h}(s,a)]$ is an *observable* confidence interval for both $Q_h^\star(s,a)$ and $Q_h^{\pi_k}(s,a)$. Specifically, our main novel insights are:

- While prior works (e.g. Lattimore and Hutter, 2012; Dann and Brunskill, 2015) control the suboptimality $Q_h^\star - Q_h^{\pi_k}$ of the policy by recursively bounding $\tilde{Q}_{k,h} - Q_h^{\pi_k}$, we instead recursively bound $\tilde{Q}_{k,h} - Q_{k,h} \leq 2\psi_{k,h} + \hat{P}_k(\tilde{V}_{k,h+1} - V_{k,h+1})$ which is not only simpler but also controls both the suboptimality of the policy and the size of the certificates simultaneously.
- As existing work (e.g. Azar, Osband, and Munos, 2017; Jin et al., 2018), we use empirical Bernstein-type concentration inequalities to construct $\tilde{Q}_{k,h}(s,a)$ as an upper bound to $Q_h^\star(s,a) = r(s,a) +$

$P(s,a)V^\star_{h+1}$. This results in a dependency of the upper bound on the variance of the optimal next state value $\sigma_{\hat{P}_k(s,a)}(V^\star_{h+1})^2$ under the empirical model. Since $V^\star_{h+1}$ is unknown this has to be upper-bounded by $\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})^2 + B$ with an additional bonus B to account for the difference between the values, $\tilde{V}_{k,h+1} - V^\star_{h+1}$, which is again unobservable. Azar, Osband, and Munos (2017) now constructs an observable bound on B through an intricate regret analysis that involves additional high-probability bounds on error terms (see their $\mathcal{E}_{fr}/\mathcal{E}_{az}$ events) which causes the suboptimal $\sqrt{H^3T}$ term in their regret bound. Instead, we use the fact that $\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}$ is an observable upper bound on $\tilde{V}_{k,h+1} - V^\star_{h+1}$ which we can directly use in our confidence widths $\psi_{k,h}$ (see the last term in Line 9 of Alg. 4). Hence, availability of lower bounds through certificates improves also our upper confidence bounds on $Q^\star_h$ and yields more sample-efficient exploration with improved performance bounds as we avoid additional high-probability bounds of error terms.

- As opposed to the upper bounds, we cannot simply apply concentration inequalities to construct $\underset{\sim}{Q}_{k,h}(s,a)$ as a lower bound to Q^{π_k} because the estimation target $Q^{\pi_k}(s,a) = r(s,a) + P(s,a)V^{\pi_k}_{h+1}$ is itself random. The policy π_k depends in highly non-trivial ways on all samples from which we also estimate the empirical model $\hat{P}_k, \hat{r}_k$. A prevalent approach in model-based policy evaluation (Strehl and Littman, 2008; Ghavamzadeh, Petrik, and Chow, 2016, e.g.) to deal with this challenge is to instead apply a concentration argument on the ℓ_1 distance of the transition estimates $\|P(s,a) - \hat{P}_k(s,a)\|_1 \le \sqrt{S}\phi(n_k(s,a))$. This yields confidence intervals that shrink at a rate of $H\sqrt{S}\phi(n_k(s,a))$. Instead, we can exploit that π_k is generated by a sample-efficient algorithm and construct $\underset{\sim}{Q}_{k,h}$ as a lower bound to the non-random quantity $r(s,a) + P(s,a)V^\star_{h+1}$. We account for the difference $P(s,a)(V^\star_{h+1} - V^{\pi_k}_{h+1}) \le P(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})$ explicitly, again through a recursive bound. This allows us to achieve confidence intervals that shrink at a faster rate of $\psi_{k,h} \approx H\phi(n_k(s,a)) + SH^2\phi(n_k(s,a))^2$ without the $\sqrt{S}$ dependency in the dominating $\phi(n_k(s,a))$ term (recall $\phi(n_k(s,a)) \le 1$ and goes to 0). Hence, by leveraging that π_k is computed by a sample-efficient approach, we improve the tightness of the certificates.

5.4.2 MDPs With Linear Side Information

We now present an algorithm for the more general setting with side information, which, for example, allows us to take background information about a customer into account and generalize across different customers. Algorithm 5 gives an extension, called `ORLC-SI`, of the OFU algorithm by Abbasi-Yadkori and Neu (2014). Its overall structure is the same as the tabular Algorithm 4 but here the empirical model are least-squares estimates of the model parameters evaluated at the current contexts. Specifically, the empirical transition probability $\hat{P}_k(s'|s,a)$ is $(x^{(p)}_k)^\top \hat{\theta}_{s',s,a}$ where $\hat{\theta}_{s',s,a}$ is the least squares estimate of model parameter $\theta_{s',s,a}$. Since transition probabilities are normalized, this estimate is then clipped to $[0,1]$. This model is estimated separately for each (s',s,a)-triple, but generalizes across different contexts. The confidence widths $\psi_{k,h}$ are derived using ellipsoid confidence sets on model parameters. We show the following IPOC bound:

Theorem 55 (Cumulative IPOC Bound for Alg. 5)**.** *For any $\delta \in (0,1)$ and regularizer $\lambda > 0$, Alg. 5 satisfies the following cumulative IPOC bound in any MDP with contexts of dimensions $d^{(r)}$ and $d^{(p)}$ and bounded parameters $\xi_{\theta^{(r)}} \le \sqrt{d^{(p)}}, \xi_{\theta^{(p)}} \le \sqrt{d^{(p)}}$. With prob. at least $1-\delta$ all return certificates contain the expected return of π_k and optimality certificates are upper bounds on the optimality gaps and their total sum after T episodes is bounded for all T by*

$$\tilde{O}\left(\sqrt{S^3AH^4T}\lambda(d^{(p)} + d^{(r)}) \log \frac{\xi^2_{x^{(p)}} + \xi^2_{x^{(r)}}}{\lambda\delta}\right).$$

Algorithm 5: `ORLC-SI` (**O**ptimistic **R**einforcement **L**earning with **C**ertificates and **S**ide **I**nformation)

Input : failure prob. $\delta \in (0,1]$, regularizer $\lambda > 0$

$\xi_{\theta^{(r)}} = \sqrt{d}$; $\xi_{\theta^{(p)}} = \sqrt{d}$; $\quad \tilde{V}_{k,H+1}(s) = 0$; $\quad \underset{\sim}{V}_{k,H+1}(s) = 0 \quad \forall s \in \mathcal{S}, k \in \mathbb{N}$;

$\phi(N, x, \xi) := \left[\sqrt{\lambda}\xi + \sqrt{\frac{1}{2}\ln\frac{S(SA+A+H)}{\delta} + \frac{1}{4}\ln\frac{\det N}{\det(\lambda I)}}\right] \|x\|_{N^{-1}}$;

for $k = 1, 2, 3, \ldots$ **do**

  Observe current contexts $x_k^{(r)}$ and $x_k^{(p)}$;

  for $s, s' \in \mathcal{S}, a \in \mathcal{A}$ **do** `// estimate model with least-squares`

    $N_{k,s,a}^{(q)} = \lambda I + \sum_{i=1}^{k-1}\sum_{h=1}^{H} \mathbf{1}\{s_{i,h} = s, a_{i,h} = a\} x_k^{(q)} (x_k^{(q)})^\top \quad$ for $q \in \{r, p\}$;

    $\hat{\theta}_{k,s,a}^{(r)} = (N_{k,s,a}^{(r)})^{-1} \sum_{i=1}^{k-1}\sum_{h=1}^{H} \mathbf{1}\{s_{i,h} = s, a_{i,h} = a\} x_k^{(r)} r_{i,h}$; $\quad \hat{r}_k(s,a) = 0 \vee (x_k^{(r)})^\top \hat{\theta}_{k,s,a}^{(r)} \wedge 1$;

    $\hat{\theta}_{s',s,a}^{(p)} = (N_{k,s,a}^{(p)})^{-1} \sum_{i=1}^{k-1}\sum_{h=1}^{H} \mathbf{1}\{s_{i,h} = s, a_{i,h} = a, s_{i,h+1} = s'\} x_k^{(p)}$;

    $\hat{P}_k(s'|s,a) = 0 \vee (x_k^{(p)})^\top \hat{\theta}_{k,s',s,a}^{(p)} \wedge 1$;

  for $h = H$ **to** 1 ***and*** $s \in \mathcal{S}$ **do** `// optimistic planning with ellipsoid confidence bounds`

    for $a \in \mathcal{A}$ **do**

      $\psi_{k,h}(s,a) = \|\tilde{V}_{k,h+1}\|_1 \phi(N_{k,s,a}^{(p)}, x_k^{(p)}, \xi_{\theta^{(p)}}) + \phi(N_{k,s,a}^{(r)}, x_k^{(r)}, \xi_{\theta^{(r)}})$;

      $\tilde{Q}_{k,h}(s,a) = 0 \vee \left(\hat{r}_k(s,a) + \hat{P}_k(s,a)\tilde{V}_{k,h+1} + \psi_{k,h}(s,a)\right) \wedge V_h^{\max}$; `// UCB of` $Q_{h+1}^\star$

      $\underset{\sim}{Q}_{k,h}(s,a) = 0 \vee \left(\hat{r}_k(s,a) + \hat{P}_k(s,a)\underset{\sim}{V}_{k,h+1} - \psi_{k,h}(s,a)\right) \wedge V_h^{\max}$; `// LCB of` $Q_{h+1}^{\pi_k}$

    $\pi_k(s,h) = \operatorname{argmax}_a \tilde{Q}_{k,h}(s,a)$; $\quad \tilde{V}_{k,h}(s) = \tilde{Q}_{k,h}(s, \pi_k(s,t))$; $\quad \underset{\sim}{V}_{k,h}(s) = \underset{\sim}{Q}_{k,h}(s, \pi_k(s,t))$;

  output policy π_k with certificates $\mathcal{I}_k = [\underset{\sim}{V}_{k,1}(s_{1,1}), \tilde{V}_{k,1}(s_{1,1})]$ and $\epsilon_k = |\mathcal{I}_k|$;

  sample episode k with policy π_k ; `// Observe` $s_{k,1}, a_{k,1}, r_{k,1}, s_{k,2}, \ldots, s_{k,H}, a_{k,H}, r_{k,H}$

By Proposition 51, this IPOC bound implies a regret bound of the same order which improves on the $\tilde{O}(\sqrt{d^2S^4AH^5T\log 1/\delta})$ regret bound of Abbasi-Yadkori and Neu (2014) with $d = d^{(p)} + d^{(r)}$ by a factor of $\sqrt{SAH}$. While they make a different modelling assumption (generalized linear instead of linear), we believe at least our better S dependency is due to using improved least-squares estimators for the transition dynamics [2] and can likely be transferred to their setting. The mistake-type PAC bound by Modi et al. (2018) is not comparable because our cumulative IPOC bound does not imply a mistake-type PAC bound.[3] Nonetheless, loosely translating our result to a PAC-like bound yields $\tilde{O}\left(\frac{d^2S^3AH^5}{\epsilon^2}\right)$ which is much smaller than their $\tilde{O}\left(\frac{d^2SAH^4}{\epsilon^5}\max\{d^2, S^2\}\right)$ bound for small ϵ.

The confidence bounds in Alg. 5 are more general but looser than those for the tabular case of Alg. 4. Instantiating the IPOC bound for Alg. 5 from Theorem 55 for tabular MDPs ($x_k^{(r)} = x_k^{(p)} = 1$) yields $\tilde{O}(\sqrt{S^3AH^4T})$ which is worse than the cumulative IPOC bound $\tilde{O}(\sqrt{SAH^2T} + S^2AH^3)$ of Alg. 4 implied by Thm. 53 and Prop. 52.

[2] They estimate $\theta_{s',s,a}$ only from samples where the transition $s, a \to s'$ was observed instead of all occurrences of s, a (no matter whether s' was the next state).

[3] An algorithm with a sub-linear cumulative IPOC bound can output a certificate larger than a threshold $\epsilon_k \geq \epsilon$ infinitely often as long as it does so sufficiently less frequently (see Section 5.11).

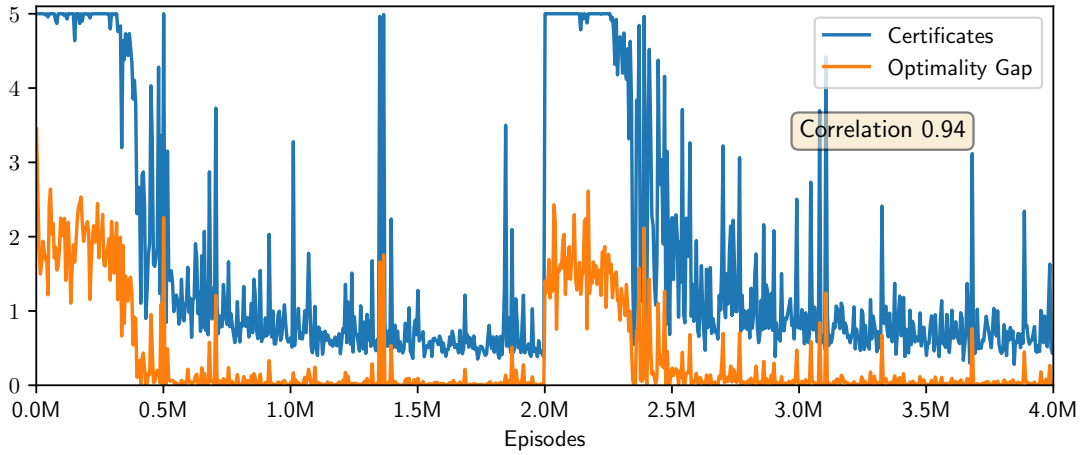

Figure 5.1: Certificates and (unobserved) optimality gaps of Algorithm 5 for 4M episodes on an MDP with context distribution shift after 2M (episodes sub-sampled for better visualization)

By Prop. 52, a mistake IPOC bound is stronger than the cumulative version we proved for Algorithm 5. One might wonder if Algorithm 5 also satisfies a mistake bound, but in Section 5.11 we show that this is not the case because of its non-decreasing ellipsoid confidence sets. There could be other algorithms with mistake IPOC bounds for this setting, but they they would likely require entirely different confidence sets.

5.5 Simulation Experiment

One important use case for certificates is to detect sudden performance drops when the distribution of contexts changes. For example, in a call center dialogue system, there can be a sudden increase of customers calling due to a certain regional outage. We demonstrate that certificates can identify such performance drops caused by context shifts. We consider a simulated MDP with 10 states, 40 actions and horizon 5 where rewards depend on a 10-dimensional context and let the distribution of contexts change after 2M episodes. As seen in Figure 5.1, this causes a spike in optimality gap as well as in the optimality certificates. While our certificates need to upper bound the optimality gap / contain the expected return in each episode up to a small failure probability, even for the worst case, our algorithm reliably can detect this sudden decrease of performance. In fact, the optimality certificates have a very high correlation of 0.94 with the unobserved optimality gaps.

One also may wonder if our algorithms leads to improvements over prior approaches in practice or only in the theoretical bounds. To help answer this, we present results in Section 5.12 both on analyzing the policy certificates provided, and examining `ORLC`'s performance in tabular MDPs versus other recent papers with similar regret Azar, Osband, and Munos, 2017 or PAC Dann, Lattimore, and Brunskill, 2017 bounds. Encouragingly in the small simulation MDPs considered, we find that our algorithms lead to faster learning and better performance. Therefore while our primary contribution is theoretical results, these simulations suggest the potential benefits of the ideas underlying our proposed framework and algorithms.

5.6 Related Work

The connection of IPOC to other frameworks is formally discussed in Section 5.3. Our algorithms essentially compute confidence bounds as in OFU methods, and then use those in model-based policy evaluation to

obtain policy certificates. There are many works on off-policy policy evaluation (e.g., Jiang and Li, 2016; Thomas and Brunskill, 2016; Mahmood, Yu, and Sutton, 2017), some of which provide non-asymptotic confidence intervals (e.g., Thomas, Theocharous, and Ghavamzadeh, 2015a; Thomas, Theocharous, and Ghavamzadeh, 2015b; Sajed, Chung, and White, 2018). However, these methods focus on the batch setting where a set of episodes sampled by fixed policies is given. Many approaches rely on importance weights that require stochastic data-collecting policies but most sample-efficient algorithms for which we would like to provide certificates deploy deterministic policies. One could treat previous episodes to be collected by one stochastic data-dependent policy but that introduces bias in the importance-weighting estimators that is not accounted for in the analyses.

Interestingly, there is very recent work (Zanette and Brunskill, 2019) that also observed the benefits of using lower bounds in optimism-based exploration in tabular episodic RL. Though both their and our work obtain improved theoretical results, the specific forms of the optimistic bonuses are distinct and the analyses differ in many parts (e.g., we provide (Uniform-)PAC and regret bounds instead of only regret bounds). Most importantly, our work provides policy certificate guarantees as a main contribution whereas that work focuses on problem-dependent regret bounds.

Approaches on safe exploration (Kakade and Langford, 2002; Pirotta et al., 2013; Thomas, Theocharous, and Ghavamzadeh, 2015b; Ghavamzadeh, Petrik, and Chow, 2016) guarantee monotonically increasing performance by operating in a batch loop. Our work is orthogonal, as we are not restricting exploration but rather exposing its impact to the users and give them the choice to intervene.

5.7 Summary

We introduced policy certificates to improve accountability in RL by enabling users to intervene if the guaranteed performance is deemed inadequate. Bounds in our new theoretical framework IPOC ensure that certificates indeed bound the expected return and suboptimality in each episode and prescribe the rate at which certificates and policy improve. By combining optimism-based exploration with model-based policy evaluation, we have created two algorithms for RL with policy certificates, including for tabular MDPs with side information. For tabular MDPs, we demonstrated that policy certificates help optimism-based policy learning and vice versa. As a result, our new algorithm is the first to achieve minimax-optimal PAC bounds up to lower-order terms for tabular episodic MDPs, and, also the first to have both, minimax PAC and regret bounds, for this setting.

5.8 Proofs of Relationship of IPOC Bounds to Other Bounds

5.8.1 Proof of Proposition 51

Proof of Proposition 51. We prove each part separately:

Part 1: With probability at least $1-\delta$, for all T, the regret is bounded as

$$\sum_{k=1}^{T} \Delta_k \leq \sum_{k=1}^{T} \epsilon_k \leq F(W, T, \delta)$$

where the first inequality follows from condition 1 and the second from condition 2a. Hence, the algorithm satisfied a high-probability regret bound $F(W, T, \delta)$ uniformly for all T.

Part 2: By assumption, the cumulative sum of certificates is bounded by $F(W, T, \delta) = \sum_{p=0}^{N} (C_p(W, \delta)T)^{\frac{p}{p+1}}$. Since the minimum is always smaller than the average, the smallest certificates output in the first T episodes

is at most

$$\min_{k\in[T]} \epsilon_k \leq \frac{\sum_{k=1}^{T} \epsilon_k}{T} \leq \frac{F(W,T,\delta)}{T} = \sum_{p=0}^{N} C_p(W,\delta)^{\frac{p}{p+1}} T^{-\frac{1}{p+1}}.$$

For $T \geq \frac{C_p(W,\delta)^p (N+1)^{p+1}}{\epsilon^{p+1}}$ we can bound

$$C_p(W,\delta)^{\frac{p}{p+1}} T^{-\frac{1}{p+1}} \leq C_p(W,\delta)^{\frac{p}{p+1}} \left(\frac{C_p(W,\delta)^p (N+1)^{p+1}}{\epsilon^{p+1}} \right)^{-\frac{1}{p+1}} \leq \frac{\epsilon}{N}.$$

As a result, for $T \geq \sum_{p=0}^{N} \frac{C_p(W,\delta)^p (N+1)^{p+1}}{\epsilon^{p+1}} \geq \max_{p\in[N]\cup\{0\}} \frac{C_p(W,\delta)^p (N+1)^{p+1}}{\epsilon^{p+1}}$, we can ensure that $\frac{F(W,T,\delta)}{T} \leq \epsilon$, which completes the proof. □

5.8.2 Proof of Proposition 52

Proof of Proposition 52. We prove each part separately:

Part 1:

By Definition 50 and the assumption, we have that with probability at least $1-\delta$ for all $\epsilon > 0$, it holds

$$\sum_{k}^{\infty} \mathbf{1}\{\Delta_k > \epsilon\} \leq \sum_{k}^{\infty} \mathbf{1}\{\epsilon_k > \epsilon\} \leq F(W,\epsilon,\delta),$$

where the first inequality follows from condition 1 of IPOC and the second from condition 2b. This proves that the algorithm also satisfies a Uniform-PAC bound as defined by Dann, Lattimore, and Brunskill (2017).

Part 2: Since by definition of IPOC, with probability at least $1-\delta$ for all $\epsilon > 0$, the algorithm can output a certificate $\epsilon_k > \epsilon$ at most $F(W,\epsilon,\delta)$ times. By the pigeon hole principle, the algorithm has to output at least one certificate $\epsilon_k \leq \epsilon$ in the first $F(W,\epsilon,\delta)+1$ episodes.

Part 3: This part of the proof is based on the proof of Theorem A.1 in Dann, Lattimore, and Brunskill (2017). For convenience, we omit the dependency of $\tilde{C}$ and C_p on W and δ in the following. We assume

$$F(W,\epsilon,\delta) = \sum_{p=1}^{N} \frac{C_p}{\epsilon^p} \left(\ln \frac{\bar{C}}{\epsilon} \right)^{np} = \sum_{p=1}^{N} C_p g(\epsilon)^p$$

where $\bar{C}$ is chosen so that for all $p \in [N]$ holds $\bar{C}^p \geq \Delta_{\max} C_p$ as well as $\bar{C} \geq \tilde{C}$. We also defined $g(\epsilon) := \frac{1}{\epsilon} \left(\ln \frac{\bar{C}}{\epsilon} \right)^n$. Consider now the cumulative sum of certificates after T episodes. We distinguish two cases:

Case 1: $T \leq \max_{p\in[N]} \frac{e^p}{\bar{C}^p} N C_p$**.** Note that $e = \exp(1)$ here. We use the fact that all certificates are at most $\Delta_{\max}$ and bound

$$\sum_{k=1}^{T} \epsilon_k \leq \Delta_{\max} T \leq \max_{p\in[N]} \frac{e^p}{\bar{C}^p} N C_p \Delta_{\max} \leq N e^N$$

where the final inequality leverages the assumption on $\bar{C}$.

Case 2: $T \geq \max_{p\in[N]} \frac{e^p}{\bar{C}^p} N C_p$**.** The mistake bound $F(W,\epsilon,\delta)$ is monotonically decreasing for $\epsilon \in (0, \Delta_{\max}]$. If T is large enough, we can therefore find an $\epsilon_{\min} \in (0, \Delta_{\max}]$ such that $F(W,\epsilon,\delta) \leq T$ for all $\epsilon \in (\epsilon_{\min}, \Delta_{\max}]$. The cumulative sum of certificates can then be bounded as follows

$$\sum_{k=1}^{T} \epsilon_k \leq T\epsilon_{\min} + \int_{\epsilon_{\min}}^{\Delta_{\max}} F(W,\epsilon,\delta) d\epsilon. \tag{5.1}$$

This bound assumes the worst case where the algorithm first outputs as many $\epsilon_k = \Delta_{\max}$ as allowed and subsequently smaller certificates as controlled by the mistake bound.

Before further simplifying this expression, we claim that

$$\epsilon_{\min} = \frac{\ln\left(\bar{C}\min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}\right)^n}{\min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}}$$

satisfies the desired property $F(W, \epsilon_{\min}, \delta) \leq T$. To see this, it is sufficient to show that $g(\epsilon_{\min}) \leq \min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}$, as it implies

$$\sum_{p=1}^{N} C_p g(\epsilon_{\min})^p = \sum_{p=1}^{N} C_p \min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{p/p} \leq \sum_{p=1}^{N} \frac{T}{N}\frac{C_p}{C_p} = T.$$

To show the bound on $g(\epsilon_{\min})$, we verify that for any $x \geq \exp(1)/\bar{C}$

$$g\left(\frac{(\ln(\bar{C}x))^n}{x}\right) = x\frac{\ln\left(\frac{\bar{C}x}{\ln(x\bar{C})^n}\right)^n}{\ln(\bar{C}x)^n} = x\,\frac{1}{\ln(\bar{C}x)^n}\left(\ln(\bar{C}x) - n\ln(\ln(x\bar{C}))\right)^n \leq x.$$

Since $\epsilon_{\min}$ has this form for $x = \min_{p\in[N]}\left(\frac{T}{(N)C_p}\right)^{1/p}$ and $\min_{p\in[N]}\left(\frac{T}{(N)C_p}\right)^{1/p} \geq \frac{e}{\bar{C}}$ by case assumption on T, $\epsilon_{\min}$ satisfies the desired property $F(W, \epsilon_{\min}, \delta) \leq T$.

We now go back to Equation (5.1) and simplify it to

$$\begin{aligned}
\sum_{k=1}^{T}\epsilon_k \leq& T\epsilon_{\min} + \int_{\epsilon_{\min}}^{\Delta_{\max}} F(W,\epsilon,\delta)d\epsilon.\\
=& T\epsilon_{\min} + \sum_{p=1}^{N} C_p \int_{\epsilon_{\min}}^{\Delta_{\max}} g(\epsilon)^p d\epsilon\\
=& T\epsilon_{\min} + \sum_{p=1}^{N} C_p \int_{\epsilon_{\min}}^{\Delta_{\max}} \frac{1}{\epsilon^p}\ln\left(\frac{\bar{C}}{\epsilon}\right)^{np} d\epsilon\\
\leq& T\epsilon_{\min} + \sum_{p=1}^{N} C_p \ln\left(\frac{\bar{C}}{\epsilon_{\min}}\right)^{np} \int_{\epsilon_{\min}}^{\Delta_{\max}} \frac{1}{\epsilon^p} d\epsilon\\
=& T\epsilon_{\min} + C_1\left(\ln\frac{\bar{C}}{\epsilon_{\min}}\right)^n \ln\frac{\Delta_{\max}}{\epsilon_{\min}} + \sum_{p=2}^{N}\frac{C_p}{1-p}\left(\ln\frac{\bar{C}}{\epsilon_{\min}}\right)^{np}\left[\Delta_{\max}^{1-p} - \epsilon_{\min}^{1-p}\right]. \qquad (5.2)
\end{aligned}$$

For each term in the final expression, we show that it is $\tilde{O}\left(\sum_{p=1}^{N} C_p^{1/p} T^{\frac{p-1}{p}}\,\mathrm{polylog}(\bar{C}T)\right)$. Starting with

the first, we bound

$$\begin{aligned}
T\epsilon_{\min} =& \frac{T\ln\left(\bar{C}\min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}\right)^n}{\min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}} = \ln\left(\bar{C}\min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}\right)^n \max_{p\in[N]}\frac{TN^{1/p}C_p^{1/p}}{T^{1/p}} \\
\leq& \ln\left(\bar{C}\frac{T}{NC_1}\right)^n N\max_{p\in[N]} T^{\frac{p-1}{p}}C_p^{1/p} \leq \ln\left(\bar{C}T\right)^n N\max_{p\in[N]} T^{\frac{p-1}{p}}C_p^{1/p} \\
=& \tilde{O}\left(\sum_{p=1}^N C_p^{1/p}T^{\frac{p-1}{p}}\operatorname{polylog}(\bar{C}T)\right).
\end{aligned}$$

For the second term, we start with bounding the inverse of $\epsilon_{\min}$ separately leveraging the case assumption on T:

$$\begin{aligned}
\frac{1}{\epsilon_{\min}} =& \min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}\frac{1}{\ln\left(\bar{C}\min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}\right)^n} \leq \min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p}\frac{1}{\ln\left(\bar{C}\min_{p\in[N]}\left(\frac{e^p}{C^p}\right)^{1/p}\right)^n} \\
\leq& \min_{p\in[N]}\left(\frac{T}{NC_p}\right)^{1/p} \leq T.
\end{aligned}$$

The second term of Equation (5.2) can now be upper bounded by:

$$C_1\left(\ln\frac{\bar{C}}{\epsilon_{\min}}\right)^n \ln\frac{\Delta_{\max}}{\epsilon_{\min}} \leq C_1\ln(\bar{C}T)^n\ln(\Delta_{\max}T) \leq C_1\ln(\bar{C}T)^{n+1} = \tilde{O}\left(\sum_{p=1}^N C_p^{1/p}T^{\frac{p-1}{p}}\operatorname{polylog}(\bar{C}T)\right)$$

where the last inequality leverages the definition of $\bar{C}$. Finally, consider the last term of Equation (5.2) for $p > 2$:

$$\begin{aligned}
&\frac{C_p}{1-p}\left(\ln\frac{\bar{C}}{\epsilon_{\min}}\right)^{np}\left[\Delta_{\max}^{1-p}-\epsilon_{\min}^{1-p}\right] = \frac{C_p}{p-1}\left(\ln\frac{\bar{C}}{\epsilon_{\min}}\right)^{np}\left[\epsilon_{\min}^{1-p}-\Delta_{\max}^{1-p}\right] \leq \frac{C_p}{p-1}\ln(\bar{C}T)^{np}\epsilon_{\min}^{1-p} \\
=& \frac{C_p}{p-1}\ln(\bar{C}T)^{np}(\epsilon_{\min}^{-1})^{p-1} \leq \frac{C_p}{p-1}\ln(\bar{C}T)^{np}\left(\frac{T}{NC_p}\right)^{(p-1)/p} \leq \ln(\bar{C}T)^{np}C_p^{1/p}T^{(p-1)/p} \\
=& \tilde{O}\left(\sum_{p=1}^N C_p^{1/p}T^{\frac{p-1}{p}}\operatorname{polylog}(\bar{C}T)\right).
\end{aligned}$$

Combining all bounds above we obtain that

$$\sum_{k=1}^T \epsilon_k \leq \tilde{O}\left(\sum_{p=1}^N C_p^{1/p}T^{\frac{p-1}{p}}\operatorname{polylog}(\bar{C}T)\right) \leq \tilde{O}\left(\sum_{p=1}^N C_p^{1/p}T^{\frac{p-1}{p}}\operatorname{polylog}(\Delta_{\max},\tilde{C},T)\right).$$

□

5.9 Theoretical Analysis of Algorithm 4 for Tabular MDPs

To ease the presentation, we chose valid but slightly loose confidence widths $\psi_{k,h}$ in Algorithm 4. In Algorithm 6 is a version of ORLC with slightly tighter confidence intervals. It uses different width for upper $\tilde{\psi}_{k,h}$ and lower $\underset{\sim}{\psi}_{k,h}$ confidence widths and is expected to perform better empirically. The IPOC analysis below applies to both algorithms.

To ease the notation in the analysis of ORLC, we first introduce several helpful definitions:

$$
\begin{aligned}
w_{k,h}(s,a) =& \mathbb{E}\left[\mathbf{1}\{s_{k,h}=s, a_{k,h}=a\} \,\middle|\, a_{k,1:h}\sim\pi_k, s_{k,1}=s_{k,1}\right] \\
w_k(s,a) =& \sum_{h=1}^{H} w_{k,h}(s,a) \\
w_{\min} =& \frac{\epsilon c_\epsilon}{S(A\wedge H)H} \quad \text{where } c_\epsilon = e^{-6}/4 \\
L_k =& \{(s,a)\in\mathcal{S}\times\mathcal{A} : w_k(s,a)\geq w_{\min}\} \\
\operatorname{llnp}(x) =& \ln(\ln(\max\{x,e\})) \\
\operatorname{rng}(x) =& \max(x)-\min(x) \quad \text{for vector } x \\
\delta' =& \frac{\delta}{5SAH+4SA+4S^2A} \\
\phi(n) =& 1\wedge\sqrt{\frac{0.52}{n}\left(1.4\operatorname{llnp}(2n)+\log\frac{5.2}{\delta'}\right)}.
\end{aligned}
\tag{5.3}
$$

The proof proceeds in four main steps. First, we define all concentration arguments needed in the form of a failure event and gives an upper bound for its probability. We then prove that all value estimates $\tilde{Q}$ and $\underset{\sim}{Q}$ are indeed optimistic / pessimistic outside the failure event. In a third step, we show a bound on the certificates in the form of a weighted sum of decreasing terms and finally apply a refined pigeon hole argument to bound the number of times this bound can exceed a given threshold.

5.9.1 Failure event and all probabilistic arguments

The failure event is defined as $F = F^N \cup F^P \cup F^{PE} \cup F^V \cup F^{VE} \cup F^{L1} \cup F^R$ where

$$
\begin{aligned}
F^R =& \{\exists\, k,s,a:\ |\hat{r}_k(s,a)-r(s,a)|\geq\phi(n_k(s,a))\} \\
F^V =& \left\{\exists k,s,a,h:\ |(\hat{P}_k(s,a)-P(s,a))V^\star_{h+1}|\geq \operatorname{rng}(V^\star_{h+1})\phi(n_k(s,a))\right\} \\
F^{VE} =& \left\{\exists k,s,a,h:\ |(\hat{P}_k(s,a)-P(s,a))V^\star_{h+1}|\geq \sqrt{4\hat{P}_k(s,a)[(V^\star_{h+1}-P(s,a)V^\star_{h+1})^2]}\phi(n_k(s,a))\right. \\
&\qquad\left. + 4.66\operatorname{rng}(V^\star_{h+1})\phi(n_k(s,a))^2\right\} \\
F^P =& \left\{\exists\, k,s,s',a:\ |\hat{P}_k(s'|s,a)-P(s'|s,a)|\geq\sqrt{4P(s'|s,a)}\phi(n_k(s,a))+1.56\phi(n_k(s,a))^2\right\} \\
F^{PE} =& \left\{\exists\, k,s,s',a:\ |\hat{P}_k(s'|s,a)-P(s'|s,a)|\geq\sqrt{4\hat{P}_k(s'|s,a)}\phi(n_k(s,a))+4.66\phi(n_k(s,a))^2\right\} \\
F^{L1} =& \left\{\exists\, k,s,a:\ \|\hat{P}_k(s,a)-P(s,a)\|_1\geq 2\sqrt{S}\phi(n_k(s,a))\right\}
\end{aligned}
$$

Algorithm 6: `ORLC` with tighter confidence widths

Input : failure tolerance $\delta \in (0, 1]$

$\phi(n) = 1 \wedge \sqrt{\frac{0.52}{n}\left(1.4 \ln\ln(e \vee n) + \ln\frac{26SA(H+1+S)}{\delta}\right)}$; $\quad \tilde{V}_{k,H+1}(s) = 0$; $\quad \underset{\sim}{V}_{k,H+1}(s) = 0 \quad \forall s \in \mathcal{S}, k \in \mathbb{N}$;

for $k = 1, 2, 3, \ldots$ **do**

  for $s', s \in \mathcal{S}, a \in \mathcal{A}$ **do** `// update empirical model and number of observations`

    $n_k(s,a) = \sum_{i=1}^{k-1}\sum_{h=1}^{H} \mathbf{1}\{s_{i,h} = s, a_{i,h} = a\}$;

    $\hat{r}_k(s,a) = \frac{1}{n_k(s,a)}\sum_{i=1}^{k-1}\sum_{h=1}^{H} r_{i,h}\mathbf{1}\{s_{i,h} = s, a_{i,h} = a\}$;

    $\hat{P}_k(s'|s,a) = \frac{1}{n_k(s,a)}\sum_{i=1}^{k-1}\sum_{h=1}^{H} \mathbf{1}\{s_{i,h} = s, a_{i,h} = a, s_{i,h+1} = s'\}$

  for $h = H$ **to** 1 ***and*** $s \in \mathcal{S}$ **do** `// optimistic planning leveraging upper and lower confidence bounds`

    for $a \in \mathcal{A}$ **do**

      $\tilde{\psi}_{k,h}(s,a) = \min\Big\{(V_{h+1}^{\max}+1)\phi(n_k(s,a)),$

        $(1+\sqrt{12})\sqrt{\sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}) + \hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})^2}\phi(n_k(s,a))$

          $+8.13V_{h+1}^{\max}\phi(n_k(s,a))^2,$

        $(1+\sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}))\phi(n_k(s,a)) + \frac{1}{H}\hat{P}(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})$

          $+(20.13H\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1)\phi(n_k(s,a))^2\Big\}$;

      $\underset{\sim}{\psi}_{k,h}(s,a) = \min\Big\{(2\sqrt{S}V_{h+1}^{\max}+1)\phi(n_k(s,a)),$

        $\left(V_{h+1}^{\max} + 1 + 2\sqrt{P_k(s,a)}(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})\right)\phi(n_k(s,a))$

          $+4.66\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1\phi(n_k(s,a))^2,$

        $\sqrt{12}\sqrt{\sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}) + \hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})^2}\phi(n_k(s,a))$

          $+\left(1 + 2\sqrt{P_k(s,a)}(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})\right)\phi(n_k(s,a))$

          $+(8.13V_{h+1}^{\max} + 4.66\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1)\phi(n_k(s,a))^2,$

        $(1+\sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}))\phi(n_k(s,a)) + \frac{1}{H}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})$

          $+(8.13V_{h+1}^{\max} + (32H + 4.66)\|\tilde{V}_{k,h+1} -$

        $\underset{\sim}{V}_{k,h+1}\|_1)\phi(n_k(s,a))^2\Big\}$;

      $\tilde{Q}_{k,h}(s,a) = 0 \vee \left(\hat{r}_k(s,a) + \hat{P}_k(s,a)\tilde{V}_{k,h+1} + \tilde{\psi}_{k,h}(s,a)\right) \wedge V_h^{\max}$;

      $\underset{\sim}{Q}_{k,h}(s,a) = 0 \vee \left(\hat{r}_k(s,a) + \hat{P}_k(s,a)\underset{\sim}{V}_{k,h+1} - \underset{\sim}{\psi}_{k,h}(s,a)\right) \wedge V_h^{\max}$;

    $\pi_k(s,h) = \operatorname{argmax}_a \tilde{Q}_{k,h}(s,a)$; $\quad \tilde{V}_{k,h}(s) = \tilde{Q}_{k,h}(s, \pi_k(s,h))$; $\quad \underset{\sim}{V}_{k,h}(s) = \underset{\sim}{Q}_{k,h}(s, \pi_k(s,h))$;

  output policy π_k with certificate $\epsilon_k = \tilde{V}_1(s_{k,1}) - \underset{\sim}{V}_1(s_{k,1})$;

  sample episode k with policy π_k;

$$F^N = \left\{ \exists\, k, s, a :\, n_k(s,a) < \frac{1}{2}\sum_{i<k} w_i(s,a) - H \ln \frac{SAH}{\delta'} \right\}.$$

The following lemma shows that F has low probability.

Lemma 56. *For any parameter $\delta' > 0$, the probability of each failure event is bounded as*

$$\begin{aligned} &\mathbb{P}\left(F^V\right) \leq 2SAH\delta' && \mathbb{P}(F^{VE}) \leq 2SAH\delta' && \mathbb{P}(F^R) \leq 2SA\delta' && \mathbb{P}(F^P) \leq 2S^2A\delta' \\ &\mathbb{P}(F^{PE}) \leq 2S^2A\delta' && \mathbb{P}(F^{L1}) \leq 2SA\delta' && \mathbb{P}(F^N) \leq SAH\delta'. \end{aligned}$$

The failure probability is thus bounded by $\mathbb{P}(F) \leq \delta'(5SAH + 4SA + 4S^2A) = \delta$, since we set $\delta' = \frac{\delta}{5SAH+4SA+4S^2A}$.

Proof. When proving that these failure events indeed have low probability, we need to consider sequences of random variables whenever a particular state and action pair (s,a) was observed. Since the number of times a particular (s,a) was observed as well as in which episodes, is random, we have to treat this carefully. To that end, we first define σ-fields $\mathcal{G}_i^{s,a}$ which correspond to all observations up to exactly i observations of that (s,a)-pair.

Consider a fixed $(s,a) \in \mathcal{S} \times \mathcal{A}$, and denote by $\mathcal{F}_{(k-1)H+h}$ the sigma-field induced by the first $k-1$ episodes and the k-th episode up to $s_{k,h}$ and $a_{k,h}$ but not $s_{k,h+1}$. Define

$$\tau_i = \inf \left\{ (k-1)H + h \;:\; \sum_{j=1}^{k}\sum_{t=1}^{H} \mathbf{1}\{s_{j,t} = s, a_{j,t} = a\} + \sum_{t=1}^{h} \mathbf{1}\{s_{k,t} = s, a_{k,t} = a\} \geq i \right\}$$

to be the index where (s,a) was observed the ith time. Note that τ_i are stopping times with respect to $\mathcal{F}_i$. Hence, the stopped version $\mathcal{G}_i^{s,a} = \mathcal{F}_{\tau_i} = \{A \in \mathcal{F}_\infty \,:\, A \cap \{\tau_i \leq t\} \in \mathcal{F}_t \;\forall\, t \geq 0\}$ is a filtration as well. We are now ready to bound the probability of each failure event.

Failure event F^V: For a fixed $s \in \mathcal{S}, a \in \mathcal{A}, h \in [H]$, we define $X_i = \frac{1}{\operatorname{rng}(V_{h+1}^\star)}(V_{h+1}^\star(s_i') - P(s,a)V_{h+1}^\star)\mathbf{1}\{\tau_i < \infty\}$ where s_i' is the value of the successor state when (s,a) was observed the ith time (formally $s_{k,j+1}$ with $k = \lfloor \tau_i/H \rfloor$ and $j = \tau_i \mod H$) or arbitrary, if $\tau_i = \infty$).

By the Markov property of the MDP, X_i is a martingale difference sequence with respect to the filtration $\mathcal{G}_i^{s,a}$, that is, $\mathbb{E}[X_i | \mathcal{G}_{i-1}^{s,a}] = 0$. Furthermore, it is bounded as

$$X_i \in \left[\frac{\min V_{h+1}^\star - P(s,a)V_{h+1}^\star}{\operatorname{rng}(V_{h+1}^\star)}, \frac{\max V_{h+1}^\star - P(s,a)V_{h+1}^\star}{\operatorname{rng}(V_{h+1}^\star)} \right]$$

where the range is

$$\frac{\max V_{t+1}^\star - P(s,a)V_{t+1}^\star}{\operatorname{rng}(V_{t+1}^\star)} - \frac{\min V_{t+1}^\star - P(s,a)V_{t+1}^\star}{\operatorname{rng}(V_{t+1}^\star)} = \frac{\operatorname{rng} V_{t+1}^\star}{\operatorname{rng} V_{t+1}^\star} = 1.$$

Hence, $S_j = \sum_{i=1}^{j} X_i$ with $V_j = j/4$ satisfies Assumption 1 by Howard et al. (2018) (see Hoeffding I entry in Table 2 therein) with any sub-Gaussian boundary ψ_G. The same is true for the sequence $-S_k$. Using the sub-Gaussian boundary from Corollary 71, we get that with probability at least $1 - 2\delta'$ for all $n \in \mathbb{N}$

$$\left| \sum_{i=1}^{n} X_i \right| \leq 1.44 \sqrt{\frac{n}{4}\left(1.4 \operatorname{llnp}(n/2) + \log \frac{5.2}{\delta'}\right)}. \tag{5.4}$$

Since that holds after each observation, this is in particular true before each episode $k+1$ where (s,a) has been observed $n_k(s,a)$ times so far. We can now rewrite the value of the martingale as

$$\left|\sum_{i=1}^{n_k(s,a)} X_i\right| = \frac{1}{\operatorname{rng}(V_{h+1}^\star)}\left|\sum_{i=1}^{n_k(s,a)} (V_{h+1}^\star(s_i') - P(s,a)V_{h+1}^\star)\right| = \frac{n_k(s,a)}{\operatorname{rng}(V_{h+1}^\star)}\left|\frac{\sum_{i=1}^{n_k(s,a)} V_{h+1}^\star(s_i')}{n_k(s,a)} - P(s,a)V_{h+1}^\star\right|$$

$$= \frac{n_k(s,a)}{\operatorname{rng}(V_{h+1}^\star)}\left|\hat{P}_k(s,a)V_{h+1}^\star - P(s,a)V_{h+1}^\star\right| \tag{5.5}$$

and combine this equation with Equation (5.4) to realize that for all k

$$\begin{aligned}|(\hat{P}_k(s,a) - P(s,a))V_{h+1}^\star| &\le \operatorname{rng}(V_{h+1}^\star)\sqrt{\frac{0.52}{n_k(s,a)}\left(1.4\operatorname{llnp}\left(\frac{n_k(s,a)}{2}\right) + \log\frac{5.2}{\delta'}\right)} \\ &\le \operatorname{rng}(V_{h+1}^\star)\sqrt{\frac{0.52}{n_k(s,a)}\left(1.4\operatorname{llnp}\left(2n_k(s,a)\right) + \log\frac{5.2}{\delta'}\right)}\end{aligned}$$

holds with probability at least $1-2\delta'$. Since in addition $|(\hat{P}_k(s,a) - P(s,a))^\top V_{h+1}^\star| \le \operatorname{rng}(V_{h+1}^\star)$ at all times, we can bound $|(\hat{P}_k(s,a) - P(s,a))^\top V_{h+1}^\star| \le \operatorname{rng}(V_{h+1}^\star)\phi(n_k(s,a))$ which shows that F^V has low probability for a single (s,a,h) triple. Applying a union bound over all $h \in [H]$ and $s,a \in \mathcal{S}\times\mathcal{A}$, we can conclude that $\mathbb{P}(F^V) \le 2SAH\delta'$.

Failure event F^{VE}: As an alternative to the Hoeffding-style bound above, we can use Theorem 5 by Howard et al. (2018) with the sub-exponential bound from Corollary 71 and the predictable sequence $\hat{X}_i = 0$. This gives that with $V_n = \sum_{i=1}^n (X_i - \hat{X}_i)^2 = \sum_{i=1}^n X_i^2 \le n$ it holds with probability at least $1-2\delta'$ for all $n \in \mathbb{N}$

$$\begin{aligned}\left|\sum_{i=1}^n X_i\right| &\le 1.44\sqrt{V_n\left(1.4\operatorname{llnp}(2V_n) + \log\frac{5.2}{\delta'}\right)} + 2.42\left(1.4\operatorname{llnp}(2V_n) + \log\frac{5.2}{\delta}\right) \\ &\le 1.44\sqrt{V_n\left(1.4\operatorname{llnp}(2n) + \log\frac{5.2}{\delta'}\right)} + 2.42\left(1.4\operatorname{llnp}(2n) + \log\frac{5.2}{\delta}\right).\end{aligned}$$

Hence, in particular before each episode k when there are $n_k(s,a)$ observations, we have by the identity in Equation (5.5) that in the same event as above

$$\begin{aligned}|(\hat{P}_k(s,a) - P(s,a))^\top V_{h+1}^\star| \le &\sqrt{\frac{4\operatorname{rng}(V_{h+1}^\star)^2 V_{n_k(s,a)}}{n_k(s,a)}}\sqrt{\frac{0.52}{n_k(s,a)}\left(1.4\operatorname{llnp}\left(2n_k(s,a)\right) + \log\frac{5.2}{\delta'}\right)} \\ &+ \frac{2.42\operatorname{rng}(V_{h+1}^\star)}{0.52}\frac{0.52}{n_k(s,a)}\left(1.4\operatorname{llnp}(2n_k(s,a)) + \log\frac{5.2}{\delta}\right).\end{aligned}$$

Similar to Equation (5.5), the following identity holds

$$\frac{4\operatorname{rng}(V_{h+1}^\star)^2 V_{n_k(s,a)}}{n_k(s,a)} = 4\hat{P}_k(s,a)[(V_{h+1}^\star - P(s,a)V_{h+1}^\star)^2]$$

and since $|(\hat{P}_k(s,a) - P(s,a))V_{h+1}^\star| \le 4.66\operatorname{rng}(V_{h+1}^\star)$ at any time

$$|(\hat{P}_k(s,a) - P(s,a))^\top V_{h+1}^\star| \le \sqrt{4\hat{P}_k(s,a)[(V_{h+1}^\star - P(s,a)V_{h+1}^\star)^2]\phi(n_k(s,a))^2} + 4.66\operatorname{rng}(V_{h+1}^\star)\phi(n_k(s,a))^2.$$

This shows that F^{VE} has probability at most $1-2\delta'$ for a specific (s,a,t) triple. Hence, with an appropriate union bound, we get the desired bound $\mathbb{P}(F^{VE}) \leq 2SAH\delta'$.

Failure event F^R: For this event, we define X_i as $X_i = (r'_i - r(s,a))\mathbf{1}\{\tau_i < \infty\}$ where r'_i is the immediate reward when s,a was observed the ith time (formally $r_{j,l}$ with $j = \lfloor \tau_i/H \rfloor$ and $l = \tau_i \mod H$ or arbitrary (e.g. 1), if $\tau_i = \infty$).

Similar to above, X_i is a martingale w.r.t. $\mathcal{G}_i^{s,a}$ and by assumption is bounded as $X_i \in [-r(s,a), 1 - r(s,a)]$, i.e., has a range of 1. Therefore, $S_n = \sum_{i=1}^n X_i$ under the current definition with $V_n = n/4$ satisfies Assumption 1 by Howard et al. (2018) and Corollary 71 gives that with probability at least $1-2\delta'$ for all $n \in \mathbb{N}$

$$\left|\sum_{i=1}^n X_i\right| \leq 1.44\sqrt{\frac{n}{4}\left(1.4\,\mathrm{llnp}(n/2) + \log\frac{5.2}{\delta'}\right)}.$$

Identical to above, this implies that with probability at least $1-2\delta'$ for all episodes $k \in \mathbb{N}$ it holds that $|\hat{r}_k(s,a) - r(s,a)| \leq \phi(n_k(s,a))$ for this particular s,a. Applying a union bound over $\mathcal{S}\times\mathcal{A}$ finally yields that $\mathbb{P}(F^R) \leq 2SA\delta'$.

Failure event F^P: In addition to s,a, consider a fixed $s' \in \mathcal{S}$. We here define X_i as $X_i = (\mathbf{1}\{s' = s'_i\} - P(s'|s,a))\mathbf{1}\{\tau_i < \infty\}$ where s'_i is the successor state when s,a was observed the ith time (formally $s_{k,j}$ with $k = \lfloor \tau_i/H \rfloor$ and $j = \tau_i \mod H$) or arbitrary, if $\tau_i = \infty$). By the Markov property, X_i is a martingale with respect to $\mathcal{G}_i^{s,a}$ and is bounded in $[-1,1]$.

Hence, $S_n = \sum_{i=1}^n X_i$ with $V_n = \sum_{i=1}^n \mathbb{E}[X_i^2|\mathcal{G}_{i-1}^{s,a}] = P(s,a)(\mathbf{1}\{s'=\cdot\} - P(s'|s,a))^2 \sum_{i=1}^n \mathbf{1}\{\tau_i < \infty\} \leq n$ satisfies Assumption 1 by Howard et al. (2018) (see Bennett entry in Table 2 therein) with sub-Gaussian ψ_P. The same is true for the sequence $-S_n$. Using Corollary 71, we get that with probability at least $1-2\delta'$ for all $n \in \mathbb{N}$

$$\begin{aligned}
|S_n| \leq& 1.44\sqrt{V_n\left(1.4\,\mathrm{llnp}(2V_n) + \log\frac{5.2}{\delta'}\right)} + 0.81\left(1.4\,\mathrm{llnp}(2V_n) + \log\frac{5.2}{\delta}\right) \\
\leq& 1.44\sqrt{V_n\left(1.4\,\mathrm{llnp}(2n) + \log\frac{5.2}{\delta'}\right)} + 0.81\left(1.4\,\mathrm{llnp}(2n) + \log\frac{5.2}{\delta}\right).
\end{aligned}$$

Hence, in particular after each episode k, we have in the same event because $S_{n_k(s,a)} = n_k(s,a)(\hat{P}_k(s'|s,a) - P(s'|s,a))$ that

$$\begin{aligned}
|\hat{P}_k(s'|s,a) - P(s'|s,a)| \leq& 1.44\sqrt{\frac{V_n}{0.52 n_k(s,a)}}\sqrt{\frac{0.52}{n_k(s,a)}\left(1.4\,\mathrm{llnp}(2n_k(s,a)) + \log\frac{5.2}{\delta'}\right)} \\
&+ \frac{0.81}{0.52}\frac{0.52}{n_k(s,a)}\left(1.4\,\mathrm{llnp}(2n_k(s,a)) + \log\frac{5.2}{\delta}\right).
\end{aligned}$$

Combining this bound with $|\hat{P}_k(s'|s,a) - P(s'|s,a)| \leq 1.56$ gives the inequality $|\hat{P}_k(s'|s,a) - P(s'|s,a)| \leq \sqrt{\frac{1.44^2 V_n}{0.52 n_k(s,a)}}\phi(n_k(s,a)) + 1.56\phi(n_k(s,a))^2$. It remains to bound the first coefficient as

$$\begin{aligned}
\frac{1.44^2 V_n}{0.52 n_k(s,a)} \leq& 4P(s,a)(\mathbf{1}\{s'=\cdot\} - P(s'|s,a))^2 = 4P(s,a)\mathbf{1}\{s'=\cdot\}^2 - 4P(s'|s,a)^2 \\
=& 4P(s'|s,a) - 4P(s'|s,a)^2 \leq 4P(s'|s,a).
\end{aligned}$$

Hence, for a fixed s', s, a, with probability at least $1 - \delta'$ the following inequality holds for all episodes k

$$|\hat{P}_k(s'|s,a) - P(s'|s,a)| \leq \sqrt{4P(s'|s,a)}\phi(n_k(s,a)) + 1.56\phi(n_k(s,a))^2.$$

Applying a union bound over $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, we get that $\mathbb{P}(F^P) \leq 2S^2A\delta'$.

Failure event F^{PE}: The bound in F^P uses the predictable variance of X_i which eventually leads to a dependency on the unknown $P(s'|s,a)$ in the bound. In F^{PE}, the bound instead depends on the observed $P_k(s'|s,a)$. To achieve that bound, we use Theorem 5 by Howard et al. (2018) in combination with Corollary 71, similar to event F^{VE}. For the same definition of X_i as in F^P, we then get that with probability at least $1 - 2\delta'$ for all $n \in \mathbb{N}$

$$\begin{aligned}|S_n| \leq& 1.44\sqrt{V_n\left(1.4\,\mathrm{llnp}(2V_n) + \log\frac{5.2}{\delta'}\right)} + 2.42\left(1.4\,\mathrm{llnp}(2V_n) + \log\frac{5.2}{\delta}\right)\\ \leq& 1.44\sqrt{V_n\left(1.4\,\mathrm{llnp}(2n) + \log\frac{5.2}{\delta'}\right)} + 2.42\left(1.4\,\mathrm{llnp}(2n) + \log\frac{5.2}{\delta}\right),\end{aligned}$$

where $V_n = \sum_{i=1}^n \left((X_i + P(s'|s,a))\mathbf{1}\{\tau_i < \infty\}\right)^2 \leq n$ (that is, we choose the predictable sequence as $\hat{X}_i = -P(s'|s,a)\mathbf{1}\tau_i < \infty$). Analogous to F^P, we have in the same event for all k

$$|\hat{P}_k(s'|s,a) - P(s'|s,a)| \leq \sqrt{\frac{1.44^2 V_n}{0.52 n_k(s,a)}}\phi(n_k(s,a)) + 4.66\phi(n_k(s,a))^2$$

and the first coefficient can be written as

$$\frac{1.44^2 V_n}{0.52 n_k(s,a)} \leq \frac{4}{n_k(s,a)} \sum_{i=1}^{n_k(s,a)} (\mathbf{1}\{s' = s_i'\})^2 \mathbf{1}\{\tau_i < \infty\})^2 = 4\hat{P}_k(s'|s,a).$$

After applying a union bound over all s', s, a, we get the desired failure probability bound $\mathbb{P}(F^{PE}) \leq 2S^2A\delta'$.

Failure event F^{L1}: Consider a fix $s \in \mathcal{S}, a \in \mathcal{A}$ and $\mathcal{B} \subseteq \mathcal{S}$ and define $X_i = (\mathbf{1}\{s_i' \in \mathcal{B}\} - P(s' \in \mathcal{B}|s,a))\mathbf{1}\{\tau_i < \infty\}$. In complete analogy to F^R, we can show that with probability at least $1 - 2\delta'$ the following bound holds for all episodes k

$$|\hat{P}_k(s' \in \mathcal{B}|s,a) - P(s' \in \mathcal{B}|s,a)| \leq \phi(n_k(s,a)).$$

We can use this result with $\delta'/2^S$ in combination with union bound over all possible subsets $\mathcal{B} \subseteq \mathcal{S}$ to get that

$$\max_{\mathcal{B} \subseteq \mathcal{S}} |\hat{P}_k(s' \in \mathcal{B}|s,a) - P(s' \in \mathcal{B}|s,a)| \leq \sqrt{S}\phi(n_k(s,a)).$$

with probability at least $1 - 2\delta'$ for all k. Finally, the fact about total variation

$$\|p - q\|_1 = 2\max_{\mathcal{B} \subseteq \mathcal{S}} |p(\mathcal{B}) - q(\mathcal{B})|$$

as well as a union bound over $\mathcal{S} \times \mathcal{A}$ gives that with probability at least $1 - 2SA\delta'$ for all k, s, a it holds that $\|\hat{P}_k(s,a) - P(s,a)\|_1 \leq 2\sqrt{S}\phi(n_k(s,a))$, i.e., $\mathbb{P}(F^{L1}) \leq 2SA\delta'$.

Failure event F^N: Consider a fixed $s \in \mathcal{S}, a \in \mathcal{A}, t \in [H]$. We define $\mathcal{F}_k$ to be the sigma-field induced by the first $k-1$ episodes and $s_{k,1}$. Let X_k as the indicator whether s, a was observed in episode k at time t. The probability $\mathbb{P}(s = s_{k,t}, a = a_{k,t}|s_{k,1}, \pi_k)$ of whether $X_k = 1$ is $\mathcal{F}_k$-measurable and hence we can apply Lemma F.4 by Dann, Lattimore, and Brunskill (2017) with $W = \ln\frac{SAH}{\delta'}$ and obtain that $\mathbb{P}(F^N) \leq SAH\delta'$ after summing over all statements for $t \in [H]$ and applying the union bound over s, a, t. □

5.9.2 Admissibility of Certificates

We now show that the algorithm always gives a valid certificate in all episodes, outside the failure event F. We call its complement, F^c, the "good event". The following three lemmas prove the admissibility.

Lemma 57 (Lower bounds admissible)**.** *Consider event F^c and an episode k, time step $h \in [H]$ and $s, a \in \mathcal{S} \times \mathcal{A}$. Assume that $\tilde{V}_{k,h+1} \geq V^\star_{h+1} \geq V^{\pi_k}_{h+1} \geq \underset{\sim}{V}_{k,h+1}$ and that the lower confidence bound width is at least*

$$\underset{\sim}{\psi}_{k,h}(s,a) \geq \alpha \hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + \beta \phi(n_k(s,a))^2 + \gamma \phi(n_k(s,a))$$

where there are four possible choices for α, β and γ:

$$\alpha = 0 \quad \beta = 0 \quad \gamma = 2\sqrt{S} V^{\max}_{h+1} + 1 \quad \text{or} \tag{5.6}$$

$$\alpha = 0 \quad \beta = 4.66\|\rho\|_1 \quad \gamma = 2\left[\sqrt{\hat{P}_k(s,a)}\right]\rho + V^{\max}_{h+1} + 1 \quad \text{or} \tag{5.7}$$

$$\alpha = 0 \quad \beta = (8.13 V^{\max}_{h+1} + 4.66\|\rho\|_1) \quad \gamma = 1 + \sqrt{12}\sqrt{\sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}) + \hat{P}_k(s,a)\rho^2} + 2\left[\sqrt{\hat{P}_k(s,a)}\right]\rho \tag{5.8}$$

$$\alpha = \frac{1}{C} \quad \beta = (8.13 V^{\max}_{h+1} + (32C + 4.66)\|\rho\|_1) \quad \gamma = 1 + \sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}) \tag{5.9}$$

with $\rho = \tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}$ and for any $C > 0$. Then the lower confidence bound at time h is admissible, i.e., $Q^{\pi_k}_h(s,a) \geq \underset{\sim}{Q}_{k,h}(s,a)$.

Proof. We want to show that $Q^{\pi_k}_h(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \geq 0$. Since $Q^{\pi_k}_h \geq 0$, this quantity is non-negative when the Q-value bound is clipped, i.e., $\underset{\sim}{Q}_{k,h}(s,a) = 0$. The non-clipped case is left, in which

$$Q^{\pi_k}_h(s,a) - \underset{\sim}{Q}_{k,h}(s,a) = P(s,a)V^{\pi_k}_{h+1} + r(s,a) - \hat{r}_k(s,a) + \underset{\sim}{\psi}_{k,h}(s,a) - \hat{P}_k(s,a)\underset{\sim}{V}_{k,h+1}. \tag{5.10}$$

For the first coefficient choice from Equation (5.6), we rewrite this quantity as

$$\begin{aligned}
&Q^{\pi_k}_h(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \\
&= \underset{\sim}{\psi}_{k,h}(s,a) + P(s,a)(V^{\pi_k}_{h+1} - \underset{\sim}{V}_{k,h+1}) + (P(s,a) - \hat{P}_k(s,a))\underset{\sim}{V}_{k,h+1} + r(s,a) - \hat{r}_k(s,a)
\end{aligned}$$

using the induction hypothesis for the second term and applying Hölder's inequality to the third term

$$\geq \underset{\sim}{\psi}_{k,h}(s,a) + 0 - \|P(s,a) - \hat{P}_k(s,a)\|_1 \|\underset{\sim}{V}_{k,h+1}\|_\infty - |r(s,a) - \hat{r}_k(s,a)|$$

applying definition of the good event F^c to the last terms and using the first choice of coefficients for $\underset{\sim}{\psi}_{k,h}$

$$\geq 2\sqrt{S} V^{\max}_h \phi(n_k(s,a)) - 2\sqrt{S}\phi(n_k(s,a)) V^{\max}_{h+1} - \phi(n_k(s,a)) \geq 0\,.$$

This completes the proof for the first coefficient choice. It remain to show the same for the second and third coefficient choice. To that end, we rewrite the quantity in Equation (5.10) as

$$\begin{aligned}
&Q^{\pi_k}_h(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \\
&= \underset{\sim}{\psi}_{k,h}(s,a) + (P(s,a) - \hat{P}_k(s,a))(V^{\pi_k}_{h+1} - V^\star_{h+1}) + \hat{P}_k(s,a)(V^{\pi_k}_{h+1} - \underset{\sim}{V}_{k,h+1}) \\
&\quad + (P(s,a) - \hat{P}_k(s,a))V^\star_{h+1} + r(s,a) - \hat{r}_k(s,a)
\end{aligned}$$

using the induction hypothesis, we can infer that $\hat{P}_k(s,a)(V^{\pi_k}_{h+1} - \underset{\sim}{V}_{k,h+1}) \geq 0$ and get

$$\geq \underset{\sim}{\psi}_{k,h}(s,a) - |(P(s,a) - \hat{P}_k(s,a))(V^{\pi_k}_{h+1} - V^\star_{h+1})| - |(P(s,a) - \hat{P}_k(s,a))V^\star_{h+1}| - |r(s,a) - \hat{r}_k(s,a)|$$

applying definition of the good event F^c to the last term and reordering gives

$$\geq -|(P(s,a) - \hat{P}_k(s,a))(V_{h+1}^{\pi_k} - V_{h+1}^\star)| - |(P(s,a) - \hat{P}_k(s,a))V_{h+1}^\star| \\ - \phi(n_k(s,a)) + \underset{\sim}{\psi}_{k,h}(s,a). \tag{5.11}$$

We now first consider $|(P(s,a) - \hat{P}_k(s,a))(V_{h+1}^{\pi_k} - V_{h+1}^\star)|$ and bound it using Lemma 68 where we bind $f = V_{h+1}^\star - V_{h+1}^{\pi_k}$ and with $\|f\|_1 \leq \|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1$

$$|(P(s,a) - \hat{P}_k(s,a))(V_{h+1}^{\pi_k} - V_{h+1}^\star)| \\ \leq 4.66\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1\phi(n_k(s,a))^2 + 2\phi(n_k(s,a))\sqrt{\hat{P}_k(s,a)}(V_{h+1}^\star - V_{h+1}^{\pi_k})$$

and since $0 \leq V_{h+1}^\star - V_{h+1}^{\pi_k} \leq \tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}$ this is upper-bounded by

$$\leq 4.66\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1\phi(n_k(s,a))^2 + 2\phi(n_k(s,a))\sqrt{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) \tag{5.12}$$

and again by Lemma 68 we can get a nicer form for any $C > 0$ as follows

$$\leq \frac{1}{C}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) - (4C + 4.66)\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1\phi(n_k(s,a))^2. \tag{5.13}$$

After deriving runtime-computable bounds for $|(P(s,a) - \hat{P}_k(s,a))(V_{h+1}^{\pi_k} - V_{h+1}^\star)|$, it remains to upper-bound $|(P(s,a) - \hat{P}_k(s,a))V_{h+1}^\star|$ in Equation (5.11). Here, we can apply the definition of the failure event F^V and bound $|(P(s,a) - \hat{P}_k(s,a))V_{h+1}^\star| \leq V_{h+1}^{\max}\phi(n_k(s,a))$. Plugging this bound together with the bound from (5.12) back into (5.11) gives

$$Q_h^{\pi_k}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \geq -\left(V_{h+1}^{\max} + 1 + 2\sqrt{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})\right)\phi(n_k(s,a)) \\ - 4.66\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1\phi(n_k(s,a))^2 + \underset{\sim}{\psi}_{k,h}(s,a)$$

which is non-negative when we use the second coefficient choice from Equation (5.7) for $\underset{\sim}{\psi}_{k,h}$. Alternatively, we can apply the definition of the failure event F^{VE} which uses an empirical variance instead of the range of $V_{h+1}^\star$ and bound

$$|(P(s,a) - \hat{P}_k(s,a))V_{h+1}^\star| \\ \leq \sqrt{4\hat{P}_k(s,a)[(V_{h+1}^\star(\cdot) - P(s,a)V_{h+1}^\star)^2]}\phi(n_k(s,a)) + 4.66V_{h+1}^{\max}\phi(n_k(s,a))^2 \\ \leq \sqrt{12}\sqrt{\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})^2 + \sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})}\phi(n_k(s,a)) + 8.13V_{h+1}^{\max}\phi(n_k(s,a))^2 \tag{5.14} \\ \leq \sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\phi(n_k(s,a)) + \frac{1}{C}P_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) \\ + (8.13V_{h+1}^{\max} + 12C\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1)\phi(n_k(s,a))^2 \tag{5.15}$$

where we applied Lemma 60. Plugging the bound from (5.14) and (5.12) into (5.11) gives

$$Q_h^{\pi_k}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \geq -\sqrt{12}\sqrt{\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})^2 + \sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})}\phi(n_k(s,a)) \\ - \left(1 + 2\sqrt{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})\right)\phi(n_k(s,a)) \\ - (8.13V_{h+1}^{\max} + 4.66\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1)\phi(n_k(s,a))^2 + \underset{\sim}{\psi}_{k,h}(s,a).$$

Applying the coefficient choice from Equation (5.8) for $\psi_{k,h}$ shows that this bound becomes non-negative as well. Finally, we plug the bound from (5.15) and (5.13) into (5.11) to get

$$\begin{aligned}
&Q_h^{\pi_k}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \\
&\geq -\frac{2}{C}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) - (8.13V_{h+1}^{\max} + (16C + 4.66)\|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1)\phi(n_k(s,a))^2 \\
&\quad - (1 + \sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}))\phi(n_k(s,a)) + \underset{\sim}{\psi}_{k,h}(s,a).
\end{aligned}$$

We rebind $C \leftarrow 2C$ and use the last coefficient choice from Equation (5.9) for $\underset{\sim}{\psi}_{k,h}$ to show the above is non-negative. Hence, we have shown that for all choices for coefficients $Q_h^{\pi_k}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \geq 0$. □

Lemma 58 (Upper bounds admissible)**.** *Consider event F^c and an episode k, time step $h \in [H]$ and $s, a \in \mathcal{S} \times \mathcal{A}$. Assume that $\tilde{V}_{k,h+1} \geq V_{h+1}^\star \geq V_{h+1}^{\pi_k} \geq \underset{\sim}{V}_{k,h+1}$ and that the upper confidence bound width is at least*

$$\tilde{\psi}_{k,h}(s,a) \geq \alpha\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + \beta\phi(n_k(s,a))^2 + \gamma\phi(n_k(s,a))$$

where there are three possible choices for α, β and γ:

$$\alpha = 0 \qquad \beta = 0 \qquad \gamma = 1 + V_{h+1}^{\max} \qquad \text{or} \tag{5.16}$$

$$\alpha = 0 \qquad \beta = 8.13V_{h+1}^{\max} \qquad \gamma = 1 + 3.47\sqrt{\sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}) + \hat{P}_k(s,a)(\rho^2)} \tag{5.17}$$

$$\alpha = \frac{1}{C} \qquad \beta = (8.13V_{h+1}^{\max} + 12C\|\rho\|_1) \qquad \gamma = 1 + 3.47\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}) \tag{5.18}$$

with $\rho = \tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}$ and $C > 0$ arbitrary. Then the upper confidence bound at time h is admissible; that is, $Q_h^\star(s,a) \leq \tilde{Q}_{k,h}(s,a)$.

Proof. We want to show that $\tilde{Q}_{k,h}(s,a) - Q_h^\star(s,a) \geq 0$. Since $Q_h^\star \leq V_h^{\max}$, this quantity is non-negative when the optimistic Q-value is clipped, i.e., $\tilde{Q}_{k,h}(s,a) = V_h^{\max}$. It remains to show that this quantity is non-negative in the non-clipped case in which

$$\begin{aligned}
\tilde{Q}_{k,h}(s,a) - Q_h^\star(s,a) &= \hat{r}_k(s,a) + \tilde{\psi}_{k,h}(s,a) + \hat{P}_k(s,a)\tilde{V}_{k,h+1} - P(s,a)V_{h+1}^\star - r(s,a) \\
&= \hat{r}_k(s,a) - r(s,a) + \hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{h+1}^\star) + (\hat{P}_k(s,a) - P(s,a))V_{h+1}^\star + \tilde{\psi}_{k,h}(s,a)
\end{aligned}$$

by induction hypothesis, we know that $\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{h+1}^\star) \geq 0$, which allows us to bound

$$\begin{aligned}
&\geq \hat{r}_k(s,a) - r(s,a) + (\hat{P}_k(s,a) - P(s,a))V_{h+1}^\star + \tilde{\psi}_{k,h}(s,a) \\
&\geq -|\hat{r}_k(s,a) - r(s,a)| - |(\hat{P}_k(s,a) - P(s,a))V_{h+1}^\star| + \tilde{\psi}_{k,h}(s,a)
\end{aligned}$$

and applying the definition of the failure event F^R to the first term

$$\geq -\phi(n_k(s,a)) - |(\hat{P}_k(s,a) - P(s,a))V_{h+1}^\star| + \tilde{\psi}_{k,h}(s,a). \tag{5.19}$$

It remains to bound the $|(\hat{P}_k(s,a) - P(s,a))V_{h+1}^\star|$ term for which we have two ways. First, we can apply the definition of F^V which allows us to use $|(\hat{P}_k(s,a) - P(s,a))V_{h+1}^\star| \leq V_h^{\max}\phi(n_k(s,a))$. This yields

$$\tilde{Q}_{k,h}(s,a) - Q_h^\star(s,a) \geq \tilde{\psi}_{k,h}(s,a) - \phi(n_k(s,a)) - V_{h+1}^{\max}\phi(n_k(s,a))$$

which is non-negative using the first choice of coefficients for $\tilde{\psi}_{k,h}$ from Equation (5.16). Second, we can apply the definition of F^{VE} which relies on the empirical variance instead of the range of the optimal value of the successor state. This bound gives

$$\begin{aligned}
&|(P(s,a) - \hat{P}_k(s,a))V^\star_{h+1}| \\
\leq&\sqrt{4\hat{P}_k(s,a)[(V^\star_{h+1}(\cdot) - P(s,a)V^\star_{h+1})^2]\phi(n_k(s,a))} + 4.66V^{\max}_{h+1}\phi(n_k(s,a))^2 \qquad (5.20)\\
\leq&\sqrt{12}\sqrt{\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})^2 + \sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\phi(n_k(s,a))} + 8.13V^{\max}_{h+1}\phi(n_k(s,a))^2
\end{aligned}$$

where we applied Lemma 60. Plugging that result into the bound in Equation (5.19) yields

$$\begin{aligned}
\tilde{Q}_{k,h}(s,a) - Q^\star_h(s,a) \geq&\tilde{\psi}_{k,h}(s,a) - \phi(n_k(s,a)) - 8.13V^{\max}_{h+1}\phi(n_k(s,a))^2 \\
&- \sqrt{12}\sqrt{\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})^2 + \sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\phi(n_k(s,a))}.
\end{aligned}$$

This lower bound is non-negative when we use the second coefficient choice from Equation (5.17) for $\tilde{\psi}_{k,h}$. Finally, we can also apply the second inequality from Lemma 60 to Equation (5.20) to get

$$\begin{aligned}
&|(P(s,a) - \hat{P}_k(s,a))V^\star_{h+1}| \\
\leq&\sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\phi(n_k(s,a)) + (8.13V^{\max}_{h+1} + 12C\|\mathbf{1}\{\hat{P}_k(\cdot|s,a) > 0\}(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})\|_2)\phi(n_k(s,a))^2 \\
&+ \frac{1}{C}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})
\end{aligned}$$

and plugging that result into the bound in Equation (5.19) with $\|\mathbf{1}\{\hat{P}_k(\cdot|s,a) > 0\}(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})\|_2 \leq \|\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}\|_1$ shows that the result holds for the third coefficient choice from Equation (5.18) for $\tilde{\psi}_{k,h}$. Hence, we have shown that under either coefficient choice, we have $\tilde{Q}_{k,h}(s,a) - Q^\star_h(s,a) \geq 0$. □

Lemma 59 (Optimality guarantees admissible)**.** *In the good event F^c, for all episodes k, the certificate is valid, that is, $\Delta_k \leq \epsilon_k$. In addition, all Q-value bounds are admissible, i.e., for all $k \in \mathbb{N}$, $h \in [H]$ and $s \in \mathcal{S}, a \in \mathcal{A}$,*

$$\underset{\sim}{Q}_{k,h}(s,a) \leq Q^{\pi_k}_h(s,a) \leq Q^\star_h(s,a) \leq \tilde{Q}_{k,h}(s,a).$$

Proof. Consider the good event F^c. Since we assume that the initial state is deterministic, we have $\Delta_k = V^\star_1(s_{k,1}) - V^{\pi_k}_1(s_{k,1})$. By induction we can show that $\tilde{V}_{k,h}(s) \geq V^\star_h(s) \geq V^{\pi_k}_h(s) \geq \underset{\sim}{V}_{k,h}(s)$ for all k, h, s, a. The induction start is $h = H + 1$ which holds by definition and due to the specific values of $\underset{\sim}{\psi}$ and $\tilde{\psi}$ in the algorithm, we can apply Lemmas 57 and 58 in each induction step. It then follows that in particular $V^{\pi_k}_1(s_{k,1}) \geq \underset{\sim}{V}_1(s_{k,1})$ and the claim follows from

$$\Delta_k = V^\star(s_{k,1}) - V^{\pi_k}(s_{k,1} \leq \tilde{V}_{k,1}(s_{k,1}) - \underset{\sim}{V}_{k,1}(s_{k,1}) = \epsilon_k\,.$$

□

The following two lemmas give us upper bounds on the empirical variance terms. The first lemma is used to show that the algorithm produces admissible bounds while the second is relevant for bounding the number of certificate mistakes.

Lemma 60. *Consider the good event F^c and any episode $k \in \mathbb{N}$ and time step $h \in [H]$. If $\underline{V}_{k,h+1} \leq V^{\pi_k}_{h+1}$ and $V^\star_{h+1} \leq \tilde{V}_{k,h+1}$, then for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$ and $C > 0$*

$$\begin{aligned}
&\sqrt{4\hat{P}_k(s,a)[(V^\star_{h+1}(\cdot) - P(s,a)V^\star_{h+1})^2]}\phi(n_k(s,a)) \\
\leq\ & \sqrt{12}\sqrt{\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1})^2 + \sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})}\phi(n_k(s,a)) + \sqrt{12}V^{\max}_{h+1}\phi(n_k(s,a))^2 \\
\leq\ & \sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\phi(n_k(s,a)) + (\sqrt{12}V^{\max}_{h+1} + 12C\|\mathbf{1}\{\hat{P}_k(\cdot|s,a) > 0\}(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1})\|_2)\phi(n_k(s,a))^2 \\
&+ \tfrac{1}{C}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1}).
\end{aligned}$$

Proof. We first focus on the inner term

$$\begin{aligned}
&\hat{P}_k(s,a)[(V^\star_{h+1}(\cdot) - P(s,a)V^\star_{h+1})^2] \\
=&\hat{P}_k(s,a)[(\tilde{V}_{k,h+1} - \hat{P}_k(s,a)\tilde{V}_{k,h+1} + V^\star_{h+1} - \tilde{V}_{k,h+1} + \hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V^\star_{h+1}) + (\hat{P}_k(s,a) - P(s,a))V^\star_{h+1})^2]
\end{aligned}$$

applying the Cauchy-Schwarz inequality which gives $(a+b+c)^2 \leq 3a^2 + 3b^2 + 3c^2$

$$\begin{aligned}
\leq& 3\hat{P}_k(s,a)[(\tilde{V}_{k,h+1} - \hat{P}_k(s,a)\tilde{V}_{k,h+1})^2] \\
&+ 3\hat{P}_k(s,a)[(V^\star_{h+1} - \tilde{V}_{k,h+1} + \hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V^\star_{h+1}))^2] + 3((\hat{P}_k(s,a) - P(s,a))V^\star_{h+1})^2.
\end{aligned}$$

The term $\hat{P}_k(s,a)[(V^\star_{h+1} - \tilde{V}_{k,h+1} + \hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V^\star_{h+1}))^2]$ is the variance of a r.v. $V^\star_{h+1}(s') - \tilde{V}_{k,h+1}(s')$ when $\tilde{V}_{k,h+1}$ is fixed and s' is drawn from $\hat{P}_k(s,a)$. We can apply the standard identity of variances $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ and rewrite this term as $\hat{P}_k(s,a)(V^\star_{h+1} - \tilde{V}_{k,h+1})^2 - (\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V^\star_{h+1}))^2$. Plugging this back in gives the bound

$$\begin{aligned}
&3\hat{P}_k(s,a)[(\tilde{V}_{k,h+1} - \hat{P}_k(s,a)\tilde{V}_{k,h+1})^2] + 3\hat{P}_k(s,a)(V^\star_{h+1} - \tilde{V}_{k,h+1})^2 \\
&- 3(\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V^\star_{h+1}))^2 + 3((\hat{P}_k(s,a) - P(s,a))V^\star_{h+1})^2
\end{aligned}$$

leveraging $(F^V)^c$ for the final term and dropping the third term which cannot be positive

$$\leq 3\hat{P}_k(s,a)[(\tilde{V}_{k,h+1} - \hat{P}_k(s,a)\tilde{V}_{k,h+1})^2] + 3\hat{P}_k(s,a)(V^\star_{h+1} - \tilde{V}_{k,h+1})^2 + 3(V^{\max}_{h+1})^2\phi(n_k(s,a))^2.$$

We substitute this bound on $\hat{P}_k(s,a)[(V^\star_{h+1}(\cdot) - P(s,a)V^\star_{h+1})^2]$ back into Equation (5.21):

$$\begin{aligned}
&\sqrt{4\hat{P}_k(s,a)[(V^\star_{h+1}(\cdot) - P(s,a)V^\star_{h+1})^2]}\phi(n_k(s,a)) \\
\leq&\sqrt{12\sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}) + 12\hat{P}_k(s,a)(V^\star_{h+1} - \tilde{V}_{k,h+1})^2}\phi(n_k(s,a)) + \sqrt{12}V^{\max}_{h+1}\phi(n_k(s,a))^2.
\end{aligned}$$

We now leverage that $\underline{V}_{k,h+1} \leq V^{\pi_k}_{h+1} \leq V^\star_{h+1} \leq \tilde{V}_{k,h+1}$ to get a computable bound

$$\leq\sqrt{12\sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1}) + 12\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1})^2}\phi(n_k(s,a)) + \sqrt{12}V^{\max}_{h+1}\phi(n_k(s,a))^2.$$

This is the first inequality to show. For the second inequality, we first bound this expression further as

$$\sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\phi(n_k(s,a)) + \sqrt{12}V^{\max}_{h+1}\phi(n_k(s,a))^2 + \sqrt{12\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1})^2}\phi(n_k(s,a)). \tag{5.25}$$

We now treat the last term in Equation (5.25) separately as

$$\sqrt{12\hat{P}_k(s,a)(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1})^2\phi(n_k(s,a))^2}=\sqrt{\sum_{s'\in\mathcal{S}}\frac{12\phi(n_k(s,a))^2}{\hat{P}_k(s'|s,a)}\hat{P}_k(s'|s,a)^2(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1})^2}$$

splitting the sum based on whether $\hat{P}_k(s'|s,a)\leq 12C^2\phi(n_k(s,a))^2$ for $C>0$ and making repeated use of $\sqrt{\sum_i a_i}\leq\sum_i\sqrt{a_i}$

$$\leq\sqrt{\sum_{s'\in\mathcal{S}}\frac{1}{C^2}\hat{P}_k(s'|s,a)^2(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1})^2}+12\sqrt{\sum_{s'\in\mathcal{S}}\mathbf{1}\{\hat{P}_k(s'|s,a)>0\}C^2\phi(n_k(s,a))^4(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1})^2}$$

$$\leq\frac{1}{C}\hat{P}_k(s,a)(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1})+12C\|\mathbf{1}\{\hat{P}_k(\cdot|s,a)>0\}(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1})\|_2\phi(n_k(s,a))^2.$$

Plugging this bound for the final term back in to Equation 5.25 gives the desired statement

$$\begin{aligned}
&\sqrt{4\hat{P}_k(s,a)[(V^\star_{h+1}(\cdot)-P(s,a)V^\star_{h+1})^2]}\phi(n_k(s,a))\\
\leq&\sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\phi(n_k(s,a))\\
&+(\sqrt{12}V^{\max}_{h+1}+12C\|\mathbf{1}\{\hat{P}_k(\cdot|s,a)>0\}(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1})\|_2)\phi(n_k(s,a))^2\\
&+\frac{1}{C}P_k(s,a)(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1}).
\end{aligned}$$

□

Lemma 61. *Consider the good event F^c and any episode $k\in\mathbb{N}$ and time step $h\in[H]$. If $\underline{V}_{k,h+1}\leq V^{\pi_k}_{h+1}$ and $V^\star_{h+1}\leq\tilde{V}_{k,h+1}$, then for any $s\in\mathcal{S}$, $a\in\mathcal{A}$ and $C,D>0$*

$$\begin{aligned}
\sqrt{D\sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})}\phi(n_k(s,a))\leq\quad&\sqrt{4D\sigma^2_{P(s,a)}(V^{\pi_k}_{h+1})}\phi(n_k(s,a))\\
&+(6\sqrt{D}+4CD)V^{\max}_{h+1}S\phi(n_k(s,a))^2\\
&+\tfrac{1}{C}\hat{P}_k(s,a)(\tilde{V}_{k,h+1}-\underline{V}_{k,h+1}).
\end{aligned}$$

Proof. Note that the proof proceeds in the same fashion as Lemma 60. We first focus on the inner term

$$\begin{aligned}
&\hat{P}_k(s,a)[(\tilde{V}_{k,h+1}-\hat{P}_k(s,a)\tilde{V}_{k,h+1})^2]\\
=&\hat{P}_k(s,a)[(V^{\pi_k}_{h+1}-P(s,a)V^{\pi_k}_{h+1}+\tilde{V}_{k,h+1}-V^{\pi_k}_{h+1}-\hat{P}_k(s,a)(\tilde{V}_{k,h+1}-V^{\pi_k}_{h+1})+(P(s,a)-\hat{P}_k(s,a))V^{\pi_k}_{h+1})^2]
\end{aligned}$$

applying $(a+b)^2\leq 2a^2+2b^2$ twice

$$\begin{aligned}
&\leq 2\hat{P}_k(s,a)[(V^{\pi_k}_{h+1}-P(s,a)V^{\pi_k}_{h+1})^2]\\
&\quad+4\hat{P}_k(s,a)[(\tilde{V}_{k,h+1}-V^{\pi_k}_{h+1}-\hat{P}_k(s,a)(\tilde{V}_{k,h+1}-V^{\pi_k}_{h+1}))^2]+4((P(s,a)-\hat{P}_k(s,a))V^{\pi_k}_{h+1})^2
\end{aligned}$$

using the identity of variances applied to the variance of $\tilde{V}_{k,h+1}-V^{\pi_k}_{h+1}$ w.r.t. $\hat{P}_k(s,a)$

$$\begin{aligned}
&=2\hat{P}_k(s,a)[(V^{\pi_k}_{h+1}-P(s,a)V^{\pi_k}_{h+1})^2]+4\hat{P}_k(s,a)(\tilde{V}_{k,h+1}-V^{\pi_k}_{h+1})^2\\
&\quad-4(\hat{P}_k(s,a)(\tilde{V}_{k,h+1}-V^{\pi_k}_{h+1}))^2+4((P(s,a)-\hat{P}_k(s,a))V^{\pi_k}_{h+1})^2
\end{aligned}$$

dropping the third term which cannot be positive and applying the definition of event F^{L1} to the last term

$$\begin{aligned}
&\leq 2\hat{P}_k(s,a)[(V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2] + 4\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{h+1}^{\pi_k})^2 + 16(V_{h+1}^{\max}\sqrt{S}\phi(n_k(s,a)))^2 \\
&= 2P(s,a)[(V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2] + 2(\hat{P}_k(s,a) - P(s,a))[(V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2] \\
&\quad + 4\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{h+1}^{\pi_k})^2 + 16(V_{h+1}^{\max})^2 S\phi(n_k(s,a))^2
\end{aligned}$$

applying Lemma 68 to the second term with $C = 1$ and $f = (V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2$

$$\begin{aligned}
&\leq 4P(s,a)[(V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2] + 12.5\|(V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2\|_1\phi(n_k(s,a))^2 \\
&\quad + 4\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{h+1}^{\pi_k})^2 + 16(V_{h+1}^{\max})^2 S\phi(n_k(s,a))^2 \\
&\leq 4P(s,a)[(V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2] + 4\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{h+1}^{\pi_k})^2 + 28.5(V_{h+1}^{\max})^2 S\phi(n_k(s,a))^2.
\end{aligned}$$

We now plug this result into the right hand expression of Equation 5.26 to get

$$\begin{aligned}
\sqrt{D\hat{P}_k(s,a)[(\tilde{V}_{k,h+1} - \hat{P}_k(s,a)\tilde{V}_{k,h+1})^2]}\phi(n_k(s,a)) \leq &\sqrt{4DP(s,a)[(V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2]}\phi(n_k(s,a)) \\
&+ 6\sqrt{D}V_{h+1}^{\max}\sqrt{S}\phi(n_k(s,a))^2 \\
&+ \sqrt{4D\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{h+1}^{\pi_k})^2}\phi(n_k(s,a)).
\end{aligned}$$

We now treat the last term separately and start by using the assumption that $\underline{V}_{k,h+1} \leq V_{h+1}^{\pi_k} \leq \tilde{V}_{k,h+1}$

$$\begin{aligned}
&\sqrt{4D\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{h+1}^{\pi_k})^2\phi(n_k(s,a))^2} \leq \sqrt{4D\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1})^2\phi(n_k(s,a))^2} \\
&= \sqrt{\sum_{s'\in\mathcal{S}} \frac{4D\phi(n_k(s,a))^2}{\hat{P}_k(s'|s,a)}\hat{P}_k(s'|s,a)^2(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1})^2}
\end{aligned}$$

splitting the sum based on whether $P_k(s'|s,a) \leq 4DC^2\phi(n_k(s,a))^2$ for $C > 0$ and making repeated use of $\sqrt{\sum_i a_i} \leq \sum_i \sqrt{a_i}$

$$\begin{aligned}
&\leq \sqrt{\sum_{s'\in\mathcal{S}} \frac{1}{C^2}\hat{P}_k(s'|s,a)^2(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1})^2} + \sqrt{\sum_{s'\in\mathcal{S}} (4D)^2C^2\phi(n_k(s,a))^4(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1})^2} \\
&\leq \frac{1}{C}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1}) + 4DC\|\tilde{V}_{k,h+1} - \underline{V}_{k,h+1}\|_1\phi(n_k(s,a))^2.
\end{aligned}$$

Plugging this bound for the final term back in gives the desired statement

$$\begin{aligned}
\sqrt{D\hat{P}_k(s,a)[(\tilde{V}_{k,h+1} - \hat{P}_k(s,a)\tilde{V}_{k,h+1})^2]}\phi(n_k(s,a)) \leq &\sqrt{4DP(s,a)[(V_{h+1}^{\pi_k} - P(s,a)V_{h+1}^{\pi_k})^2]}\phi(n_k(s,a)) \\
&+ (6\sqrt{D} + 4CD)V_{h+1}^{\max}S\phi(n_k(s,a))^2 \\
&+ \frac{1}{C}P_k(s,a)(\tilde{V}_{k,h+1} - \underline{V}_{k,h+1}).
\end{aligned}$$

□

5.9.3 Bound on the size of a certificate

We start by deriving an upper bound on each certificate in terms of the confidence bound widths.

Lemma 62 (Upper bound on certificates)**.** *Let $w_{k,h}(s,a) = \mathbb{P}(s_{k,h} = s, a_{k,h} = a | s_{k,1}, \pi_k)$ be the probability of encountering s, a at time h in the kth episode. In the good event F^c, for all episodes k, the following bound on the optimality-guarantee holds*

$$\epsilon_k \leq \exp(6) \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{h=1}^{H} w_{k,h}(s,a)(H \wedge (\beta \phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a))))$$

with $\beta = 336SH^2$ and $\gamma_{k,h}(s,a) = 14\sigma_{P(s,a)}(V_{h+1}^{\pi_k}) + 2$.

Proof. In this lemma, we use the definition of $\underset{\sim}{\psi}_{k,h} = \tilde{\psi}_{k,h} = \frac{1}{H}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + 45SH^2\phi(n_k(s,a))^2 + \left(1 + \sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\right)\phi(n_k(s,a))$ from the algorithm in the main text (Algorithm 4 and note that is an upper bound on the definition of the bonus terms in Algorithm 6. Hence, the lemma holds for both. We start by bounding the sum of confidence widths as

$$\tilde{\psi}_{k,h}(s,a) + \underset{\sim}{\psi}_{k,h}(s,a) \leq \frac{2}{H}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + \hat{\beta}\phi(n_k(s,a))^2 + \hat{\gamma}_{k,h}(s,a)\phi(n_k(s,a))$$

where we define

$$\hat{\beta} = 90SH^2$$
$$\hat{\gamma}_{k,h}(s,a) = 2\left(1 + \sqrt{12}\sigma_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})\right).$$

Before moving on, we further bound the final term using Lemma 61 as

$$\begin{aligned}
\hat{\gamma}_{k,h}(s,a)\phi(n_k(s,a)) =& 2\phi(n_k(s,a)) + \sqrt{48\sigma^2_{\hat{P}_k(s,a)}(\tilde{V}_{k,h+1})}\phi(n_k(s,a)) \\
\leq& (14\sigma_{P(s,a)}(V_{h+1}^{\pi_k}) + 2)\phi(n_k(s,a)) \\
&+ (42 + 192H)V_{h+1}^{\max} S\phi(n_k(s,a))^2 + \frac{1}{H}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}).
\end{aligned}$$

Hence, we can bound the sum of confidence widths as

$$\tilde{\psi}_{k,h}(s,a) + \underset{\sim}{\psi}_{k,h}(s,a) \leq \frac{3}{H}\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + \tilde{\beta}\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a))$$

where we define

$$\tilde{\beta} = (42 + 192 + 90)SH^2 = 324SH^2$$
$$\gamma_{k,h}(s,a) = 14\sigma_{P(s,a)}(V_{h+1}^{\pi_k}) + 2.$$

By definition of the upper and lower bound estimates

$$\begin{aligned}
&\tilde{Q}_{k,h}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \\
&\leq \tilde{\psi}_{k,h}(s,a) + \underset{\sim}{\psi}_{k,h}(s,a) + \hat{P}_k(s,a)\tilde{V}_{k,h+1} - \hat{P}_k(s,a)\underset{\sim}{V}_{k,h+1} \\
&\leq \left(1 + \frac{3}{H}\right)\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + \tilde{\beta}\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a)). \qquad (5.29)
\end{aligned}$$

We first treat $\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})$ separately as

$$\hat{P}_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) \leq P(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + (\hat{P}_k(s,a) - P(s,a))(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})$$

and bound the final term with Lemma 68 binding $f = \tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1} \in [0, H]$ and $C = H/3$

$$\leq \left(1 + \frac{3}{H}\right) P(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + 3SH^2\phi(n_k(s,a))^2.$$

Plugging this result back in the expression in (5.29) and setting $\beta = 336SH^2 \geq \tilde{\beta} + 3(1+3/H)SH^2$ yields

$$\begin{aligned}
&\tilde{Q}_{k,h}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \\
&\leq \left(1 + \frac{3}{H}\right)^2 P(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + \beta\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a)) \\
&= \left(1 + \frac{3}{H}\right)^2 P_h^{\pi_k}(s,a)(\tilde{Q}_{k,h+1} - \underset{\sim}{Q}_{k,h+1}) + \beta\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a)).
\end{aligned}$$

Here, $P_h^{\pi_k}(s,a)f = \mathbb{E}[f(s_{k,h+1}, \pi_k(s_{k,h+1}, h+1))|s_{k,h} = s, a_{k,h} = a, \pi_k]$ denotes the composition of $P(s,a)$ and the policy action selection operator at time $h+1$. In addition to the bound above, by construction also $0 \leq \tilde{Q}_{k,h}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \leq V_h^{\max}$ holds at all times. Resolving this recursive bound yields

$$\begin{aligned}
&\epsilon_k = (\tilde{V}_{k,1} - \underset{\sim}{V}_{k,1})(s_{k,1}) = (\tilde{Q}_{k,1} - \underset{\sim}{Q}_{k,1})(s_{k,1}, \pi_k(s_{k,1}, 1)) \\
&\leq \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{h=1}^{H} \left(1 + \frac{3}{H}\right)^{2h} w_{k,h}(s,a)(V_h^{\max} \wedge +(\beta\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a)))) \\
&\leq \exp(6) \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{h=1}^{H} w_{k,h}(s,a)(H \wedge (\beta\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a)))).
\end{aligned}$$

Here we bounded with $x = 2H$

$$\left(1 + \frac{3}{H}\right)^{2h} \leq \left(1 + \frac{3}{H}\right)^{2H} = \left(1 + \frac{6}{x}\right)^{x} \leq \lim_{x\to\infty} \left(1 + \frac{6}{x}\right)^{x} = \exp(6).$$

□

5.9.4 Mistake IPOC bound proof

We now follow the proof structure of Dann, Lattimore, and Brunskill (2017) and define *nice* episodes, in which all state-action pairs either have low probability of occurring or the sum of probability of occurring in the previous episodes is large enough so that outside the failure event we can guarantee that

$$n_k(s,a) \geq \frac{1}{4} \sum_{i<k} w_i(s,a).$$

This allows us then to bound the number of nice episodes with large certificates by the number of times terms of the form

$$\sum_{s,a \in L_k} w_k(s,a) \sqrt{\frac{\text{llnp}(n_k(s,a)) + D}{n_k(s,a)}}$$

can exceed a chosen threshold (see Lemma 69 below).

Definition 63 (Nice Episodes)**.** *An episode k is* nice *if and only if for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ the following condition holds:*

$$w_k(s,a) \leq w_{\min} \quad \vee \quad \frac{1}{4}\sum_{i<k} w_i(s,a) \geq H \ln \frac{SAH}{\delta'}.$$

We denote the set of indices of all nice episodes as $\mathcal{N} \subseteq \mathbb{N}$.

Lemma 64 (Properties of nice episodes)**.** *If an episode k is nice, i.e., $k \in \mathcal{N}$, then in the good event F^c, for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ the following statement holds:*

$$w_k(s,a) \leq w_{\min} \quad \vee \quad n_k(s,a) \geq \frac{1}{4}\sum_{i<k} w_i(s,a).$$

Proof. Since we consider the event F^{Nc}, it holds for all s,a pairs with $w_k(s,a) > w_{\min}$

$$n_k(s,a) \geq \frac{1}{2}\sum_{i<k} w_i(s,a) - H \ln \frac{SAH}{\delta'} \geq \frac{1}{4}\sum_{i<k} w_i(s,a)$$

for $k \in N$. □

Lemma 65 (Number of episodes that are not nice)**.** *On the good event F^c, the number of episodes that are not nice is at most*

$$\frac{4S^2A(A \wedge H)H^2}{c_\epsilon \epsilon} \ln \frac{SAH}{\delta'}.$$

Proof. If an episode k is not nice, then there is s,a with $w_k(s,a) > w_{\min}$ and

$$\sum_{i<k} w_i(s,a) < 4H \ln \frac{SAH}{\delta'}.$$

The sum on the left-hand side of this inequality increases by at least $w_{\min}$ after the episode while the right hand side stays constant, this situation can occur at most

$$\frac{4SAH}{w_{\min}} \ln \frac{SAH}{\delta'} = \frac{4S^2A(A \wedge H)H^2}{c_\epsilon \epsilon} \ln \frac{SAH}{\delta'}$$

times in total. □

5.9.5 Proof of IPOC bound of ORLC, Theorem 53

We are now equipped with all tools to complete the proof of Theorem 53:

Proof of Theorem 53. Consider event F^c which has probability at least $1-\delta$ due to Lemma 56. In this event, all optimality guarantees are admissible by Lemma 59. Further, using Lemma 62, the optimality guarantees are bounded as

$$\epsilon_k \leq \exp(6) \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{h=1}^{H} w_{k,h}(s,a)(H \wedge (\beta\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a))))$$

where $\beta = 336SH^2$ and $\gamma_{k,h}(s,a) = 14\sigma_{P(s,a)}(V_{h+1}^{\pi_k}) + 2$. It remains to show that for any given threshold $\epsilon > 0$ this bound does not exceed ϵ except for at most the number of times prescribed by Equation 53. Recall the definition of L_k as the set of state-action pairs with significant probability of occurring, $L_k = \{(s,a) \in \mathcal{S} \times \mathcal{A} : w_k(s,a) \geq w_{\min}\}$, and split the sum as

$$\begin{aligned}\epsilon_k \leq & \exp(6) \sum_{s,a \notin L_k} w_k(s,a) H \\ & + \exp(6)\beta \sum_{s,a \in L_k} w_k(s,a)\phi(n_k(s,a))^2 + \exp(6) \sum_{s,a \in L_k} \sum_{h=1}^{H} w_{k,h}(s,a)\gamma_{k,h}(s,a)\phi(n_k(s,a))\end{aligned}$$

and bound each of the three remaining terms individually. First, the definition of L_k was chosen such that

$$e^6 \sum_{s,a \notin L_k} w_k(s,a) H \leq e^6 H w_{\min} S(A \wedge H) = \frac{e^6 HS(A \wedge H)\epsilon c_\epsilon}{HS(A \wedge H)} = c_\epsilon e^6 \epsilon,$$

where we used the fact that the number of positive $w_k(s,a)$ is no greater than SA or SH per episode k.

Second, we use Corollary 70 with $r = 1, C = 0.728, D = 0.72 \ln \frac{5.2}{\delta'}$ (which satisfies $D > 1$ for any $\delta' \leq 1$) and $\epsilon' = \frac{c_\epsilon \epsilon}{\beta}$ to bound

$$\exp(6)\beta \sum_{s,a \in L_k} w_k(s,a)\phi(n_k(s,a))^2 \leq c_\epsilon \epsilon e^6$$

on all but at most

$$O\left(\frac{SA\beta}{\epsilon} \operatorname{polylog}(S, A, H, 1/\epsilon, \ln(1/\delta))\right) = O\left(\frac{S^2 AH^2}{\epsilon} \operatorname{polylog}(S, A, H, 1/\epsilon, \ln(1/\delta))\right)$$

nice episodes. Third, we use Lemma 69 with $r = 2, C = 0.728, D = 0.72 \ln \frac{5.2}{\delta'}$ and $\epsilon' = c_\epsilon \epsilon$ to bound

$$\exp(6) \sum_{s,a \in L_k} \sum_{h=1}^{H} w_{k,h}(s,a)\gamma_{k,h}(s,a)\phi(n_k(s,a)) \leq c_\epsilon e^6 \epsilon$$

on all but at most

$$O\left(\frac{SAB}{\epsilon^2} \operatorname{polylog}(S, A, H, 1/\epsilon, \ln(1/\delta))\right) = O\left(\frac{SAH^2}{\epsilon^2} \operatorname{polylog}(S, A, H, 1/\epsilon, \ln(1/\delta))\right)$$

nice episodes. Here we choose $B = 400H^2 + 4H$ which is valid since

$$\begin{aligned}\sum_{s,a \in L_k} \sum_{h=1}^{H} w_{k,h}(s,a)\gamma_{k,h}(s,a)^2 \leq & \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{h=1}^{H} w_{k,h}(s,a)\gamma_{k,h}(s,a)^2 \\ \leq & 2 \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{h=1}^{H} w_{k,h}(s,a) 2^2 + 2 \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{h=1}^{H} w_{k,h}(s,a) 14^2 \sigma^2_{P(s,a)}(V_{h+1}^{\pi_k}) \\ \leq & 8H + 400 \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{h=1}^{H} w_{k,h}(s,a)\sigma^2_{P(s,a)}(V_{h+1}^{\pi_k}) \leq 8H + 400H^2.\end{aligned}$$

The first inequality comes from the definition of $\gamma_{k,h}$ and applying the fact $(a+b)^2 \leq 2a^2 + 2b^2$. The second inequality follows from the fact that $w_{k,h}$ is a probability distribution over state and actions and hence their total sum over all time steps is H. Finally, we applied Lemma 4 by Dann and Brunskill (2015) which tells us that the sum of variances is simply the variance of the sum of rewards per episode and hence bounded by H^2.

Combining the bounds for the three terms above, we obtain that $\epsilon_k \leq 3c_\epsilon \epsilon \leq \epsilon$ on all nice episodes except at most

$$O\left(\left(\frac{SAH^2}{\epsilon^2} + \frac{S^2AH^2}{\epsilon}\right)\text{polylog}(S, A, H, 1/\epsilon, \ln(1/\delta))\right)$$

nice episodes. Further, Lemma 65 states that the number of episodes that are not nice is bounded by

$$O\left(\frac{S^2A(A \wedge H)H^2}{\epsilon}\text{polylog}(S, A, H, 1/\epsilon, \ln(1/\delta))\right).$$

Taking all these bounds together, we can bound $\epsilon_k \leq 4c_\epsilon \epsilon \leq \epsilon$ for all episodes k except at most

$$O\left(\left(\frac{SAH^2}{\epsilon^2} + \frac{S^2A(A \wedge H)H^2}{\epsilon}\right)\text{polylog}(S, A, H, 1/\epsilon, \ln(1/\delta))\right)$$

which completes the proof. □

5.9.6 Tighter cumulative IPOC bound

Theorem 66 (Restatement of Theorem 54)**.** *Algorithm 4 and Algorithm 6 satisfy the following cumulative IPOC bound:*

$$\tilde{O}\left(\sqrt{SAH^2T}\ln\frac{1}{\delta} + S^2AH^2\ln\frac{T}{\delta}\right)$$

Proof. Consider event F^c which has probability as least $1-\delta$ due to Lemma 56. In this event, all certificates are admissible by Lemma 59 and using Lemma 62 the size of the certificate ϵ_k is bounded for all episodes k by

$$\epsilon_k \leq \exp(6)\sum_{s,a\in\mathcal{S}\times\mathcal{A}}\sum_{h=1}^{H} w_{k,h}(s,a)(H \wedge (\beta\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a))))$$

where $\beta = 336SH^2$ and $\gamma_{k,h}(s,a) = 14\sigma_{P(s,a)}(V_{h+1}^{\pi_k}) + 2$. Let us now leverage this bound in the cumulative certificate size after T episodes

$$\sum_{k=1}^{T}\epsilon_k \leq \exp(6)\sum_{k=1}^{T}\sum_{s,a\in\mathcal{S}\times\mathcal{A}}\sum_{h=1}^{H} w_{k,h}(s,a)\left[H \wedge (\beta\phi(n_k(s,a))^2 + \gamma_{k,h}(s,a)\phi(n_k(s,a)))\right].$$

We now split this sum over state-action pairs into three different categories: state-action pairs that have low expected visitation $L_k^c = \{(s,a) \in \mathcal{S}\times\mathcal{A} : w_k(s,a) < w_{\min}\}$, state-action pairs that have substantial visitation probability but have not been observed often enough $U_k = \left\{s,a \in L_k : \sum_{i<k} w_i(s,a) < 4H\ln\frac{SAH}{\delta'}\right\}$ and the remaining state action pairs $W_k = \left\{s,a \in L_k : \sum_{i<k} w_i(s,a) \geq 4H\ln\frac{SAH}{\delta'}\right\}$. Note that these

definitions include $w_{\min}$ which we leave arbitrary for now and will set differently than in the IPOC mistake bound proof. This yields

$$\sum_{k=1}^{T} \epsilon_k \leq \exp(6) \sum_{k=1}^{T} \sum_{s,a \notin L_k} w_k(s,a) H + \exp(6) \sum_{k=1}^{T} \sum_{s,a \in U_k} w_k(s,a) H + \exp(6) \sum_{k=1}^{T} \sum_{s,a \in W_k} \sum_{h=1}^{H} w_{k,h}(s,a) \left[H \wedge (\beta \phi(n_k(s,a))^2 + \gamma_{k,h}(s,a) \phi(n_k(s,a))) \right] \quad (5.30)$$

We now bound each term individually, starting with the first,

$$\sum_{k=1}^{T} \sum_{s,a \notin L_k} w_k(s,a) H \leq SAHT w_{\min}.$$

The second term is bounded as

$$\sum_{k=1}^{T} \sum_{s,a \in U_k} w_k(s,a) H = H \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{k=1}^{T} w_k(s,a) \mathbf{1} \left\{ \sum_{i<k} w_i(s,a) < 4H \ln \frac{SAH}{\delta'} \right\}$$

and now let for each s, a be $y_{s,a} \in \mathbb{N}$ the largest index so that $\sum_{k=1}^{y_{s,a}} w_k(s,a) < 4H \ln \frac{SAH}{\delta'}$ and write the sum as

$$\leq H \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \left[4 \sum_{k=1}^{y_{s,a}} w_k(s,a) + w_{y_{s,a}+1}(s,a) \right] \leq SAH \left(4H \ln \frac{SAH}{\delta'} + H \right).$$

Let us now move on to the final term in (5.30). We first look at the sub-term of the form

$$\sum_{k=1}^{T} \sum_{s,a \in W_k} w_k(s,a) \phi(n_k(s,a))^2 = \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{k=1}^{T} \mathbf{1}\{s,a \in W_k\} w_k(s,a) \phi(n_k(s,a))^2 \quad (5.31)$$

We upper-bound $\phi(n_k(s,a))^2$ by a slightly simpler expression $\frac{J}{n_k(s,a)}$ where $J = 0.75 \ln \frac{5.2 \ln(HT)}{\delta'} \geq 0.52 \times 1.4 \ln \frac{5.2 \ln(e \vee n_k(s,a))}{\delta'} \geq 0.52(1.4 \ln \ln(e \vee n_k(s,a)) + \ln(5.2/\delta'))$ which replaces the dependency on the number of observations $n_k(s,a)$ in the log term by the total number of time steps $HT \geq Hk \geq n_k(s,a)$. Applying this bound to (5.31) gives

$$J \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{k=1}^{T} \mathbf{1}\{s,a \in W_k\} \frac{w_k(s,a)}{n_k(s,a)}. \quad (5.32)$$

By the definition of W_k, we know that for all $(s,a) \in W_k$ the following chain of inequalities holds

$$\sum_{i<k} w_i(s,a) \geq 4H \ln \frac{SAH}{\delta'} \geq 8H \geq 8 w_k(s,a).$$

The second inequality is true because of the definition of δ' in Equation (5.3) gives $\frac{SAH}{\delta'} = \frac{SAH(5SAH + 4SA + 4S^2A)}{\delta}$ which is lower bounded by $13 \geq \exp(2)$ because $\delta \leq 1$ and $S, A, H \geq 1$. Leveraging this chain of inequalities in combination with $(F^N)^C$, we can obtain similar to the property of nice episodes a lower bound on $n_k(s,a)$ for $(s,a) \in W_k$ as

$$n_k(s,a) \geq \frac{1}{2} \sum_{i<k} w_i(s,a) - H \ln \frac{SAH}{\delta'} \geq \frac{1}{4} \sum_{i<k} w_i(s,a) \geq \frac{2}{9} \sum_{i \leq k} w_i(s,a).$$

Plugging this back into (5.32) and applying Lemma 36 gives

$$\frac{9J}{2}\sum_{s,a\in\mathcal{S}\times\mathcal{A}}\sum_{k=1}^{T}\mathbf{1}\{s,a\in W_k\}\frac{w_k(s,a)}{\sum_{i\le k}w_i(s,a)}\le\frac{9JSA}{2}\ln\frac{HTe}{w_{\min}}.$$

We have just derived that $\sum_{k=1}^{T}\sum_{s,a\in W_k}w_k(s,a)\phi(n_k(s,a))^2\le\frac{9JSA}{2}\ln\frac{HTe}{w_{\min}}$ and we now move on to the dominant term of the form

$$\begin{aligned}&\sum_{k=1}^{T}\sum_{s,a\in W_k}\sum_{h=1}^{H}w_{k,h}(s,a)\gamma_{k,h}(s,a)\phi(n_k(s,a))\\&\le\sqrt{\sum_{k=1}^{T}\sum_{s,a\in W_k}\sum_{h=1}^{H}w_{k,h}(s,a)\gamma_{k,h}(s,a)^2}\sqrt{\sum_{k=1}^{T}\sum_{s,a\in W_k}\sum_{h=1}^{H}w_{k,h}(s,a)\phi(n_k(s,a))^2}\end{aligned}$$

where the inequality follows from Cauchy-Schwarz. In the proof of Theorem 53, we have derived that the expression under the first $\sqrt{}$ can be upper-bounded by $8T(H+50H^2)$ and we just derived that the expression under the second $\sqrt{}$ is upper bounded by $\frac{9JSA}{2}\ln\frac{HTe}{w_{\min}}$. Putting all these pieces together gives

$$\sum_{k=1}^{T}\sum_{s,a\in W_k}\sum_{h=1}^{H}w_{k,h}(s,a)\gamma_{k,h}(s,a)\phi(n_k(s,a))\le\sqrt{1836JSAH^2T\ln\frac{HTe}{w_{\min}}}.$$

Plugging all the bounds together in (5.30) gives

$$\begin{aligned}\sum_{k=1}^{T}\epsilon_k&\le SAHTw_{\min}+SAH\left(4H\ln\frac{SAH}{\delta'}+H\right)\\&\quad+\exp(6)\beta SA\frac{9J}{2}\ln\frac{HTe}{w_{\min}}+43\exp(6)\sqrt{JSAH^2T\ln\frac{HTe}{w_{\min}}}.\end{aligned}$$

Setting $w_{\min}=\frac{1}{\sqrt{SAT}}$ gives

$$\begin{aligned}\sum_{k=1}^{T}\epsilon_k&=O\left(\sqrt{SAH^2T}\right)+\tilde{O}(SAH^2\ln 1/\delta)\\&\quad+\tilde{O}(S^2AH^2\ln T/\delta)+\tilde{O}(\sqrt{SAH^2T}\ln 1/\delta)\\&=\tilde{O}\left(\sqrt{SAH^2T}\ln\frac{1}{\delta}+S^2AH^2\ln\frac{T}{\delta}\right)\end{aligned}$$

□

5.9.7 Technical Lemmas

Lemma 67. *Let $\tau\in(0,\hat{\tau}]$ and $D\ge 1$. Then for all $x\ge\check{x}=\frac{\ln(C/\tau)+D}{\tau}$ with $C=16\vee\hat{\tau}D^2$, the following inequality holds*

$$\frac{\mathrm{llnp}(x)+D}{x}\le\tau.$$

Proof. Since by Lemma 37 the function $\frac{\mathrm{llnp}(x)+D}{x}$ is monotonically decreasing in x, we can bound

$$\frac{\mathrm{llnp}(x)+D}{x} \leq \frac{\mathrm{llnp}(\check{x})+D}{\check{x}} = \frac{\mathrm{llnp}(\check{x})+D}{\ln(C/\tau)+D}\tau.$$

It remains to show that $\ln(\check{x}) \vee 1 \leq \frac{C}{\tau}$. First, note that $\frac{C}{\tau} \geq \frac{C}{\check{\tau}} \geq D^2 \geq 1$. Also, we can bound using $\ln(x) \leq 2\sqrt{x}$

$$\begin{aligned}\ln(\check{x}) =& \ln\left(\frac{\ln(C/\tau)+D}{\tau}\right) \leq 2\sqrt{\frac{\ln(C/\tau)+D}{\tau}} \leq 2\sqrt{\frac{2\sqrt{C/\tau}+D}{\tau}} \\ \leq& 4\left(\frac{C}{\tau^3}\right)^{1/4} \leq \left(\frac{C}{\tau}\right)^{3/4} \leq \frac{C}{\tau},\end{aligned}$$

since $\sqrt{C} \geq 4$ and $C/\tau \geq 1$. □

Lemma 68. *Let $f : \mathcal{S} \mapsto [0,\infty]$ be a (potentially random) function. In the good event F^c, for all episodes k, states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$, the following bound holds for any $C > 0$*

$$\begin{aligned}|(\hat{P}_k - P)(s,a)f| \leq& 1.56\|f\|_1\phi(n_k(s,a))^2 + 2\phi(n_k(s,a))\sum_{s\in\mathcal{S}}\sqrt{P(s'|s,a)}f(s') \\ \leq& (4C+1.56)\|f\|_1\phi(n_k(s,a))^2 + \frac{1}{C}P(s,a)f\end{aligned}$$

and

$$\begin{aligned}|(\hat{P}_k - P)(s,a)f| \leq& 4.66\|f\|_1\phi(n_k(s,a))^2 + 2\phi(n_k(s,a))\sum_{s\in\mathcal{S}}\sqrt{\hat{P}_k(s'|s,a)}f(s') \\ \leq& (4C+4.66)\|f\|_1\phi(n_k(s,a))^2 + \frac{1}{C}\hat{P}_k(s,a)f.\end{aligned}$$

Proof.

$$|(\hat{P}_k - P)(s,a)f| \leq \sum_{s'\in\mathcal{S}}|(\hat{P}_k - P)(s'|s,a)|f(s')$$

We now apply the definition of F^c on each $|(\tilde{P}_k - P)(s'|s,a)|$ individually. Specifically, we use F^P and F^{PE} for the first and second bound respectively. To unify their treatment, we use $\tilde{P}$ for P and $\hat{P}_k$, $c_1 = 4$ and c_2 for 1.56 and 4.66 respectively.

$$\begin{aligned}\leq& \sum_{s'\in\mathcal{S}} f(s')c_2\phi(n_k(s,a)^2 + \sqrt{c_1\tilde{P}(s'|s,a)}\phi(n_k(s,a))f(s')) \\ =& c_2\|f\|_1\phi(n_k(s,a))^2 + \phi(n_k(s,a))\sum_{s'\in\mathcal{S}}\sqrt{c_1\tilde{P}(s'|s,a)}f(s')\end{aligned}$$

This is the first inequality to show but we can further rewrite this to show the second inequality as follows

$$= c_2\|f\|_1\phi(n_k(s,a))^2 + \phi(n_k(s,a))\sum_{s'\in\mathcal{S}}\sqrt{\frac{c_1}{\tilde{P}(s'|s,a)}}\tilde{P}(s'|s,a)f(s')\,.$$

Splitting the last sum based on whether $\sqrt{\tilde{P}(s'|s,a)}$ is smaller or larger than $\sqrt{c_1}C\phi(n_k(s,a))$

$$
\begin{aligned}
&\leq c_2\|f\|_1\phi(n_k(s,a))^2 + \frac{1}{C}\tilde{P}(s,a)f \\
&\quad + \phi(n_k(s,a))\sum_{s'\in\mathcal{S}}\sqrt{\frac{c_1}{\tilde{P}(s'|s,a)}}\tilde{P}(s'|s,a)f(s')\mathbf{1}\{\sqrt{\tilde{P}(s'|s,a)} < \sqrt{c_1}C\phi(n_k(s,a))\} \\
&\leq c_2\|f\|_1\phi(n_k(s,a))^2 + \frac{1}{C}\tilde{P}(s,a)f + \phi(n_k(s,a))^2\sum_{s'\in\mathcal{S}}c_1Cf(s') \\
&\leq c_2\|f\|_1\phi(n_k(s,a))^2 + \frac{1}{C}\tilde{P}(s,a)f + \phi(n_k(s,a))^2c_1C\|f\|_1 \\
&\leq (c_1C + c_2)\|f\|_1\phi(n_k(s,a))^2 + \frac{1}{C}\tilde{P}(s,a)f\,.
\end{aligned}
$$

□

Lemma 69 (Rate Lemma, Adaption of Lemma E.3 by Dann, Lattimore, and Brunskill (2017))**.** *Fix $r \geq 1$, $\epsilon' > 0$, $C > 0$ and $D \geq 1$, where C and D may depend polynomially on relevant quantities. Let $w_{k,h}(s,a) = \mathbb{P}(s_{k,h} = s, a_{k,h} = a|s_{k,1}, \pi_k)$ be the probability of encountering s, a at time h in the kth episode. Then for any functions $\gamma_{k,h} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}_+$ indexed by $h \in [H]$*

$$
\sum_{s,a\in L_k}\sum_{h=1}^{H} w_{k,h}(s,a)\gamma_{k,h}(s,a)\left(\frac{C(\mathrm{llnp}(2n_k(s,a)) + D)}{n_k(s,a)}\right)^{1/r} \leq \epsilon'
$$

on all but at most

$$
\frac{6CASB^{r-1}}{\epsilon'^r}\,\mathrm{polylog}(S, A, H, \delta^{-1}, \epsilon'^{-1}), \quad \textit{where} \quad B \geq \sum_{s,a\in L_k}\sum_{h=1}^{H} w_{k,h}(s,a)\gamma_{k,h}(s,a)^{r/(r-1)}
$$

nice episodes.

Proof. Define

$$
\begin{aligned}
\Delta_k &= \sum_{s,a\in L_k}\sum_{h=1}^{H} w_{k,h}(s,a)\gamma_{k,h}(s,a)\left(\frac{C(\mathrm{llnp}(2n_k(s,a)) + D)}{n_k(s,a)}\right)^{1/r} \\
&= \sum_{s,a\in L_k}\sum_{h=1}^{H}(w_k(s,a)\gamma_{k,h}(s,a)^{r/(r-1)})^{1-\frac{1}{r}}\left(w_{k,h}(s,a)\frac{C(\mathrm{llnp}(2n_k(s,a)) + D)}{n_k(s,a)}\right)^{1/r}.
\end{aligned}
$$

We first bound using Hölder's inequality

$$
\begin{aligned}
\Delta_k &\leq \left(\sum_{s,a\in L_k}\sum_{h=1}^{H} w_{k,h}(s,a)\gamma_{k,h}(s,a)^{r/(r-1)}\right)^{1-\frac{1}{r}}\left(\sum_{s,a\in L_k}\sum_{h=1}^{H}\frac{Cw_{k,h}(s,a)(\mathrm{llnp}(2n_k(s,a)) + D)}{n_k(s,a)}\right)^{\frac{1}{r}} \\
&\leq \left(\sum_{s,a\in L_k}\frac{CB^{r-1}w_k(s,a)(\mathrm{llnp}(2n_k(s,a)) + D)}{n_k(s,a)}\right)^{\frac{1}{r}}.
\end{aligned}
$$

Using the property in Lemma 64 of nice episodes as well as the fact that $w_k(s,a) \leq H$ and $\sum_{i<k} w_i(s,a) \geq 4H \ln \frac{SAH}{\delta'} \geq 4H \ln(2) \geq 2H$, we bound

$$n_k(s,a) \geq \frac{1}{4} \sum_{i<k} w_i(s,a) \geq \frac{1}{6} \sum_{i \leq k} w_i(s,a).$$

The function $\frac{\mathrm{llnp}(2x)+D}{x}$ is monotonically decreasing in $x \geq 0$ since $D \geq 1$ (see Lemma 37). This allows us to bound

$$\begin{aligned}
\Delta_k^r \leq & \sum_{s,a \in L_k} \frac{CB^{r-1} w_k(s,a)(\mathrm{llnp}(2n_k(s,a)) + D)}{n_k(s,a)} \\
\leq & 6CB^{r-1} \sum_{s,a \in L_k} \frac{w_k(s,a) \left(\mathrm{llnp}\left(\frac{1}{3}\sum_{i \leq k} w_i(s,a)\right) + D\right)}{\sum_{i \leq k} w_i(s,a)} \\
\leq & 6CB^{r-1} \sum_{s,a \in L_k} \frac{w_k(s,a) \left(\mathrm{llnp}\left(\sum_{i \leq k} w_i(s,a)\right) + D\right)}{\sum_{i \leq k} w_i(s,a)} \\
\leq & 6CB^{r-1} \left[0 \vee \max_{s,a \in L_k} \frac{\mathrm{llnp}\left(\sum_{i \leq k} w_i(s,a)\right) + D}{\sum_{i \leq k} w_i(s,a)}\right] \sum_{s,a \in L_k} w_k(s,a)
\end{aligned}$$

which can be further bounded by leveraging that the sum of all weights w_k sum to H for each episode k

$$\leq 6CHB^{r-1} \left[0 \vee \max_{s,a \in L_k} \frac{\mathrm{llnp}\left(\sum_{i \leq k} w_i(s,a)\right) + D}{\sum_{i \leq k} w_i(s,a)}\right].$$

Assume now $\Delta_k > \epsilon'$. In this case, the right-hand side of the inequality above is also larger than ϵ'^r and there is at least one (s,a) for which $w_k(s,a) > w_{\min}$ and

$$\begin{aligned}
\frac{6CHB^{r-1} \left(\mathrm{llnp}\left(\sum_{i \leq k} w_i(s,a)\right) + D\right)}{\sum_{i \leq k} w_i(s,a)} > & \epsilon'^r \\
\Leftrightarrow \frac{\mathrm{llnp}\left(\sum_{i \leq k} w_i(s,a)\right) + D}{\sum_{i \leq k} w_i(s,a)} > & \frac{\epsilon'^r}{6CHB^{r-1}}.
\end{aligned}$$

Let us denote $C' = \frac{6CHB^{r-1}}{\epsilon'^r}$. Since $\frac{\mathrm{llnp}(x)+D}{x}$ is monotonically decreasing and $x = C'^2 + 3C'D$ satisfies $\frac{\mathrm{llnp}(x)+D}{x} \leq \frac{\sqrt{x}+D}{x} \leq \frac{1}{C'}$, we know that if $\sum_{i \leq k} w_i(s,a) \geq C'^2 + 3C'D$ then the above condition cannot be satisfied for s,a. Since each time the condition is satisfied, it holds that $w_k(s,a) > w_{\min}$ and so $\sum_{i \leq k} w_i(s,a)$ increases by at least $w_{\min}$, it can happen at most

$$m \leq \frac{SA(C'^2 + 3C'D)}{w_{\min}}$$

times that $\Delta_k > \epsilon'$. Define $K = \{k : \Delta_k > \epsilon'\} \cap N$ and we know that $|K| \leq m$. Now we consider the

sum

$$\sum_{k\in K}\Delta_k^r \leq \sum_{k\in K} 6CB^{r-1}\sum_{s,a\in L_k}\frac{w_k(s,a)\left(\operatorname{llnp}\left(\sum_{i\leq k}w_i(s,a)\right)+D\right)}{\sum_{i\leq k}w_i(s,a)}$$
$$\leq 6CB^{r-1}\left(\operatorname{llnp}\left(C'^2+3C'D\right)+D\right)\sum_{s,a\in L_k}\sum_{k\in K}\frac{w_k(s,a)}{\sum_{i\leq k}w_i(s,a)\mathbf{1}\{w_i(s,a)\geq w_{\min}\}}.$$

For every (s,a), we consider the sequence of $w_i(s,a)\in[w_{\min},H]$ with $i\in I=\{i\in\mathbb{N}:w_i(s,a)\geq w_{\min}\}$ and apply Lemma 36. This yields that

$$\sum_{k\in K}\frac{w_k(s,a)}{\sum_{i\leq k}w_i(s,a)\mathbf{1}\{w_i(s,a)\geq w_{\min}\}}\leq 1+\ln(mH/w_{\min})=\ln\left(\frac{Hme}{w_{\min}}\right)$$

and hence

$$\sum_{k\in K}\Delta_k^r \leq 6CASB^{r-1}\ln\left(\frac{Hme}{w_{\min}}\right)\left(\operatorname{llnp}\left(C'^2+3C'D\right)+D\right).$$

Since each element in K has to contribute at least ϵ'^r to this bound, we can conclude that

$$\sum_{k\in N}\mathbf{1}\{\Delta_k\geq\epsilon'\}\leq\sum_{k\in K}\mathbf{1}\{\Delta_k\geq\epsilon'\}\leq|K|\leq\frac{6CASB^{r-1}}{\epsilon'^r}\ln\left(\frac{Hme}{w_{\min}}\right)\left(\operatorname{llnp}\left(C'^2+3C'D\right)+D\right).$$

Since $\ln\left(\frac{Hme}{w_{\min}}\right)\left(\operatorname{llnp}\left(C'^2+3C'D\right)+D\right)$ is $\operatorname{polylog}(S,A,H,\delta^{-1},\epsilon'^{-1})$, the proof is complete. □

Corollary 70. *Fix $r\geq 1$, $\epsilon'>0$, $C>0$, and $D\geq 1$, where C and D may depend polynomially on relevant quantities. Then,*

$$\sum_{s,a\in L_k}w_k(s,a)\left(\frac{C(\operatorname{llnp}(2n_k(s,a))+D)}{n_k(s,a)}\right)^{1/r}\leq\epsilon'$$

on all but at most

$$\frac{6CASH^{r-1}}{\epsilon'^r}\operatorname{polylog}(S,A,H,\delta^{-1},\epsilon'^{-1}).$$

nice episodes.

Proof. This corollary follows directly from Lemma 69 with $\gamma_h(s,a)=1$ and noting that

$$H\geq\sum_{s,a\in\mathcal{S}\times\mathcal{A}}\sum_{h=1}^{H}w_{k,h}(s,a)\geq\sum_{s,a\in L_k}\sum_{h=1}^{H}w_{k,h}(s,a)\gamma_{k,h}(s,a)^{r/(r-1)}.$$

□

Corollary 71. *Using the terminology by Howard et al. (2018), for any $c>0$, $\delta\in(0,1)$, the following function is a sub-gamma boundary (and as such also a sub-exponential boundary) with scale parameter c and crossing probability δ:*

$$u_{c,\delta}(v)=1.44\sqrt{v\left(1.4\operatorname{llnp}(2v)+\log\frac{5.2}{\delta}\right)}+2.42c\left(1.4\operatorname{llnp}(2v)+\log\frac{5.2}{\delta}\right).$$

Further $u_{0,\delta}$ is a sub-Gaussian boundary with crossing probability δ and $u_{c/3,\delta}$ is a sub-Poisson boundary with crossing probability δ for scale parameter c.

Proof. This result follows directly from Theorem 1 by Howard et al. (2018) instantiated with $h(k) = (k+1)^s\zeta(s)$, $s = 1.4$ and $\eta = 2$. The final statements follows from the fact that ψ_N is a special case of ψ_G with $c = 0$ and Proposition 5 in Howard et al. (2018). □

5.10 Theoretical analysis of Algorithm 5 for finite episodic MDPs with side information

5.10.1 Failure event and bounding the failure probability

We define the following failure event

$$F = F^{(r)} \cup F^{(p)} \cup F^{O}$$

where

$$F^{(r)} = \left\{ \exists s, a \in \mathcal{S} \times \mathcal{A}, k \in \mathbb{N} \,:\, \|\hat{\theta}^{(r)}_{k,s,a} - \theta^{(r)}_{s,a}\|_{N^{(r)}_{k,s,a}} \geq \sqrt{\lambda}\|\theta^{(r)}_{s,a}\|_2 + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}\log\frac{\det N^{(r)}_{k,s,a}}{\det \lambda I}} \right\},$$

$$\begin{aligned} F^{(p)} =& \Bigg\{ \exists s', s, a \in \mathcal{S} \times \mathcal{S} \times \mathcal{A}, k \in \mathbb{N} \,:\, \|\hat{\theta}^{(p)}_{k,s',s,a} - \theta^{(p)}_{s',s,a}\|_{N^{(p)}_{k,s,a}} \\ &\geq \sqrt{\lambda}\|\theta^{(p)}_{s',s,a}\|_2 + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}\log\frac{\det N^{(p)}_{k,s,a}}{\det \lambda I}} \Bigg\}, \\ F^{O} =& \Bigg\{ \exists T \in \mathbb{N} \,:\, \sum_{k=1}^{T} \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{h=1}^{H} [\mathbb{P}(s_{k,h} = s, a_{k,h} = a | s_{k,1}, \pi_k) - \mathbf{1}\{s = s_{k,h}, a = a_{k,h}\}] \\ &\geq SH\sqrt{T\log\frac{6\log(2T)}{\delta'}} \Bigg\}, \\ \delta' =& \frac{\delta}{SA + S^2A + SH}. \end{aligned}$$

Lemma 72. *The failure probability $\mathbb{P}(F)$ is bounded by δ.*

Proof. Consider an arbitrary $s \in \mathcal{S}$, $a \in \mathcal{A}$ and define $\mathcal{F}_t$ where $t = Hk + h$ with $h \in [H]$ is the running time step index as follows: $\mathcal{F}_t$ is the sigma-field induced by all observations up to $s_{k,h}$ and $a_{k,h}$ including x_k but not $r_{k,h}$ and not $s_{k,h+1}$. Then $\eta_t = 2\mathbf{1}\{s_{k,h} = s, a_{k,h} = a\}((x^{(r)}_k)^\top\theta^{(r)}_{s,a} - r_{k,h})$ is a martingale difference sequence adapted to $\mathcal{F}_t$. Moreover, since η_t takes values in $[2(x^{(r)}_k)^\top\theta^{(r)}_{s,a} - 2, 2(x^{(r)}_k)^\top\theta^{(r)}_{s,a}]$ almost surely it is conditionally sub-Gaussian with parameter 1. We can then apply Theorem 20.2 in Lattimore and Czepesvari (2018) to get

$$2\|\hat{\theta}^{(r)}_{k,s,a} - \theta^{(r)}_{s,a}\|_{N^{(r)}_{k,s,a}} \leq \sqrt{\lambda}2\|\theta^{(r)}_{s,a}\|_2 + \sqrt{2\log\frac{1}{\delta'} + \log\frac{\det N^{(r)}_{k,s,a}}{\det \lambda I}}$$

for all $k \in \mathbb{N}$ with probability at least $1-\delta'$. Similarly for any fixed $s' \in \mathcal{S}$, using $\eta_t = 2\mathbf{1}\{s_{k,h} = s, a_{k,h} = a\}((x_k^{(p)})^\top \theta^{(p)}_{s',s,a} - \mathbf{1}\{s_{k,h+1} = s'\})$, it holds with probability at least $1-\delta'$ that

$$\|\hat{\theta}^{(p)}_{k,s',s,a} - \theta^{(p)}_{s',s,a}\|_{N^{(p)}_{k,s,a}} \leq \sqrt{\lambda}\|\theta^{(p)}_{s',s,a}\|_2 + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}\log\frac{\det N^{(p)}_{k,s,a}}{\det \lambda I}}$$

for all episodes k. Finally, for a fixed $s \in \mathcal{S}$ and $h \in [H]$ the sequence

$$\eta_k = \sum_{a\in\mathcal{A}}[\mathbb{P}(s_{k,h} = s, a_{k,h} = a | s_{k,1}, \pi_k) - \mathbf{1}\{s = s_{k,h}, a = a_{k,h}\}]$$

is a martingale difference sequence with respect to $\mathcal{G}_k$, defined as the sigma-field induced by all observations up to including episode $k-1$ and x_k and $s_{k,1}$. All but at most one action has zero probability of occurring (π_k is deterministic) and therefore $\eta_k \in [c, c+1]$ with probability 1 for some c that is measurable in $\mathcal{G}_k$. Hence, $S_t = \sum_{k=1}^t \eta_k$ satisfies Assumption 1 with $V_t = t/4$ and ψ_N and $\mathbb{E}L_0 = 1$ (Hoeffding I case in Table 2 of the appendix). This allows us to apply Theorem 1 by Howard et al. (2018) where we choose $h(k) = (1+k)^s\zeta(s)$ with $s = 1.4$ and $\eta = 2$, which gives us (see Eq. (8) and Eq. (9)) specifically) that with probability at least $1-\delta'$ for all $T \in \mathbb{N}$

$$\sum_{k=1}^{T}\sum_{a\in\mathcal{A}}[\mathbb{P}(s_{k,h} = s, a_{k,h} = a | s_{k,1}, \pi_k) - \mathbf{1}\{s = s_{k,h}, a = a_{k,h}\}] = \sum_{k=1}^{T}\eta_k \leq \sqrt{T(\log\log(T/2) + \log(6/\delta'))}.$$

Setting $\delta' = \frac{\delta}{SA+S^2A+SH}$, all statements above hold for all s', s, a, h jointly using a union bound with probability at least $1-\delta$. This implies that $\mathbb{P}(F) \leq \delta$. □

Using the bounds on the linear parameter estimates, the following lemma derives bounds on the empirical model.

Lemma 73 (Bounds on model parameters)**.** *Outside the failure event F, assuming $\|\theta^{(p)}_{s',s,a}\|_2 \leq \xi_{\theta^{(p)}}$ and $\|\theta^{(r)}_{s,a}\|_2 \leq \xi_{\theta^{(r)}}$ for all $s', s \in \mathcal{S}$ and $a \in \mathcal{A}$ we have*

$$|\hat{r}_k(s,a) - r_k(s,a)| \leq 1 \wedge \alpha_{k,s,a}\|x_k^{(r)}\|_{(N^{(r)}_{k,s,a})^{-1}}$$
$$|\hat{P}_k(s'|s,a) - P_k(s,a)\|_1 \leq 1 \wedge \gamma_{k,s,a}\|x_k^{(p)}\|_{(N^{(p)}_{k,s,a})^{-1}}$$

where

$$\alpha_{k,s,a} = \sqrt{\lambda}\xi_{\theta^{(r)}} + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}\log\frac{\det N^{(r)}_{k,s,a}}{\det(\lambda I)}}$$
$$\gamma_{k,s,a} = \sqrt{\lambda}\xi_{\theta^{(p)}} + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}\log\frac{\det N^{(p)}_{k,s,a}}{\det(\lambda I)}}\,.$$

Proof. Since $\hat{r}_k \in [0,1]$ and $r_k \in [0,1]$, we have

$$|\hat{r}_k(s,a) - r_k(s,a)| \leq 1 \wedge |(x_k^{(r)})^\top \hat{\theta}^{(r)}_{k,s,a} - r_k(s,a)|.$$

The last term can be bounded as

$$
\begin{aligned}
&|(x_k^{(r)})^\top \hat{\theta}_{k,s,a}^{(r)} - r_k(s,a)| = |(x_k^{(r)})^\top (\hat{\theta}_{k,s,a}^{(r)} - \theta_{s,a}^{(r)})| \leq \|x_k^{(r)}\|_{(N_{k,s,a}^{(r)})^{-1}} \|\hat{\theta}_{k,s,a}^{(r)} - \theta_{s,a}^{(r)}\|_{N_{k,s,a}^{(r)}} \\
\leq &\|x_k^{(r)}\|_{(N_{k,s,a}^{(r)})^{-1}} \left[\sqrt{\lambda} \|\theta_{s,a}^{(r)}\|_2 + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}\log\frac{\det(N_{k,s,a}^{(r)})}{\det(\lambda I)}} \right] \\
\leq &\alpha_{k,s,a} \|x_k^{(r)}\|_{(N_{k,s,a}^{(r)})^{-1}}
\end{aligned}
$$

where we first used Hölder's inequality, then the definition of $F^{(r)}$, and finally the assumption $\|\theta_{s,a}^{(r)}\|_2 \leq \xi_{\theta^{(r)}}$. This proves the first inequality. Consider now the second inequality, which we bound analogously as

$$
\begin{aligned}
&|\hat{P}_k(s'|s,a) - P_k(s'|s,a)| \leq 1 \wedge |(x_k^{(p)})^\top \hat{\theta}_{k,s',s,a}^{(p)} - P_k(s'|s,a)| \\
=&1 \wedge |(x_k^{(p)})^\top (\hat{\theta}_{k,s',s,a}^{(p)} - \theta_{s',s,a}^{(p)})| \leq 1 \wedge \|x_k^{(p)}\|_{(N_{k,s,a}^{(p)})^{-1}} \|\hat{\theta}_{k,s',s,a}^{(p)} - \theta_{s',s,a}^{(p)}\|_{N_{k,s,a}^{(p)}} \\
\leq &1 \wedge \|x_k^{(p)}\|_{(N_{k,s,a}^{(p)})^{-1}} \left[\sqrt{\lambda} \|\theta_{s',s,a}\|_2 + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}\log\frac{\det(N_{k,s,a}^{(p)})}{\det(\lambda I)}} \right] \\
\leq &1 \wedge \gamma_{k,s,a} \|x_k^{(p)}\|_{(N_{k,s,a}^{(p)})^{-1}} .
\end{aligned}
$$

□

5.10.2 Admissibility of guarantees

Lemma 74 (Upper bound admissible)**.** *Outside the failure event F, for all episodes k, $h \in [H]$ and $s, a \in \mathcal{S} \times \mathcal{A}$*

$$Q_{k,h}^\star(s,a) \leq \tilde{Q}_{k,h}(s,a).$$

Proof. Consider a fixed episode k. For $h = H+1$ the claim holds by definition. Assume the claim holds for $h+1$ and consider $\tilde{Q}_{k,h}(s,a) - Q_{k,h}^\star(s,a)$. Since $Q_{k,h}^\star \leq V_h^{\max}$, this quantity is non-negative when $\tilde{Q}_{k,h}(s,a) = V_h^{\max}$. In the other case

$$
\begin{aligned}
&\tilde{Q}_{k,h}(s,a) - Q_{k,h}^\star(s,a) \\
\geq &\hat{r}_k(s,a) + \hat{P}_k(s,a)\tilde{V}_{k,h+1} + \psi_{k,h}(s,a) - P_k(s,a)V_{k,h+1}^\star - r_k(s,a) \\
= &\hat{r}_k(s,a) - r_k(s,a) + \psi_{k,h}(s,a) + \hat{P}_k(s,a)(\tilde{V}_{k,h+1} - V_{k,h+1}^\star) \\
&+ (\hat{P}_k(s,a) - P_k(s,a))V_{k,h+1}^\star
\end{aligned}
$$

by induction hypothesis and $\hat{P}_k(s'|s,a) \geq 0$

$$
\begin{aligned}
&\geq \hat{r}_k(s,a) - r_k(s,a) + \psi_{k,h}(s,a) + (\hat{P}_k(s,a) - P_k(s,a))V_{k,h+1}^\star \\
&\geq -|\hat{r}_k(s,a) - r_k(s,a)| + \hat{\psi}_{kh}(s,a) - \sum_{s' \in \mathcal{S}} V_{k,h+1}^\star(s')|\hat{P}_k(s'|s,a) - P_k(s'|s,a)|
\end{aligned}
$$

by induction hypothesis

$$
\geq -|\hat{r}_k(s,a) - r_k(s,a)| + \hat{\psi}_{kh}(s,a) - \sum_{s' \in \mathcal{S}} \tilde{V}_{h+1}(s')|\hat{P}_k(s'|s,a) - P_k(s'|s,a)|
$$

using Lemma 73

$$\geq \psi_{k,h}(s,a) - \alpha_{k,s,a}\|x_k^{(r)}\|_{(N_{k,s,a}^{(r)})^{-1}} - \|\tilde{V}_{h+1}\|_1 \gamma_{k,s,a}\|x_k^{(p)}\|_{(N_{k,s,a}^{(p)})^{-1}} = 0\,.$$

□

Using the same technique, we can prove the following result.

Lemma 75 (Lower bound admissible)**.** *Outside the failure event F, for all episodes k, $h \in [H]$ and $s, a \in \mathcal{S} \times \mathcal{A}$*

$$Q_{k,h}^{\pi_k}(s,a) \geq \underset{\sim}{Q}_{k,h}(s,a).$$

Proof. Consider a fixed episode k. For $h = H+1$ the claim holds by definition. Assume the claim holds for $h+1$ and consider $Q_{k,h}^{\pi_k}(s,a) - \underset{\sim}{Q}_{k,h}(s,a)$. Since $Q_{k,h}^{\pi_k} \geq 0$, this quantity is non-negative when $\underset{\sim}{Q}_{k,h}(s,a) = 0$. In the other case

$$\begin{aligned}
&Q_{k,h}^{\pi_k}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \\
=&P_k(s,a)V_{k,h+1}^{\pi_k} + r_k(s,a) - \hat{r}_k(s,a) - \hat{P}_k(s,a)\underset{\sim}{V}_{k,h+1} + \psi_{k,h}(s,a) \\
=&r_k(s,a) - \hat{r}_k(s,a) + \psi_{k,h}(s,a) + P_k(s,a)(V_{k,h+1}^{\pi_k} - \underset{\sim}{V}_{k,h+1}) + (P_k - \hat{P}_k)(s,a)\underset{\sim}{V}_{k,h+1}
\end{aligned}$$

by induction hypothesis and $P_k(s'|s,a) \geq 0$

$$\geq \psi_{k,h}(s,a) - |r_k(s,a) - \hat{r}_k(s,a)| - |(P_k(s,a) - \hat{P}_k(s,a))\underset{\sim}{V}_{k,h+1}|$$

using Lemma 73

$$\geq \psi_{k,h}(s,a) - \alpha_{k,s,a}\|x_k^{(r)}\|_{(N_{k,s,a}^{(r)})^{-1}} - \|\underset{\sim}{V}_{h+1}\|_1 \gamma_{k,s,a}\|x_k^{(p)}\|_{(N_{k,s,a}^{(p)})^{-1}} = 0\,.$$

□

5.10.3 Cumulative certificate bound

Lemma 76. *Outside the failure event F, the cumulative certificates after T episodes for all T are bounded by*

$$\sum_{k=1}^{T} \epsilon_k \leq \tilde{O}\left(\sqrt{S^3AH^2T}V_1^{\max}\lambda(\xi_{\theta^{(p)}}^2 + \xi_{\theta^{(r)}}^2 + d^{(p)} + d^{(r)})\log\frac{\xi_{x^{(p)}}^2 + \xi_{x^{(r)}}^2}{\lambda\delta}\right).$$

Proof. Let $\psi_{k,h}(s,a) = \alpha_{k,s,a}\|x_k^{(r)}\|_{(N_{k,s,a}^{(r)})^{-1}} + V_{h+1}^{\max} S\gamma_{k,s,a}\|x_k^{(p)}\|_{(N_{k,s,a}^{(p)})^{-1}}$ We bound the difference between upper and lower Q-estimate as

$$\begin{aligned}
&\tilde{Q}_{k,h}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \\
&\leq 2\psi_{k,h}(s,a) + \hat{P}_k(s,a)^\top(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) \\
&= 2\psi_{k,h}(s,a) + (\hat{P}_k(s,a) - P_k(s,a))^\top(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) + P_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1}) \\
&\leq 2\psi_{k,h}(s,a) + V_{h+1}^{\max}\|\hat{P}_k(s,a) - P_k(s,a)\|_1 + P_k(s,a)(\tilde{V}_{k,h+1} - \underset{\sim}{V}_{k,h+1})
\end{aligned}$$

and by construction we also can bound $\tilde{Q}_{k,h}(s,a) - \underset{\sim}{Q}_{k,h}(s,a) \leq V_h^{\max}$. Applying both bounds above recursively, we arrive at

$$\epsilon_k = (\tilde{V}_{k,1} - \underset{\sim}{V}_{k,1})(s_{k,1}) = (\tilde{Q}_{k,1} - \underset{\sim}{Q}_{k,1})(s_{k,1}, \pi_k(s_{k,1}, 1))$$
$$\leq \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{h=1}^{H} \mathbb{P}(s_h = s, a_h = a | s_{k,1}, \pi_k)[V_h^{\max} \wedge (2\psi_{k,h}(s,a) + V_{h+1}^{\max} \|\hat{P}_k(s,a) - P_k(s,a)\|_1)]$$
$$\leq \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{h=1}^{H} \mathbb{P}(s_h = s, a_h = a | s_{k,1}, \pi_k)[V_h^{\max} \wedge (2\alpha_{k,s,a} \|x_k^{(r)}\|_{(N_{k,s,a}^{(r)})^{-1}} + 3V_{h+1}^{\max} S\gamma_{k,s,a} \|x_k^{(p)}\|_{(N_{k,s,a}^{(p)})^{-1}})]$$

where we used Lemma 73 in the last step. We are now ready to bound the cumulative certificates after T episodes as

$$\sum_{k=1}^{T} \epsilon_k$$
$$\leq \sum_{k=1}^{T} \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{h=1}^{H} \mathbb{P}(s_{k,h} = s, a_{k,h} = a | s_{k,1}, \pi_k)[V_h^{\max} \wedge (2\alpha_{k,s,a} \|x_k^{(r)}\|_{(N_{k,s,a}^{(r)})^{-1}} + 3V_{h+1}^{\max} S\gamma_{k,s,a} \|x_k^{(p)}\|_{(N_{k,s,a}^{(p)})^{-1}})]$$

$$\leq \sum_{k=1}^{T} \sum_{h=1}^{H} [V_h^{\max} \wedge (2\alpha_{k,s_{k,h},a_{k,h}} \|x_k^{(r)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(r)})^{-1}} + 3V_{h+1}^{\max} S\gamma_{k,s_{k,h},a_{k,h}} \|x_k^{(p)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(p)})^{-1}})]$$
$$+ \sum_{k=1}^{T} \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \sum_{h=1}^{H} [\mathbb{P}(s_{k,h} = s, a_{k,h} = a | s_{k,1}, \pi_k) - \mathbf{1}\{s = s_{k,h}, a = a_{k,h}\}] V_h^{\max}$$

applying definition of failure event F^O

$$\leq \sum_{k=1}^{T} \sum_{h=1}^{H} [V_h^{\max} \wedge (2\alpha_{k,s_{k,h},a_{k,h}} \|x_k^{(r)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(r)})^{-1}} + 3V_{h+1}^{\max} S\gamma_{k,s_{k,h},a_{k,h}} \|x_k^{(p)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(p)})^{-1}})]$$
$$+ V_1^{\max} SH \sqrt{T \log \frac{6\log(2T)}{\delta'}}$$

splitting reward and transition terms

$$\leq \sum_{k=1}^{T} \sum_{h=1}^{H} [V_h^{\max} \wedge 2\alpha_{k,s_{k,h},a_{k,h}} \|x_k^{(r)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(r)})^{-1}}] \tag{5.33}$$
$$+ \sum_{k=1}^{T} \sum_{h=1}^{H} [V_h^{\max} \wedge 3V_{h+1}^{\max} S\gamma_{k,s_{k,h},a_{k,h}} \|x_k^{(p)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(p)})^{-1}}] \tag{5.34}$$
$$+ V_1^{\max} SH \sqrt{T \log \frac{6\log(2T)}{\delta'}}. \tag{5.35}$$

Before bounding the first two terms further, we first derive the following useful inequality using AM-GM inequality which holds for any $s \in \mathcal{A}$ and $s \in \mathcal{S}$

$$\log \frac{\det N_{k,s,a}^{(r)}}{\det(\lambda I)} \leq \log \frac{\left(\frac{1}{d}^{(r)} \operatorname{tr} N_{k,s,a}^{(r)}\right)^{d^{(r)}}}{\lambda^{d^{(r)}}} = d^{(r)} \log \frac{\operatorname{tr} N_{k,s,a}^{(r)}}{d^{(r)}\lambda} \leq d^{(r)} \log \frac{d^{(r)}\lambda + \xi_{x^{(r)}}^2 (k-1)H}{d^{(r)}\lambda} \tag{5.36}$$

where in the last inequality we used the fact that $N^{(r)}_{k,s,a}$ is the sum of λI and at most $H(k-1)$ outer products of feature vectors. Analogously, the following inequality holds for the covariance matrix of the transition features

$$\log \frac{\det N^{(p)}_{k,s,a}}{\det(\lambda I)} \leq d^{(p)} \log \frac{d^{(p)}\lambda + \xi^2_{x^{(p)}}(k-1)H}{d^{(p)}\lambda}.$$

This inequality allows us to upper-bound for $k \leq T$

$$\begin{aligned}
\alpha_{k,s_{k,h},a_{k,h}} =& \sqrt{\lambda}\xi_{\theta^{(r)}} + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}\log\frac{\det N^{(r)}_{k,s_{k,h},a_{k,h}}}{\det(\lambda I)}} \\
\leq& \sqrt{\lambda}\xi_{\theta^{(r)}} + \sqrt{\frac{1}{2}\log\frac{1}{\delta'} + \frac{1}{4}d^{(r)}\log\frac{d^{(r)}\lambda + \xi^2_{x^{(r)}}(k-1)H}{d^{(r)}\lambda}} \\
\leq& \sqrt{\lambda}\xi_{\theta^{(r)}} + \sqrt{\frac{1}{2}d^{(r)}\log\frac{d\lambda + \xi^2_{x^{(r)}}HT}{d^{(r)}\lambda\delta'}}
\end{aligned}$$

using the fact that $\sqrt{a}+\sqrt{b} \leq 2\sqrt{a+b}$ for all $a, b \in \mathbb{R}_+$

$$\begin{aligned}
\leq& 2\sqrt{\lambda\xi^2_{\theta^{(r)}} + \frac{1}{2}d^{(r)}\log\frac{d\lambda + \xi^2_{x^{(r)}}HT}{d^{(r)}\lambda\delta'}} \\
\leq& 2V_1^{\max}\sqrt{\frac{1}{4} + \lambda\xi^2_{\theta^{(r)}} + \frac{1}{2}d^{(r)}\log\frac{d^{(r)}\lambda + \xi^2_{x^{(r)}}HT}{d^{(r)}\lambda\delta'}} =: \alpha_T.
\end{aligned}$$

Note that the last inequality ensures $\alpha_T \geq V_1^{\max}$. We now use α_T to bound the term in Equation (5.33)

$$\begin{aligned}
&\sum_{k=1}^{T}\sum_{h=1}^{H}[V_h^{\max} \wedge 2\alpha_{k,s_{k,h},a_{k,h}}\|x_k^{(r)}\|_{(N^{(r)}_{k,s_{k,h},a_{k,h}})^{-1}}] \leq \sum_{k=1}^{T}\sum_{h=1}^{H}[V_h^{\max} \wedge 2\alpha_T\|x_k^{(r)}\|_{(N^{(r)}_{k,s_{k,h},a_{k,h}})^{-1}}] \\
\leq& 2\alpha_T\sum_{k=1}^{T}\sum_{h=1}^{H}[1 \wedge \|x_k^{(r)}\|_{(N^{(r)}_{k,s_{k,h},a_{k,h}})^{-1}}]
\end{aligned}$$

using Cauchy-Schwarz inequality

$$\leq \sqrt{4\alpha_T^2 TH\sum_{k=1}^{T}\sum_{h=1}^{H}[1 \wedge \|x_k^{(r)}\|^2_{(N^{(r)}_{k,s_{k,h},a_{k,h}})^{-1}}]}. \tag{5.37}$$

Leveraging Lemma 78, we can bound the elliptical potential inside the square-root as

$$\begin{aligned}
\sum_{k=1}^{T}\sum_{h=1}^{H}[1 \wedge \|x_k^{(r)}\|^2_{(N^{(r)}_{k,s_{k,h},a_{k,h}})^{-1}}] &= \sum_{s,a\in\mathcal{S}\times\mathcal{A}}\sum_{h=1}^{H}\sum_{k=1}^{T}\mathbf{1}\{s=s_{k,h}, a=a_{k,h}\}[1 \wedge \|x_k^{(r)}\|^2_{(N^{(r)}_{k,s,a})^{-1}}] \\
&\leq \sum_{s,a\in\mathcal{S}\times\mathcal{A}} H\sum_{k=1}^{T}[1 \wedge \|x_k^{(r)}\|^2_{(N^{(r)}_{k,s,a})^{-1}}] \leq \sum_{s,a\in\mathcal{S}\times\mathcal{A}} 2H\log\frac{\det N^{(r)}_{k,s,a}}{\det\lambda I}
\end{aligned}$$

applying Equation (5.36)

$$\leq 2SAHd^{(r)}\log\frac{d^{(r)}\lambda + \xi^2_{x^{(r)}}HT}{d^{(r)}\lambda}$$

and applying the definition of α_T

$$\leq 2SAH\frac{\alpha_T^2}{2(V_1^{\max})^2} \leq \frac{SAH\alpha_T^2}{(V_1^{\max})^2}\,.$$

We plug this bound back in (5.37) to get

$$\begin{aligned}
&\sum_{k=1}^{T}\sum_{h=1}^{H}[V_h^{\max} \wedge 2\alpha_{k,s_{k,h},a_{k,h}}\|x_k^{(r)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(r)})^{-1}}] \\
\leq&\sqrt{\frac{4\alpha_T^4 SAH^2T}{(V_1^{\max})^2}} = \sqrt{SAH^2T}\frac{2\alpha_T^2}{V_1^{\max}} \\
\leq&\sqrt{SAH^2T}V_1^{\max}\left[2 + 8\lambda\xi_{\theta^{(r)}}^2 + 4d^{(r)}\log\frac{d^{(r)}\lambda + \xi_{x^{(r)}}^2 HT}{d^{(r)}\lambda\delta'}\right]. \qquad (5.38)
\end{aligned}$$

After deriving this upper bound on the term in Equation (5.33), we bound the term in Equation (5.34) in similar fashion. We start with an upper bound on $S\gamma_{k,s_{k,h},a_{k,h}}$ which holds for $k \leq T$:

$$S\gamma_{k,s_{k,h},a_{k,h}} \leq \sqrt{1 + 4\lambda S^2\xi_{\theta^{(p)}}^2 + 2S^2d^{(p)}\log\frac{d^{(p)}\lambda + \xi_{x^{(p)}}^2 HT}{d^{(p)}\lambda\delta'}} =: \gamma_T\,,$$

which is by construction at least 1. We now use this definition to bound as above

$$\sum_{k=1}^{T}\sum_{h=1}^{H}[V_h^{\max} \wedge 3V_{h+1}^{\max}S\gamma_{k,s_{k,h},a_{k,h}}\|x_k^{(p)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(p)})^{-1}}] \leq 3V_1^{\max}\gamma_T\sum_{k=1}^{T}\sum_{h=1}^{H}[1 \wedge \|x_k^{(p)}\|_{(N_{k,s_{k,h},a_{k,h}}^{(p)})^{-1}}]$$

$$\leq 3V_1^{\max}\gamma_T\sqrt{TH\sum_{k=1}^{T}\sum_{h=1}^{H}[1 \wedge \|x_k^{(p)}\|^2_{(N_{k,s_{k,h},a_{k,h}}^{(p)})^{-1}}]} \leq 3V_1^{\max}\gamma_T\sqrt{TH2SAHd^{(p)}\log\frac{d^{(p)}\lambda + \xi_{x^{(p)}}^2 HT}{d^{(p)}\lambda}}$$

$$\leq 3V_1^{\max}\gamma_T\sqrt{2SAH^2T\frac{\gamma_T^2}{2S^2}} \leq \sqrt{S^3AH^2T}V_1^{\max}\left[3 + 12\lambda\xi_{\theta^{(p)}}^2 + 6d^{(p)}\log\frac{d^{(p)}\lambda + \xi_{x^{(p)}}^2 HT}{d^{(p)}\lambda\delta'}\right]. \qquad (5.39)$$

Combining (5.35), (5.38) and (5.39), the cumulative certificates after T episodes are bounded by

$$\begin{aligned}
\sum_{k=1}^{T}\epsilon_k \leq&\sqrt{S^3AH^2T}V_1^{\max}\left[14 + 12\lambda(\xi_{\theta^{(p)}}^2 + \xi_{\theta^{(r)}}^2) + 6(d^{(p)} + d^{(r)})\log\frac{(d^{(p)} + d^{(r)})\lambda + (\xi_{x^{(p)}}^2 + \xi_{x^{(r)}}^2)HT}{(d^{(p)} \wedge d^{(r)})\lambda\delta'}\right] \\
&+ V_1^{\max}SH\sqrt{T\log\frac{6\log(2T)}{\delta'}} \\
=&\tilde{O}\left(\sqrt{S^3AH^2T}V_1^{\max}\lambda(\xi_{\theta^{(p)}}^2 + \xi_{\theta^{(r)}}^2 + d^{(p)} + d^{(r)})\log\frac{\xi_{x^{(p)}}^2 + \xi_{x^{(r)}}^2}{\lambda\delta}\right).
\end{aligned}$$

□

5.10.4 Proof of Theorem 55

We are now ready to assemble the arguments above and prove the cumulative IPOC bound for Algorithm 5:

Proof. By Lemma 72, the failure event F has probability at most δ. Outside the failure event, for every episode k, the upper and lower Q-value estimates are valid upper bounds on the optimal Q-function and lower bounds on the Q-function of the current policy π_k, respectively (Lemmas 74 and 75). Further, Lemma 76 shows that the cumulative certificates grow at the desired rate

$$\tilde{O}\left(\sqrt{S^3AH^2T}V_1^{\max}\lambda(\xi_{\theta^{(p)}}^2+\xi_{\theta^{(r)}}^2+d^{(p)}+d^{(r)})\log\frac{\xi_{x^{(p)}}^2+\xi_{x^{(r)}}^2}{\lambda\delta}\right).$$

□

5.10.5 Technical Lemmas

We now state two existing technical lemmas used in our proof.

Lemma 77 (Elliptical confidence sets; Theorem 20.1 in Lattimore and Czepesvari (2018))**.** *Let $\lambda > 0$, $\theta \in \mathbb{R}^d$ and $(r_i)_{i\in\mathbb{N}}$ and $(x_i)_{i\in\mathbb{N}}$ random processes adapted to a filtration $\mathcal{F}_i$ so that $r_i - x_i^\top\theta$ are conditionally 1-sub-Gaussian. Then with probability at least $1-\delta$ for all $k \in \mathbb{N}$*

$$\|\theta-\tilde{\theta}_k\|_{N_k(\lambda)} \le \sqrt{\lambda}\|\theta\|_2 + \sqrt{2\log\frac{1}{\delta}+\log\frac{\det(N_k(\lambda))}{\det(\lambda I)}}$$

where $N_k(\lambda) = \lambda I + \sum_{i=1}^k x_i x_i^\top$ is the covariance matrix and $\tilde{\theta}_k = N_k(\lambda)^{-1}\sum_{i=1}^k r_i x_i$ is the least-squares estimate.

Lemma 78 (Elliptical potential; Lemma 19.1 in Lattimore and Czepesvari (2018))**.** *Let $x_1,\dots,x_n \in \mathbb{R}^d$ with $L \ge \max_i \|x_i\|_2$ and $N_i = N_0 + \sum_{j=1}^i x_j x_j^\top$ with N_0 being psd. Then*

$$\sum_{i=1}^n 1 \wedge \|x_i\|_{N_{i-1}^{-1}} \le 2\log\frac{\det N_n}{\det N_0} \le 2d\log\frac{\mathrm{tr}(N_0)+nL^2}{d\det(N_0)^{1/d}}.$$

5.11 Mistake IPOC Bound for Algorithm 5?

By Proposition 52, a mistake IPOC bound is stronger than the cumulative version we proved for Algorithm 5. One might wonder whether Algorithm 5 also satisfies this stronger bound, but this is not the case:

Proposition 79. *For any $\epsilon < 1$, there is an MDP with linear side information such that Algorithm 5 outputs certificates $\epsilon_k \ge \epsilon$ infinitely often with probability 1.*

Proof. Consider a two-armed bandit where the two-dimensional context is identical to the deterministic reward for both actions. The context alternates between $x_A := \begin{bmatrix}(1+\epsilon)/2\\(1-\epsilon)/2\end{bmatrix}$ and $x_B := \begin{bmatrix}(1-\epsilon)/2\\(1+\epsilon)/2\end{bmatrix}$. That means in odd-numbered episodes, the agent receives reward $\frac{1+\epsilon}{2}$ for action 1 and reward $\frac{1-\epsilon}{2}$ for action 2 (bandit A) and conversely in even-numbered episodes (bandit B). Let $n_{A,i}$ and $n_{B,i}$ be the current number

of times action i was played in each bandit and $N_i = \text{diag}(n_{A,i} + \lambda, n_{B,i} + \lambda)$ the covariance matrix. One can show that the optimistic Q-value of action 2 in bandit A is lower bounded as

$$\begin{aligned}\tilde{Q}(2) \geq & \sqrt{\ln \det N_2} \|x_A\|_{N_2^{-1}} \wedge 1 \\ = & \sqrt{\frac{\ln(\lambda + n_{A,2}) + \ln(\lambda + n_{B,2})}{n_{A,2}}} \wedge 1. \end{aligned} \tag{5.40}$$

Assume now the agent stops playing action 2 in bandit A and playing action 1 in bandit B at some point. Then the denominator in Eq (5.40) stays constant but the numerator grows unboundedly as $n_{B,2} \to \infty$. That implies that $\tilde{Q}(2) \to 1$ but the optimistic Q-value for the other action $\tilde{Q}(1) \to \frac{1+\epsilon}{2} \leq 1$ approaches the true reward. Eventually $\tilde{Q}(2) > \tilde{Q}(1)$ and the agent will play the ϵ-suboptimal action 2 in bandit A again. Hence, Algorithm 5 has to output infinitely many $\epsilon_k \geq \epsilon$. □

This negative result is due to the non-decreasing nature of the ellipsoid confidence intervals. It does not rule out alternative algorithms with mistake IPOC bounds for this setting, but they would likely require entirely different estimators and confidence bounds.

5.12 Additional Experimental Results

5.12.1 More Details on Experimental Results in Contextual Problems

The results presented earlier in this chapter are generated on the following MDP with side information. It has $S = 10$ states, $A = 40$ actions, horizon of $H = 5$, reward context dimension $d^{(r)} = 10$, and transition context dimension $d^{(p)} = 1$. The transition context $x_k^{(p)}$ is always constant 1. We sample the reward parameters independently for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $i \in [d^{(r)}]$ as

$$\theta_{i,s,a}^{(r)} = X_{i,s,a} Y_{i,s,a}, \quad X_{i,s,a} \sim \text{Bernoulli}(0.5), \quad Y_{i,s,a} \sim \text{Unif}(0,1).$$

and the transition kernel for each $s \in \mathcal{S}, a \in \mathcal{A}$ as

$$P(s,a) = \theta_{s,a}^{(p)} \sim \text{Dirichlet}(\alpha^{(p)})$$

where $\alpha^{(p)} \in \mathbb{R}^S$ with $\alpha_i^{(p)} = 0.3$ for $i \in [S]$. The reward context is again sampled from a Dirichlet distribution with parameter $\alpha^{(r)} \in \mathbb{R}^{d^{(r)}}$ where $\alpha_i^{(r)} = 0.01$ for $i \leq 4$ in the first 2 million episodes and all other times $\alpha_i^{(r)} = 0.7$. This shift in context distribution after 2 million episodes simulates rare contexts becoming more frequent.

In addition, we applied Algorithm 5 to randomly generated contextual bandit problems ($S = H = 1$) with $d^{(r)} = 10$ dimensional context and 40 actions. We sample the reward parameters independently for all $s = 1$, $a \in \mathcal{A}$ and $i \in [d^{(r)}]$ as

$$\theta_{i,s,a}^{(r)} = X_{i,s,a} Y_{i,s,a}, \qquad X_{i,s,a} \sim \text{Bernoulli}(0.9), \quad Y_{i,s,a} \sim \text{Unif}(0,1).$$

The context in each episode is sampled from a Dirichlet distribution with parameter $\alpha \in \mathbb{R}^{d^{(r)}}$ where $\alpha_i = 0.7$ for $i \leq 7$ and $\alpha_i = 0.01$ for $i \geq 10$. This choice was made to simulate both frequent as well as a few rare context dimensions. The `ORLC-SI` algorithm was run for 8 million episodes and we changed context, certificate and policy only every 1000 episodes for faster experimentation. Figure 5.2 shows

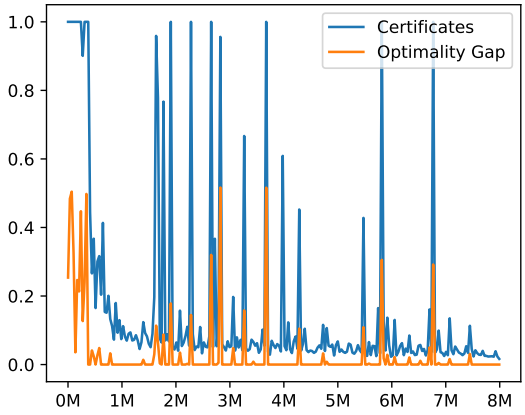

Figure 5.2: Results of ORLC-SI for 8M episodes on a linear contextual bandit problem; certificates are shown in blue and the true (unobserved) optimality gap in orange for increasing number of episodes.

certificates and optimality gaps of a representative run. Note that we sub-sampled the number of episodes shown for clearer visualization.

Certificates and optimality gaps have a correlation of 0.88 which confirms that certificates are informative about the policy's expected return. If one for example needs to intervene when the policy is more than 0.2 from optimal (e.g., by reducing the price for that customer), then in more than 42% of the cases where the certificate is above 0.2, the policy is worse than 0.2 suboptimal.

In both experiments, we use a slightly more complicated version of ORLC-SI listed in Algorithm 7 which computes the optimistic and pessimistic Q estimates $\tilde{Q}$ and $\underset{\sim}{Q}$ using subroutine ProbEstNorm in Algorithm 8. For the sake of clarity, we presented a simplified version with the same guarantees in the main text. While this simplified version of ORLC-SI does not leverage that the true transition kernel $P_k(s,a)$ has total mass 1, Algorithm 7 adds this as a constraint (see Lemma 80 below) similar to Abbasi-Yadkori and Neu (2014). This change yielded improved estimates empirically in our simulation. Note that this does not harm the theoretical properties. One can show the same cumulative IPOC bound for Algorithm 7 by slightly modifying the proof for Algorithm 5.

Lemma 80. *Let $\hat{p} \in [0,1]^d$, $\psi \geq 0$ and $v \in \mathbb{R}^d$ and define $\mathcal{P}_{\hat{p}} = \{p \in [0,1]^d \ : \ \leq \hat{p} - \psi \mathbf{1}_d \leq p \leq \hat{p} + \psi \mathbf{1}_d \ \wedge \ \|p\|_1 = 1\}$. Then, as long as $\mathcal{P}_{\hat{p}} \neq \emptyset$, the value returned by Algorithm 8 satisfies*

$$\begin{aligned} \text{ProbEstNorm}(\hat{p}, \psi, v) &= \max_{p \in \mathcal{P}_{\hat{p}}} p^\top v \\ -\text{ProbEstNorm}(\hat{p}, \psi, -v) &= \min_{p \in \mathcal{P}_{\hat{p}}} p^\top v \end{aligned}$$

and for any two $p, \tilde{p} \in \mathcal{P}_{\hat{p}}$ it holds that $|p^\top v - \tilde{p}^\top v| \leq \|v\|_1 \|p - \tilde{p}\|_\infty = 2\psi \|v\|_1$.

5.12.2 Empirical Comparison of Sample-Efficiency in Tabular Environments

The simulation study above and in Section 5.5 demonstrates that policy certificates can be a useful predictor for the (expected) performance of the algorithm in the next episode and the comparison of theoretical guarantees in Table 5.1 indicates the improved sample-efficiency of the tabular algorithm ORLC compared to existing approaches. However, do these tighter regret and PAC bounds indeed translate to an improved

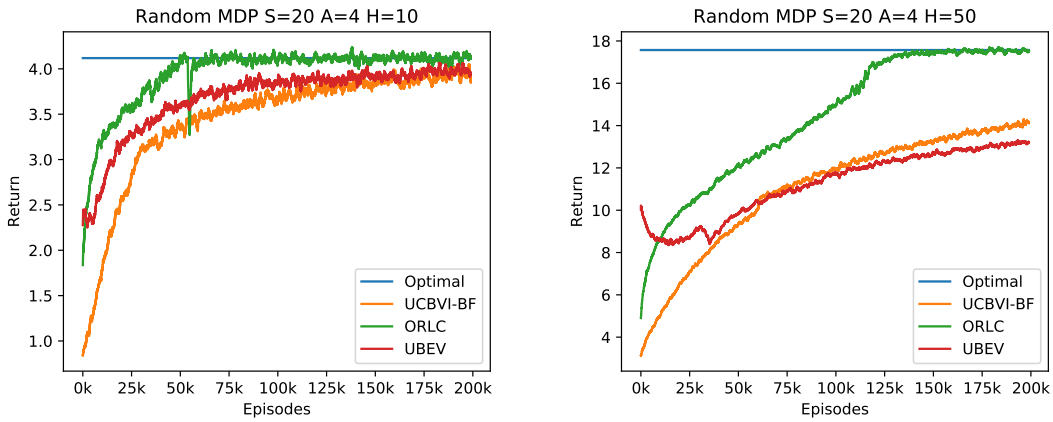

Figure 5.3: Experimental comparison of ORLC and existing approaches. The graph show the achieved sum of rewards per episode averaged over 1000 episodes each to generate smoothed curves. These results show representative single runs of each method on the same MDPs. Results are consistent across different random MDPs and different runs of the methods.

sample efficiency empirically? To answer this question, we compare ORLC against UCBVI-BF and UBEV, the methods with tightest regret and PAC bounds respectively and which we expect to perform the best among existing approaches.

We evaluate the methods on tabular MDPs which are randomly generated as follows: With probability 0.85 the average immediate reward $r(s,a)$ for any (s,a) is 0 and with probability 0.15 it is drawn from a uniform distribution $r(s,a) \sim \text{Unif}[0,1]$. The transition kernel for each $P(s,a) \sim \text{Dirichlet}(0.1)$ are drawn from a Dirichlet distribution with parameter $\alpha = 0.1$.

Figure 5.3 shows the performance of each method on MDPs with $S = 20$ states and $A = 4$ actions. The left plot shows the sum of rewards of each algorithm (averaged over a window of 1000 episodes) on a problem with small horizon $H = 10$. ORLC[4] converges to the optimal policy much faster than both UCBVI-BF and UBEV. Note that we adjusted UBEV to time-independent MDPs (the rewards and transition kernel do not depend on the time index within an episode) to make the comparison fair. On problems with larger horizon $H = 50$ the performance gap between ORLC and the competitors increases. Hence, even for problems of moderate horizon length compared $H \leq SA$, ORLC outperforms UCBVI-BF despite both methods having minimax-optimal regret bounds (in the dominant term) in this case. This difference can likely be attributed to the tighter optimism bonuses of ORLC as opposed to those of UCBVI-BF which are derived from an explicit regret-like bound with several additional approximations.

5.12.3 Policy Certificates in Problems with no Context

The simulation study above and in Section 5.5 demonstrate that policy certificates can be a useful predictor for the optimality gap of the algorithm in the next episode. However, the experimental results only consider problems with context and might therefore wonder whether simple baselines that only consider a certainty measure over contexts can produce similar results. We would like to emphasize that this is not the case as such baselines can only detect performance drops due to unfamiliar contexts but are blind to performance drops due to exploration on familiar contexts. To illustrate this point, consider a multi-armed bandit problem without context. The algorithm periodically (and less and less frequently) plays suboptimal arms

[4]We use Algorithm 6 with more refined bonuses compared to the simplified version in the main text.

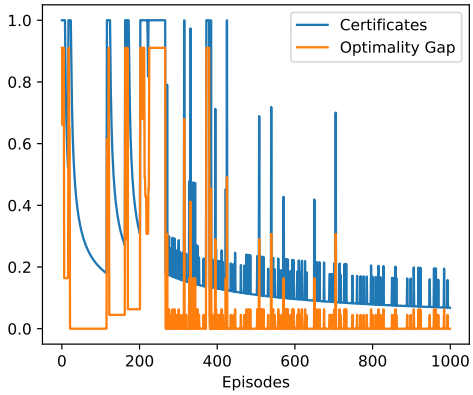

Figure 5.4: Performance and certificates of ORLC on a multi-armed bandit problem with 100 arms, generated randomly in the same way as tabular MDP instances above. Only every 10th episode is plotted to improve visibility of individual spikes.

until it can be sufficiently certain that these arms cannot be optimal. Consider Figure 5.4 where we plot the optimality gaps and optimality certificates of ORLC on a multi-armed bandit problem with $A = 100$ arms. We see occasional performance drop and as in the contextual case, our algorithm's certificates is able to predict them. A baseline that is only measuring the familiarity of context would completely fail in this case.

Algorithm 7: `ORLC-SI` algorithm with probability mass constraints

Input : failure prob. $\delta \in (0, 1]$, regularizer $\lambda > 0$

$\forall s, s' \in \mathcal{S}, a \in \mathcal{A}, h \in [H]$:

$N_{s,a}^{(p)} \leftarrow \lambda I_{d^{(p)} \times d^{(p)}}$; $\quad N_{s,a}^{(r)} \leftarrow \lambda I_{d^{(r)} \times d^{(r)}}$;

$M_{s,a}^{(r)} \leftarrow \vec{0}_{d^{(r)}}$; $\quad M_{s',s,a}^{(p)} \leftarrow \vec{0}_{d^{(p)}}$;

$\tilde{V}_{H+1} \leftarrow \vec{0}_S \quad \underset{\sim}{V}_{H+1} \leftarrow \vec{0}_S \quad V_h^{\max} \leftarrow (H - h + 1)$;

$\xi_{\theta^{(r)}} \leftarrow \sqrt{d}$; $\xi_{\theta^{(p)}} \leftarrow \sqrt{d} \quad \delta' \leftarrow \frac{\delta}{S(SA+A+H)}$;

$\phi(N, x, \xi) := \left[\sqrt{\lambda}\xi + \sqrt{\frac{1}{2}\ln\frac{1}{\delta'} + \frac{1}{4}\ln\frac{\det N}{\det(\lambda I)}}\right] \|x\|_{N^{-1}}$;

for $k = 1, 2, 3, \ldots$ **do**

  Observe current contexts $x_k^{(r)}$ and $x_k^{(p)}$;

  `/* estimate model with least squares */`

  for $s, s' \in \mathcal{S}, a \in \mathcal{A}$ **do**

    $\hat{\theta}_{s,a}^{(r)} \leftarrow (N_{s,a}^{(r)})^{-1} M_{s,a}^{(r)}$;

    $\hat{r}(s, a) \leftarrow 0 \vee (x_k^{(r)})^\top \hat{\theta}_{s,a}^{(r)} \wedge 1$;

    $\hat{\theta}_{s',s,a}^{(p)} \leftarrow (N_{s,a}^{(p)})^{-1} M_{s',s,a}^{(p)}$;

    $\hat{P}(s'|s, a) \leftarrow 0 \vee (x_k^{(p)})^\top \hat{\theta}_{s',s,a}^{(p)} \wedge 1$;

  `/* optimistic planning */`

  for $h = H$ **to** 1 ***and*** $s \in \mathcal{S}$ **do**

    for $a \in \mathcal{A}$ **do**

      $\tilde{\psi}_h(s, a) \leftarrow \phi(N_{s,a}^{(r)}, x_k^{(r)}, \xi_{\theta^{(r)}})$;

      $\underset{\sim}{\psi}_h(s, a) \leftarrow \phi(N_{s,a}^{(r)}, x_k^{(r)}, \xi_{\theta^{(r)}})$;

      $\tilde{Q}_h(s, a) \leftarrow \hat{r}(s, a) + \text{ProbEstNorm}(\hat{P}(s, a), \phi(N_{s,a}^{(p)}, x_k^{(p)}, \xi_{\theta^{(p)}}), \tilde{V}_{h+1}) + \tilde{\psi}_h(s, a)$;

      $\underset{\sim}{Q}_h(s, a) \leftarrow \hat{r}(s, a) - \text{ProbEstNorm}(\hat{P}(s, a), \phi(N_{s,a}^{(p)}, x_k^{(p)}, \xi_{\theta^{(p)}}), -\underset{\sim}{V}_{h+1}) - \underset{\sim}{\psi}_h(s, a)$;

      `// clip values`

      $\tilde{Q}_h(s, a) \leftarrow 0 \vee \tilde{Q}_h(s, a) \wedge V_h^{\max}$;

      $\underset{\sim}{Q}_h(s, a) \leftarrow 0 \vee \underset{\sim}{Q}_h(s, a) \wedge V_h^{\max}$;

    $\pi_k(s, h) \leftarrow \text{argmax}_a \tilde{Q}_h(s, a)$;

    $\tilde{V}_h(s) \leftarrow \tilde{Q}_h(s, \pi_k(s, t))$;

    $\underset{\sim}{V}_h(s) \leftarrow \underset{\sim}{Q}_h(s, \pi_k(s, t))$;

  `/* Execute policy for one episode */`

  $s_{k,1} \sim P_0$;

  $\epsilon_k \leftarrow \tilde{V}_1(s_{k,1}) - \underset{\sim}{V}_1(s_{k,1})$;

  output policy π_k **with certificate** ϵ_k;

  for $h = 1$ **to** H **do**

    $a_{k,h} \leftarrow \pi_k(s_{k,h}, h)$;

    $r_{k,h} \sim P_R(s_{k,h}, a_{k,h}) \quad s_{k,h+1} \sim P(s_{k,h}, a_{k,h})$;

    `// Update statistics`

    $N_{s_{k,h},a_{k,h}}^{(p)} \leftarrow N_{s_{k,h},a_{k,h}}^{(p)} + x_k^{(p)}(x_k^{(p)})^\top$;

    $N_{s_{k,h},a_{k,h}}^{(r)} \leftarrow N_{s_{k,h},a_{k,h}}^{(r)} + x_k^{(r)}(x_k^{(r)})^\top$;

    $M_{s_{k,h+1},s_{k,h},a_{k,h}}^{(p)} \leftarrow M_{s_{k,h+1},s_{k,h},a_{k,h}}^{(p)} + x_k^{(p)}$;

    $M_{s_{k,h},a_{k,h}}^{(p)} \leftarrow M_{s_{k,h},a_{k,h}}^{(p)} + x_k^{(p)}$;

Algorithm 8: ProbEstNorm($\hat{p}$, ψ, v) function to compute normalized estimated expectation of v

Input : estimated probability vector $\hat{p} \in [0,1]^S$
Input : confidence width $\psi \in \mathbb{R}_+$
Input : value vector $v \in \mathbb{R}^S$

Compute sorting σ of v so that $v_{\sigma_i} \geq v_{\sigma_j}$ for all $i \leq j$;
$p \leftarrow \hat{p} - \psi \vee 0$;
$m \leftarrow p^\top \mathbf{1}$;
$r \leftarrow 0$;
for $i \in [S]$ **do**
    $s \leftarrow m \wedge ((\hat{p}_{\sigma_i} + \psi \wedge 1) - p_{\sigma_i})$;
    $m \leftarrow m - s$;
    $r \leftarrow r + v_{\sigma_i}(p_{\sigma_i} + s)$;

Return : r

Chapter 6

Oracle-Efficient PAC Reinforcement Learning with Rich Observations

This chapter is based on work I did during an internship with collaborators at Microsoft Research, New York City. It was published as:
Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. "On Oracle-Efficient PAC Reinforcement Learning with Rich Observations". In: *Advances in neural information processing systems* (2018)

6.1 Introduction

We study episodic reinforcement learning (RL) in environments with realistically rich observations such as images or text, which we refer to broadly as *contextual decision processes*. We aim for methods that use function approximation in a provably effective manner to find the best possible policy through strategic exploration.

While such problems are central to empirical RL research (Mnih et al., 2015), most theoretical results on strategic exploration focus on tabular MDPs with small state spaces (Kearns and Singh, 2002; Brafman and Tennenholtz, 2003; Strehl and Littman, 2005; Strehl, Li, Wiewiora, et al., 2006; Auer, Jaksch, and Ortner, 2009; Dann and Brunskill, 2015; Azar, Osband, and Munos, 2017; Dann, Lattimore, and Brunskill, 2017). Comparatively little work exists on provably effective exploration with large observation spaces that require generalization through function approximation. The few algorithms that do exist either have poor sample complexity guarantees (e.g., Kakade, Kearns, and Langford, 2003; Pazis and Parr, 2013; Grande, Walsh, and How, 2014; Pazis and Parr, 2016) or require fully deterministic environments (Wen and Van Roy, 2013; Wen and Van Roy, 2017) and are therefore inapplicable to most real-world applications and modern empirical RL benchmarks. This scarcity of positive results on efficient exploration with function approximation can likely be attributed to the challenging nature of this problem rather than a lack of interest by the research community.

On the statistical side, recent important progress was made by showing that contextual decision processes (CDPs) with rich stochastic observations and deterministic dynamics over M hidden states can be learned with a sample complexity polynomial in M (Krishnamurthy, Agarwal, and Langford, 2016). This was followed by an algorithm called `OLIVE` (Jiang, Krishnamurthy, et al., 2017) that enjoys a polynomial sample complexity guarantee for a broader range of CDPs, including ones with stochastic hidden state transitions. While encouraging, these efforts focused exclusively on statistical issues, ignoring computation altogether. Specifically, the proposed algorithms exhaustively enumerate candidate value functions to

eliminate the ones that violate Bellman equations, an approach that is computationally intractable for any function class of practical interest. Thus, while showing that RL with rich observations can be statistically tractable, these results leave open the question of computational feasibility.

In this paper, we focus on this difficult computational challenge. We work in an oracle model of computation, meaning that we aim to design sample-efficient algorithms whose computation can be reduced to common optimization primitives over function spaces, such as linear programming and cost-sensitive classification. The oracle-based approach has produced practically effective algorithms for active learning (Hsu, 2010), contextual bandits (Agarwal, Hsu, et al., 2014), structured prediction (Ross and Bagnell, 2014; Chang et al., 2015), and multi-class classification (Allwein, Schapire, and Singer, 2000), and here, we consider oracle-based algorithms for challenging RL settings.

We begin by studying the setting of Krishnamurthy, Agarwal, and Langford (2016) with deterministic dynamics over M hidden states and stochastic rich observations. In Section 6.4, we use cost-sensitive classification and linear programming oracles to develop VALOR, the first algorithm that is both *computationally* and *statistically* efficient for this setting. While deterministic hidden-state dynamics are somewhat restrictive, the model is considerably more general than fully deterministic MDPs assumed by prior work (Wen and Van Roy, 2013; Wen and Van Roy, 2017), and it accurately captures modern empirical benchmarks such as visual grid-worlds in Minecraft (Johnson et al., 2016). As such, this method represents a considerable advance toward provably efficient RL in practically relevant scenarios.

Nevertheless, we ultimately seek efficient algorithms for more general settings, such as those with stochastic hidden-state transitions. Working toward this goal, we study the computational aspects of the OLIVE algorithm (Jiang, Krishnamurthy, et al., 2017), which applies to a wide range of environments. Unfortunately, in Section 6.5.1, we show that OLIVE *cannot* be implemented efficiently in the oracle model of computation. As OLIVE is the only known statistically efficient approach for this general setting, our result establishes a significant barrier to computational efficiency. We also describe two other oracle-based algorithms for the deterministic-dynamics setting that are considerably different from VALOR. The negative results identify where the hardness lies while the positive results provide a suite of new algorithmic tools. Together, these results advance our understanding of efficient reinforcement learning with rich observations.

6.2 Related Work

There is abundant work on strategic exploration in the tabular setting (Kearns and Singh, 2002; Brafman and Tennenholtz, 2003; Strehl and Littman, 2005; Strehl, Li, Wiewiora, et al., 2006; Auer, Jaksch, and Ortner, 2009; Dann and Brunskill, 2015; Azar, Osband, and Munos, 2017; Dann, Lattimore, and Brunskill, 2017). The computation in these algorithms often involves planning in optimistic models and can be solved efficiently via dynamic programming. To extend the theory to the more practical settings of large state spaces, typical approaches include (1) distance-based state identity test under smoothness assumptions (e.g., Kakade, Kearns, and Langford, 2003; Pazis and Parr, 2013; Grande, Walsh, and How, 2014; Pazis and Parr, 2016), or (2) working with factored MDPs (e.g., Kearns and Koller, 1999). The former approach is similar to the use of state abstractions Li, Walsh, and Littman, 2006, and typically incurs exponential sample complexity in state dimension. The latter approach does have sample-efficient results, but the factored representation assumes relatively disentangled state variables which cannot model rich sensory inputs (such as images).

Azizzadenesheli, Lazaric, and Anandkumar (2016a) have studied regret minimization in rich observation MDPs, a special case of contextual decision processes with a small number of hidden states and reactive policies. They do not utilize function approximation, and hence incur polynomial dependence on the number of unique observations in both sample and computational complexity. Therefore, this approach,

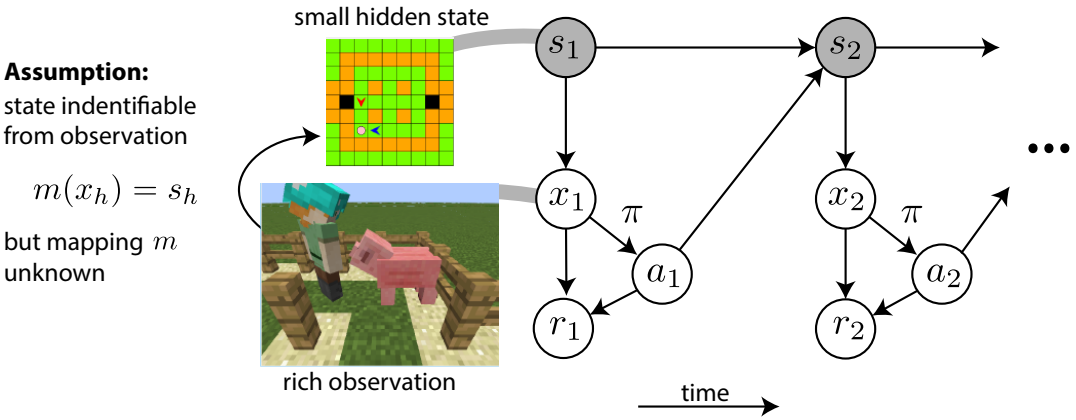

Figure 6.1: Graphical representation of the problem class considered by our algorithm, VALOR: The main assumptions that enable sample-efficient learning are (1) that the small hidden state s_h is identifiable from the rich observation x_h and (2) that the next state is a deterministic function of the previous state and action. State and observation examples are from https://github.com/Microsoft/malmo-challenge.

along with related works Azizzadenesheli, Lazaric, and Anandkumar, 2016b; Guo, Doroudi, and Brunskill, 2016, does not scale to the rich observation settings that we focus on here.

Wen and Van Roy (2013) and Wen and Van Roy (2017) have studied exploration with function approximation in fully deterministic MDPs, which is considerably more restrictive than our setting of deterministic hidden state dynamics with stochastic observations and rewards. Moreover, their analysis measures representation complexity using *eluder dimension* Russo and Van Roy, 2013; Osband and Van Roy, 2014, which is only known to be small for some simple function classes. In comparison, our bounds scale with more standard complexity measures and can easily extend to VC-type quantities, which allows our theory to apply to practical and popular function approximators including neural networks (Anthony and Bartlett, 2009).

6.3 Setting and Background

We consider reinforcement learning (RL) in a common special case of contextual decision processes (Krishnamurthy, Agarwal, and Langford, 2016; Jiang, Krishnamurthy, et al., 2017), sometimes referred to as rich observation MDPs (Azizzadenesheli, Lazaric, and Anandkumar, 2016a). We assume an H-step process where in each episode, a random *trajectory* $s_1, x_1, a_1, r_1, s_2, x_2, \ldots, s_H, x_H, a_H, r_H$ is generated. For each time step (or *level*) $h \in [H]$, $s_h \in \mathcal{S}$ where $\mathcal{S}$ is a finite hidden state space, $x_h \in \mathcal{X}$ where $\mathcal{X}$ is the rich observation (context) space, $a_h \in \mathcal{A}$ where $\mathcal{A}$ is a finite action space of size K, and $r_h \in \mathbb{R}$. Each hidden state $s \in \mathcal{S}$ is associated with an emission process $O_s \in \Delta(\mathcal{X})$, and we use $x \sim s$ as a shorthand for $x \sim O_s$. We assume that each rich observation contains enough information so that s can in principle be identified just from $x \sim O_s$—hence x is a Markov state and the process is in fact an MDP over $\mathcal{X}$—but the mapping $x \mapsto s$ is unavailable to the agent and s is never observed. The hidden states $\mathcal{S}$ introduce structure into the problem, which is essential since we allow the observation space $\mathcal{X}$ to be infinitely large.[1] The issue of partial observability is not the focus of the paper.

[1]Indeed, the lower bound in Proposition 6 in Jiang, Krishnamurthy, et al. (2017) show that ignoring underlying structure precludes provably-efficient RL, even with function approximation.

Let $\Gamma : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ define transition dynamics over the hidden states, and let $\Gamma_1 \in \Delta(\mathcal{S})$ denote an initial distribution over hidden states. $R : \mathcal{X} \times \mathcal{A} \to \Delta(\mathbb{R})$ is the reward function; this differs from partially observable MDPs where reward depends only on s, making the problem more challenging. With this notation, a trajectory is generated as follows: $s_1 \sim \Gamma_1$, $x_1 \sim s_1$, $r_1 \sim R(x_1, a_1)$, $s_2 \sim \Gamma(s_1, a_1)$, $x_2 \sim s_2, \ldots, s_H \sim \Gamma(s_{H-1}, a_{H-1}), x_H \sim s_H, r_H \sim R(x_H, a_H)$, with actions $a_{1:H}$ chosen by the agent. We emphasize that $s_{1:H}$ are unobservable to the agent.

To simplify notation, we assume that each observation and hidden state can only appear at a particular level. This implies that $\mathcal{S}$ is partitioned into $\{\mathcal{S}_h\}_{h=1}^H$ with size $M := \max_{h \in [H]} |\mathcal{S}_h|$. For regularity, assume $r_h \geq 0$ and $\sum_{h=1}^H r_h \leq 1$ almost surely.

In this setting, the learning goal is to find a policy $\pi : \mathcal{X} \to \mathcal{A}$ that maximizes the expected return $V^\pi := \mathbb{E}[\sum_{h=1}^H r_h \,|\, a_{1:H} \sim \pi]$. Let $\pi^\star$ denote the optimal policy, which maximizes V^π, with optimal value function $g^\star$ defined as $g^\star(x) := \mathbb{E}[\sum_{h'=h}^H r_{h'} | x_h = x, a_{h:H} \sim \pi^\star]$. As is standard, $g^\star$ satisfies the Bellman equation: $\forall x$ at level h,

$$g^\star(x) = \max_{a \in \mathcal{A}} \mathbb{E}[r_h + g^\star(x_{h+1}) | x_h = x, a_h = a],$$

with the understanding that $g^\star(x_{H+1}) \equiv 0$. A similar equation holds for the optimal Q-value function $Q^\star(x, a) := \mathbb{E}[\sum_{h'=h}^H r_{h'} | x_h = x, a_h = a, a_{h+1:H} \sim \pi^\star]$, and $\pi^\star = \operatorname{argmax}_{a \in \mathcal{A}} Q^\star(x, a)$.[2]

Below are two special cases of the setting described above that will be important for later discussions.
Tabular MDPs: An MDP with a finite and small state space is a special case of this model, where $\mathcal{X} = \mathcal{S}$ and O_s is the identity map for each s. This setting is relevant in our discussion of oracle-efficiency of the existing `OLIVE` algorithm in Section 6.5.1.
Deterministic dynamics over hidden states: Our algorithm, `VALOR`, works in this special case, which requires Γ_1 and $\Gamma(s, a)$ to be point masses. Originally proposed by Krishnamurthy, Agarwal, and Langford (2016), this setting can model some challenging benchmark environments in modern reinforcement learning, including visual grid-worlds common to the deep RL literature (e.g., Johnson et al., 2016). In such tasks, the state records the position of each game element in a grid but the agent observes a rendered 3D view. Figure 6.1 shows a visual summary of this setting. We describe `VALOR` in detail in Section 6.4.

Throughout the paper, we use $\hat{\mathbf{E}}_D[\cdot]$ to denote empirical expectation over samples from a data set D.

6.3.1 Function Classes and Optimization Oracles

As $\mathcal{X}$ can be rich, the agent must use function approximation to generalize across observations. To that end, we assume a given value function class $\mathcal{G} \subset (\mathcal{X} \to [0, 1])$ and policy class $\Pi \subset (\mathcal{X} \to \mathcal{A})$. Our algorithm is agnostic to the specific function classes used, but for the guarantees to hold, they must be expressive enough to represent the optimal value function and policy, that is, $\pi^\star \in \Pi$ and $g^\star \in \mathcal{G}$. Prior works often use $\mathcal{F} \subset (\mathcal{X} \times \mathcal{A} \to [0, 1])$ to approximate $Q^\star$ instead, but for example Jiang, Krishnamurthy, et al. (2017) point out that their `OLIVE` algorithm can equivalently work with $\mathcal{G}$ and Π. This $(\mathcal{G}, \Pi)$ representation is useful in resolving the computational difficulty in the deterministic setting, and has also been used in practice (Dai et al., 2018).

When working with large and abstract function classes as we do here, it is natural to consider an oracle model of computation and assume that these classes support various optimization primitives. We adopt this *oracle-based* approach here, and specifically use the following oracles:

[2]Note that the optimal policy and value functions depend on x and not just s even if s was known, since reward is a function of x.

Cost-Sensitive Classification (CSC) on Policies. A cost-sensitive classification (CSC) oracle receives as inputs a parameter ϵ_{sub} and a sequence $\{(x^{(i)}, c^{(i)})\}_{i\in[n]}$ of observations $x^{(i)} \in \mathcal{X}$ and cost vectors $c^{(i)} \in \mathbb{R}^K$, where $c^{(i)}(a)$ is the cost of predicting action $a \in \mathcal{A}$ for $x^{(i)}$. The oracle returns a policy whose average cost is within ϵ_{sub} of the minimum average cost, $\min_{\pi\in\Pi} \frac{1}{n}\sum_{i=1}^{n} c^{(i)}(\pi(x^{(i)}))$. While CSC is NP-hard in the worst case, CSC can be further reduced to binary classification (Beygelzimer, Langford, and Ravikumar, 2009; Langford and Beygelzimer, 2005) for which many practical algorithms exist and actually form the core of empirical machine learning. As further motivation, the CSC oracle has been used in practically effective algorithms for contextual bandits (Langford and Zhang, 2008; Agarwal, Hsu, et al., 2014), imitation learning (Ross and Bagnell, 2014), and structured prediction (Chang et al., 2015).

Linear Programs (LP) on Value Functions. A linear program (LP) oracle considers an optimization problem where the objective $o : \mathcal{G} \to \mathbb{R}$ and the constraints $h_1, \ldots h_m$ are linear functionals of $\mathcal{G}$ generated by finitely many function evaluations. That is, o and each h_j have the form $\sum_{i=1}^{n} \alpha_i g(x_i)$ with coefficients $\{\alpha_i\}_{i\in[n]}$ and contexts $\{x_i\}_{i\in[n]}$. Formally, for a program of the form

$$\max_{g\in\mathcal{G}} o(g), \quad \text{subject to } h_j(g) \le c_j, \quad \forall j \in [m],$$

with constants $\{c_j\}_{j\in[m]}$, an LP oracle with approximation parameters $\epsilon_{\text{sub}}, \epsilon_{\text{feas}}$ returns a function $\hat{g}$ that is at most ϵ_{sub}-suboptimal and that violates each constraint by at most ϵ_{feas}. For intuition, if the value functions $\mathcal{G}$ are linear with parameter vector $\theta \in \mathbb{R}^d$, i.e., $g(x) = \langle\theta, x\rangle$, then this reduces to a linear program in $\mathbb{R}^d$ for which a plethora of provably efficient solvers exist. Beyond the linear case, such problems can be practically solved using standard continuous optimization methods. LP oracles are also employed in prior work focusing on deterministic MDPs (Wen and Van Roy, 2013; Wen and Van Roy, 2017).

Least-Squares (LS) Regression on Value Functions. We also consider a least-squares regression (LS) oracle that returns the value function which minimizes a square-loss objective. Since `VALOR` does not use this oracle, we defer details to later sections.

We define the following notion of oracle-efficiency based on the optimization primitives above.

Definition 81 (Oracle-Efficient)**.** *An algorithm is* oracle-efficient *if it can be implemented with polynomially many basic operations and calls to CSC, LP, and LS oracles.*

Note that our algorithmic results continue to hold if we include additional oracles in the definition, while our hardness results easily extend, provided that the new oracles can be efficiently implemented in the tabular setting (i.e., they satisfy Proposition 86; see Section 6.5).

6.4 `VALOR`: An Oracle-Efficient Algorithm

In this section we propose and analyze a new algorithm, `VALOR` (Values stored Locally for RL) shown in Algorithm 9 (with 10 & 11 as subroutines). As we will show, this algorithm is oracle-efficient and enjoys a polynomial sample-complexity guarantee in the deterministic hidden-state dynamics setting described earlier, which was originally introduced by Krishnamurthy, Agarwal, and Langford (2016).

Algorithm 9: VALOR (Values stored Locally for RL) Algorithm

Global: $\mathcal{D}_1, \dots \mathcal{D}_H$ initialized as $\emptyset$;
Function MetaAlg
  dfslearn $(\varnothing)$; // Alg.11
  for $k = 1, \dots, MH$ **do**
    $\hat{\pi}^{(k)}, \hat{V}^{(k)} \leftarrow$ polvalfun() ; // Alg.10
    $T \leftarrow$ sample n_{eval} trajectories with $\hat{\pi}^{(k)}$;
    $\hat{V}^{\hat{\pi}^{(k)}} \leftarrow$ average return of T;
    if $\hat{V}^{(k)} \leq \hat{V}^{\hat{\pi}^{(k)}} + \frac{\epsilon}{2}$ **then return** $\hat{\pi}^{(k)}$;
    for $h = 1 \dots H-1$ **do**
      for *all* $a_{1:h}$ *of* n_{expl} *traj.* $\in T$ **do**
        dfslearn $(a_{1:h})$; // Alg.11
  return failure;

Algorithm 10: VALOR Subroutine: Policy optimization with local values

Function polvalfun()
  $\hat{V}^\star \leftarrow V$ of the only dataset in $\mathcal{D}_1$;
  for $h = 1 : H$ **do**
    // CSC-oracle
    $\hat{\pi}_h \leftarrow \operatorname{argmax}_{\pi \in \Pi_h} \sum_{(D,V,\{V_a\}) \in \mathcal{D}_h} V_D(\pi; \{V_a\})$;
  return $\hat{\pi}_{1:H}, \hat{V}^\star$;

Notation:
$V_D(\pi; \{V_a\}) := \hat{\mathbf{E}}_D[K\mathbf{1}\{\pi(x) = a\}(r + V_a)]$

Algorithm 11: VALOR Subroutine: DFS Learning of local values

1 $\epsilon_{\text{feas}} = \epsilon_{\text{sub}} = \epsilon_{\text{stat}} = \tilde{O}(\epsilon^2/MH^3)$; // see exact values in Table 6.1
2 $\phi_h = (H + 1 - h)(6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}})$; // accuracy of learned values at level h
3 **Function** dfslearn (*path* p *with length* $h-1$)
4 **for** $a \in \mathcal{A}$ **do**
5 $D' \leftarrow$ Sample n_{test} trajectories with actions $p \circ a$;
  // compute optimistic / pessimistic values using LP-oracle
6 $V_{opt} \leftarrow \max_{g \in \mathcal{G}_{h+1}} \hat{\mathbf{E}}_{D'}[g(x_{h+1})]$ (and $V_{pes} \leftarrow \min_{g \in \mathcal{G}_{h+1}} \hat{\mathbf{E}}_{D'}[g(x_{h+1})]$)
  s.t. $\forall (D, V, \_) \in \mathcal{D}_{h+1}$: $|V - \hat{\mathbf{E}}_D[g(x_{h+1})]| \leq \phi_{h+1}$;
7 **if** $|V_{opt} - V_{pes}| \leq 2\phi_{h+1} + 4\epsilon_{stat} + 2\epsilon_{feas}$ **then**
8 $V_a \leftarrow (V_{opt} + V_{pes})/2$; // consensus among remaining functions
9 **else**
10 $V_a \leftarrow$ dfslearn$(p \circ a)$; // no consensus, descend
11 $\tilde{D} \leftarrow$ Sample n_{train} traj. with p and $a_h \sim \text{Unif}(K)$;
12 $\tilde{V} \leftarrow \max_{\pi \in \Pi_h} V_{\tilde{D}}(\pi; \{V_a\})$; // CSC-oracle
13 Add $(\tilde{D}, \tilde{V}, \{V_a\}_{a \in \mathcal{A}})$ to $\mathcal{D}_h$;
14 **return** $\tilde{V}$;

Since hidden states can be deterministically reached by sequences of actions (or *paths*), from an algorithmic perspective, the process can be thought of as an exponentially large tree where each node is associated with a hidden state (such association is unknown to the agent). Similar to LSVEE (Krishnamurthy, Agarwal, and Langford, 2016), VALOR first explores this tree (Line 3) with a form of depth first search (Algorithm 11). To avoid visiting all of the exponentially many paths, VALOR performs a state identity test (Algorithm 11, Lines 5–8): the data collected so far is used to (virtually) eliminate functions in $\mathcal{G}$ (Algorithm 11, Line 6), and we do not descend to a child if the remaining functions agree on the value of the child node (Algorithm 11, Line 7).

The state identity test prevents exploring the same hidden state twice but might also incorrectly prune unvisited states if all functions happen to agree on the value. Unfortunately, with no data from such

pruned states, we are unable to learn the optimal policy on them. To address this issue, after `dfslearn` returns, we first use the stored data and values (Line 5) to compute a policy (see Algorithm 10) that is near optimal on all explored states. Then, `VALOR` deploys the computed policy (Line 6) and only terminates if the estimated optimal value is achieved (Line 8). If not, the policy has good probability of visiting those accidentally pruned states (see Section 6.8.5), so we invoke `dfslearn` on the generated paths to complement the data sets (Line 11).

In the rest of this section we describe `VALOR` in more detail, and then state its statistical and computational guarantees. `VALOR` follows a dynamic programming style and learns in a bottom-up fashion. As a result, even given stationary function classes $(\mathcal{G}, \Pi)$ as inputs, the algorithm can return a non-stationary policy $\hat{\pi}_{1:H} := (\hat{\pi}_1, \ldots, \hat{\pi}_H) \in \Pi^H$ that may use different policies at different time steps.[3] To avoid ambiguity, we define $\Pi_h := \Pi$ and $\mathcal{G}_h := \mathcal{G}$ for $h \in [H]$, to emphasize the time point h under consideration. For convenience, we also define $\mathcal{G}_{H+1}$ to be the singleton $\{x \mapsto 0\}$. This notation also allows our algorithms to handle more general non-stationary function classes.

Details of depth-first search exploration. `VALOR` maintains many data sets collected at paths visited by `dfslearn`. Each data set D is collected from some path p, which leads to some hidden state s. (Due to determinism, we will refer to p and s interchangeably throughout this section.) D consists of tuples (x, a, r) where $x \sim p$ (i.e., $x \sim O_s$), $a \sim \mathrm{Unif}(K)$, and r is the instantaneous reward. Associated with D, we also store a scalar V which approximates $V^\star(s)$, and $\{V_a\}_{a\in\mathcal{A}}$ which approximate $\{V^\star(s \circ a)\}_{a\in\mathcal{A}}$, where $s \circ a$ denotes the state reached when taking a in s. The estimates $\{V_a\}_{a\in\mathcal{A}}$ of the future optimal values associated with the current path $p \in \mathcal{A}^{h-1}$ are either determined through a recursive call (Line 10), or through a *state-identity test* (Lines 5–8 in `dfslearn`). To check if we already know $V^\star(p \circ a)$, we solve constrained optimization problems to compute optimistic and pessimistic estimates, using a small amount of data from $p \circ a$. The constraints eliminate all $g \in \mathcal{G}_{h+1}$ that make incorrect predictions for $V^\star(s')$ for any previously visited s' at level $h+1$. As such, if we have learned the value of $s \circ a$ on a different path, the optimistic and pessimistic values must agree ("consensus"), so we need not descend. Once we have the future values V_a, the value estimate $\tilde{V}$ (which approximates $V^\star(s)$) is computed (in Line 12) by maximizing the sum of immediate reward and future values, re-weighted using importance sampling to reflect the policy under consideration π:

$$V_D(\pi; \{V_a\}) := \hat{\mathbf{E}}_D[K\mathbf{1}\{\pi(x) = a\}(r + V_a)]. \tag{6.1}$$

Details of policy optimization and exploration-on-demand. `polvalfun` performs a sequence of policy optimization steps using all the data sets collected so far to find a non-stationary policy that is near-optimal at all explored states simultaneously. Note that this policy differs from that computed in (Alg. 11, Line 12) as it is common for all datasets at a level h. And finally using this non-stationary policy, `MetaAlg` estimates its suboptimality and either terminates successfully, or issues several other calls to `dfslearn` to gather more data sets. This so-called exploration-on-demand scheme is due to Krishnamurthy, Agarwal, and Langford (2016), who describe the subroutine in more detail.

6.4.1 What is new compared to `LSVEE`?

The overall structure of `VALOR` is similar to `LSVEE` (Krishnamurthy, Agarwal, and Langford, 2016). The main differences are in the pruning mechanism, where we use a novel state-identity test, and the policy optimization step in Algorithm 10.

[3] This is not rare in RL; see e.g., Chapter 3.4 of Ross (2013).

LSVEE uses a Q-value function class $\mathcal{F} \subset (\mathcal{X} \times \mathcal{A} \to [0,1])$ and a state identity test based on Bellman errors on data sets D consisting of (x, a, r, x') tuples:

$$\hat{\mathbf{E}}_D \left[\left(f(x,a) - r - \hat{\mathbf{E}}_{x' \sim a} \max_{a' \in \mathcal{A}} f(x', a') \right)^2 \right].$$

This enables a conceptually simpler statistical analysis, but the coupling between value function and the policy yield challenging optimization problems that do not obviously admit efficient solutions.

In contrast, VALOR uses dynamic programming to propagate optimal value estimates from future to earlier time points. From an optimization perspective, we fix the future value and only optimize the current policy, which can be implemented by standard oracles, as we will see. However, from a statistical perspective, the inaccuracy of the future value estimates leads to bias that accumulates over levels. By a careful design of the algorithm and through an intricate and novel analysis, we show that this bias only accumulates linearly, which leads to a polynomial sample complexity guarantee.

6.4.2 Computational and Sample Complexity of VALOR

VALOR requires two types of nontrivial computations over the function classes. We show that they can be reduced to CSC on Π and LP on $\mathcal{G}$ (recall Section 6.3.1), respectively, and hence VALOR is oracle-efficient.

First, Lines 4 in `polvalfun` and 12 in `dfslearn` involve optimizing $V_D(\pi; \{V_a\})$ (Eq. (6.1)) over Π, which can be reduced to CSC as follows: We first form tuples $(x^{(i)}, a^{(i)}, y^{(i)})$ from D and $\{V_a\}$ on which $V_D(\pi; \{V_a\})$ depends, where we bind x_h to $x^{(i)}$, a_h to $a^{(i)}$, and $r_h + V_{a_h}$ to $y^{(i)}$. From the tuples, we construct a CSC data set $(x^{(i)}, -[K\mathbf{1}\{a = a^{(i)}\}y^{(i)}]_{a \in \mathcal{A}})$. On this data set, the cost-sensitive error of any policy (interpreted as a classifier) is exactly $-V_D(\pi; \{V_a\})$, so minimizing error (which the oracle does) maximizes the original objective.

Second, the state identity test requires solving the following problem over the function class $\mathcal{G}$:

$$V_{opt} = \max_{g \in \mathcal{G}} \hat{\mathbf{E}}_{D'}[g(x_h)] \quad \text{(and min for } V_{pes}\text{)}$$
$$\text{s.t. } V - \phi_h \le \hat{\mathbf{E}}_D[g(x_h)] \le V + \phi_h, \forall (D, V) \in \mathcal{D}_h.$$

The objective and the constraints are linear functionals of $\mathcal{G}$, all empirical expectations involve polynomially many samples, and the number of constraints is $|\mathcal{D}_h|$ which remains polynomial throughout the execution of the algorithm, as we will show in the sample complexity analysis. Therefore, the LP oracle can directly handle this optimization problem.

We now formally state the main computational and statistical guarantees for VALOR.

Theorem 82 (Oracle efficiency of VALOR)**.** *Consider a contextual decision process with deterministic dynamics over M hidden states as described in Section 6.3. Assume $\pi^\star \in \Pi$ and $g^\star \in \mathcal{G}$. Then for any $\epsilon, \delta \in (0,1)$, with probability at least $1-\delta$, VALOR makes $O\left(\frac{MH^2}{\epsilon} \log \frac{MH}{\delta}\right)$ CSC oracle calls and at most $O\left(\frac{MKH^2}{\epsilon} \log \frac{MH}{\delta}\right)$ LP oracle calls with required accuracy $\epsilon_{feas} = \epsilon_{sub} = \tilde{O}(\epsilon^2/MH^3)$.*

Theorem 83 (PAC bound of VALOR)**.** *Under the same setting and assumptions as in Theorem 82, VALOR returns a policy $\hat{\pi}$ such that $V^\star - V^{\hat{\pi}} \le \epsilon$ with probability at least $1-\delta$, after collecting at most $\tilde{O}\left(\frac{M^3H^8K}{\epsilon^5} \log(|\mathcal{G}||\Pi|/\delta) \log^3(1/\delta)\right)$ trajectories.*[4]

Note that this bound assumes finite value function and policy classes for simplicity, but can be extended to infinite function classes with bounded statistical complexity using standard tools, as in Section 5.3 of

[4] $\tilde{O}(\cdot)$ suppresses logarithmic dependencies on M, K, H, $1/\epsilon$ and doubly-logarithmic dependencies on $1/\delta$, $|\mathcal{G}|$, and $|\Pi|$.

Jiang, Krishnamurthy, et al. (2017). The resulting bound scales linearly with the Natarajan and Pseudo-dimension of the function classes, which are generalizations of VC-dimension. We further expect that one can generalize the theorems above to an approximate version of realizability as in Section 5.4 of Jiang, Krishnamurthy, et al. (2017).

Compared to the guarantee for LSVEE (Krishnamurthy, Agarwal, and Langford, 2016), Theorem 83 is worse in the dependence on M, H, and ϵ. Yet, in Section 6.8.7 we show that a version of VALOR with alternative oracle assumptions enjoys a better PAC bound than LSVEE. Nevertheless, we emphasize that our main goal is to understand the interplay between statistical and computational efficiency to discover new algorithmic ideas that may lead to practical methods, rather than improve sample complexity bounds.

6.5 Toward Oracle-Efficient PAC-RL with Stochastic Hidden State Dynamics

VALOR demonstrates that provably sample- and oracle-efficient RL with rich stochastic observations is possible and, as such, makes progress toward reliable and practical RL in many applications. In this section, we discuss the natural next step of allowing stochastic hidden-state transitions.

6.5.1 OLIVE is not Oracle-Efficient

For this more general setting with stochastic hidden state dynamics, OLIVE (Jiang, Krishnamurthy, et al., 2017) is the only known algorithm with polynomial sample complexity, but its computational properties remain underexplored. We show here that OLIVE is in fact not oracle-efficient. A brief description of the algorithm is provided below, and in the theorem statement, we refer to a parameter ϕ, which the algorithm uses as a tolerance on deviations of empirical expectations.

Theorem 84. *Assuming $P \neq NP$, even with algorithm parameter $\phi = 0$ and perfect evaluation of expectations, OLIVE is not oracle-efficient, that is, it cannot be implemented with polynomially many basic arithmetic operations and calls to CSC, LP, and LS oracles.*

The assumptions of perfect evaluation of expectations and $\phi = 0$ are merely to unclutter the constructions in the proofs. We show this result by proving that even in tabular MDPs, OLIVE solves an NP-hard problem to determine its next exploration policy, while all oracles we consider have polynomial runtime in the tabular setting. While we only show this for CSC, LP, and LS oracles explicitly, we expect other practically relevant oracles to also be efficient in the tabular setting, and therefore they could not help to implement OLIVE efficiently.

This theorem shows that there are no known oracle-efficient PAC-RL methods for this general setting and that simply applying clever optimization tricks to implement OLIVE is not enough to achieve a practical algorithm. Yet, this result does not preclude tractable PAC RL altogether, and we discuss plausible directions in the subsequent section. Below we highlight the main arguments of the proof.

Proof Sketch of Theorem 84. OLIVE is round-based and follows the *optimism in the face of uncertainty* principle. At round k it selects a value function and a policy to execute $(\hat{g}_k, \hat{\pi}_k)$ that promise the highest return while satisfying all average Bellman error constraints:

$$\hat{g}_k, \hat{\pi}_k = \operatorname*{argmax}_{g \in \mathcal{G}, \pi \in \Pi} \hat{\mathbf{E}}_{D_0}[g(x)] \tag{6.2}$$
$$\text{s.t. } |\hat{\mathbf{E}}_{D_i}[K\mathbf{1}\{a = \pi(x)\}(g(x) - r - g(x'))]| \leq \phi, \ \ \forall\, D_i \in \mathcal{D}.$$

Here D_0 is a data set of initial contexts x, $\mathcal{D}$ consists of data sets of (x, a, r, x') tuples collected in the previous rounds, and ϕ is a statistical tolerance parameter. If this optimistic policy $\hat{\pi}_k$ is close to optimal, OLIVE returns it and terminates. Otherwise we add a constraint to (6.2) by (i) choosing a time point h, (ii) collecting trajectories with $\hat{\pi}_k$ but choosing the h-th action uniformly, and (iii) storing the tuples (x_h, a_h, r_h, x_{h+1}) in the new data set D_k which is added to the constraints for the next round.

The following theorem shows that OLIVE's optimization is NP-hard even in tabular MDPs.

Theorem 85. *Let $\mathcal{P}_{\texttt{OLIVE}}$ denote the family of problems of the form (6.2), parameterized by $(\mathcal{X}, \mathcal{A}, Env, t)$, which describes the optimization problem induced by running OLIVE in the MDP Env (with states $\mathcal{X}$, actions $\mathcal{A}$, and perfect evaluation of expectations) for t rounds. OLIVE is given tabular function classes $\mathcal{G} = (\mathcal{X} \to [0, 1])$ and $\Pi = (\mathcal{X} \to \mathcal{A})$ and uses $\phi = 0$. Then $\mathcal{P}_{\texttt{OLIVE}}$ is NP-hard.*

At the same time, oracles are implementable in polynomial time:

Proposition 86. *For tabular value functions $\mathcal{G} = (\mathcal{X} \to [0, 1])$ and policies $\Pi = (\mathcal{X} \to \mathcal{A})$, the CSC, LP, and LS oracles can be implemented in time polynomial in $|\mathcal{X}|$, $K = |\mathcal{A}|$ and the input size.*

Both proofs are in Section 6.10. Proposition 86 implies that if OLIVE could be implemented with polynomially many CSC/LP/LS oracle calls, its total runtime would be polynomial for tabular MDPs. Assuming P $\neq$ NP, this contradicts Theorem 85 which states that determining the exploration policy of OLIVE in tabular MDPs is NP-hard. Combining both statements therefore proves Theorem 84.

We now give brief intuition for Proposition 86. To implement the CSC oracle, for each of the polynomially many observations $x \in \mathcal{X}$, we simply add the cost vectors for that observation together and pick the action that minimizes the total cost, that is, compute the action $\hat{\pi}(x)$ as $\min_{a \in \mathcal{A}} \sum_{i \in [n]:\, x^{(i)} = x} c^{(i)}(a)$. Similarly, the square-loss objective of the LS-oracle decomposes and we can compute the tabular solution one entry at a time. In both cases, the oracle runtime is $O(nK|\mathcal{X}|)$. Finally, using one-hot encoding, $\mathcal{G}$ can be written as a linear function in $\mathbb{R}^{|\mathcal{X}|}$ for which the LP oracle problem reduces to an LP in $\mathbb{R}^{|\mathcal{X}|}$. The ellipsoid method (Khachiyan, 1980) solves these approximately in polynomial time.

6.5.2 Alternative Algorithms.

An important element of VALOR is that it explicitly stores value estimates of the hidden states, which we call "local values." Local values lead to statistical and computational efficiency under weak realizability conditions, but this approach is unlikely to generalize to the stochastic setting where the agent may not be able to consistently visit a particular hidden state. In Sections 6.8.7-6.9.2, we therefore derive alternative algorithms which do not store local values to approximate the future value $g^\star(x_{h+1})$. Inspired by classical RL algorithms, these algorithms approximate $g^\star(x_{h+1})$ by either bootstrap targets $\hat{g}_{h+1}(x_{h+1})$ (as in TD methods) or Monte-Carlo estimates of the return using a near-optimal roll-out policy $\hat{\pi}_{h+1:H}$ (as in PSDP (Bagnell et al., 2004)). Using such targets can introduce additional errors, and stronger realizability-type assumptions on $\Pi, \mathcal{G}$ are necessary for polynomial sample-complexity (see Section 6.9). Nevertheless, these algorithms are also oracle-efficient and while we only establish statistical efficiency with deterministic hidden state dynamics, we believe that they considerably expand the space of plausible algorithms for the general setting.

6.6 Summary

This paper describes new RL algorithms for environments with rich stochastic observations and deterministic hidden state dynamics. Unlike other existing approaches, these algorithms are computationally efficient in an oracle model, and we emphasize that the oracle-based approach has led to practical algorithms

for many other settings. We believe this work represents an important step toward computationally and statistically efficient RL with rich observations.

While challenging benchmark environments in modern RL (e.g. visual grid-worlds (Johnson et al., 2016)) often have the assumed deterministic hidden state dynamics, the natural goal is to develop efficient algorithms that handle stochastic hidden-state dynamics. We show that the only known approach for this setting is not implementable with standard oracles, and we also provide several constructions demonstrating other concrete challenges of RL with stochastic state dynamics. This provides insights into the key open question of whether we can design an efficient algorithm for the general setting. We hope to resolve this question in future work.

6.7 Additional Notation and Definitions

In the next few sections we analyze the new algorithms for the deterministic setting. We will adopt the following conventions:

- In the deterministic setting (which we focus on here), a path p always deterministically leads to some state s, so we use them interchangeably, e.g., $V^\star(p) \equiv V^\star(s)$, $x \sim p \Leftrightarrow x \sim s$.
- It will be convenient to define $V^\pi(s) := \mathbb{E}[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, a_{h:H} \sim \pi]$ for s at level h, which is the analogy of $V^\star(s)$ for π. Recall that $V^\pi \equiv V^\pi(\varnothing)$ and $V^\star \equiv V^\star(\varnothing)$. Also define $Q^\star(s,\pi) := \mathbb{E}_{x\sim s}[Q^\star(x,\pi(x))]$.
- We use $\hat{\mathbf{E}}_D[\cdot]$ to denote empirical expectation over samples drawn from data set D, and we use $\mathbb{E}_p[\cdot]$ to denote population averages where data is drawn from path p. Often for this latter expectation, we will draw (x,a,r,x') where $x \sim p, a \sim \mathrm{Unif}(\mathcal{A})$ and r, x' are sampled according to the appropriate conditional distributions. In the notation $\mathbb{E}_p$ we default to the uniform action distribution unless otherwise specified.

6.7.1 Additional Oracles

Least-Squares (LS) Oracle The least-squares oracle takes as inputs a parameter ϵ_{sub} and a sequence $\{(x^{(i)}, v^{(i)})\}_{i\in[n]}$ of observations $x^{(i)} \in \mathcal{X}$ and values $v^{(i)} \in \mathbb{R}$. It outputs a value function $\hat{g} \in \mathcal{G}$ whose squared error is ϵ_{sub} close to the least-squares fit

$$\min_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n}(v^{(i)} - g(x^{(i)}))^2.$$

Multi Data Set Classification Oracle The multi data set classification oracle receives as inputs a parameter ϵ_{feas}, m scalars that are upper bounds on the allowed cost $\{U_j\}_{j\in[m]} \in \mathbb{R}^m$, and m cost-sensitive classification data sets $D_1, \dots D_m$, each of which consists of a sequence of observations $\{x_j^{(i)}\}_{i\in[n]} \in \mathcal{X}^n$ and a sequence of cost vectors $\{c_j^{(i)}\}_{i\in[n]} \in \mathbb{R}^{K\times n}$, where $c_j^{(i)}(a)$ is the cost of predicting action $a \in \mathcal{A}$ for $x_j^{(i)}$. The oracle returns a policy that achieves on each data set D_j at most an average cost of $U_j + \epsilon_{\text{feas}}$, if a policy exists in Π that achieves costs at most U_j on each dataset. Formally, the oracle returns a policy in

$$\left\{\pi \in \Pi \;\middle|\; \forall j \in [m] : \; \frac{1}{n}\sum_{i=1}^{n} c_j^{(i)}(\pi(x_j^{(i)})) \le U_j + \epsilon_{\text{feas}}\right\}.$$

This oracle generalizes the CSC oracle by requiring the same policy to achieve low cost on multiple CSC data sets simultaneously. Nonetheless, it can be implemented with a CSC oracle as follows: We associate

a Lagrange parameter with each constraint, and optimize the Lagrange parameters using multiplicative weights. In each iteration, we use the multiplicative weights to combine the m constraints into a single one, and then solve the resulting cost-sensitive problem with the CSC oracle. The slack in the constraint as witnessed by the resulting policy is used as the loss to update the multiplicative weights parameters. See Arora, Hazan, and Kale, 2012 for more details.

6.7.2 Assumptions on the Function Classes

While VALOR only requires realizability of the policy and the value function classes, our other algorithms require stronger assumptions which we introduce below.

Assumption 87 (Policy realizability). $\pi^\star \in \Pi$.

Assumption 88 (Value realizability). $g^\star \in \mathcal{G}$.

Assumption 89 (Policy-value completeness). *At each level h, $\forall g' \in \mathcal{G}_{h+1}$, there exists $\pi^\star_{g'} \in \Pi_h$ such that $\forall x \in \mathcal{X}$,*

$$\pi^\star_{g'}(x) = \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}[r + g'(x_{h+1}) | x_h = x, a_h = a].$$

In addition, $\forall g' \in \mathcal{G}_{h+1}$, $\exists g_{\star,g'} \in \mathcal{G}_h$ s.t. $\forall x \in \mathcal{X}$,

$$g_{\star,g'}(x) = \mathbb{E}[r + g'(x') | x_h = x, a_h = \pi^\star_{g'}(x)].$$

Assumption 90 (Policy completeness). *For every h, and every non-stationary policy $\pi_{h+1:H}$, there exists a policy $\pi \in \Pi_h$ such that, for all $x \in \mathcal{X}_h$, we have*

$$\pi(x) = \operatorname{argmax}_a \mathbb{E}[\textstyle\sum_{h'=h}^{H} r_{h'} | x, a, a_{h+1:H} \sim \pi_{h+1:H}].$$

Fact 91 (Relationship between the assumptions).
Assum.89 $\Rightarrow$ Assum.90 $\Rightarrow$ Assum.87. *Assum.89 $\Rightarrow$ Assum.88.*

In words, these assumptions ask that for any possible approximation of the future value that we might use, the induced square loss or cost-sensitive problems are realizable using $\mathcal{G}, \Pi$, which is a much stronger notion of realizability than Assumptions 87 and 88. Such assumptions are closely related to the conditions needed to analyze Fitted Value/Policy Iteration methods (see e.g., Munos and Szepesvári, 2008; Antos, Szepesvári, and Munos, 2008).

6.8 Analysis of VALOR

Definition 92. *A state $s \in \mathcal{S}_h$ is called* learned *if there is a data set in $\mathcal{D}_h$ that is sampled from a path leading to that state. The set of all learned states at level h is $\mathcal{S}_h^{learned}$ and $\mathcal{S}^{learned} := \bigcup_{h \in [H]} \mathcal{S}_h^{learned}$.*

6.8.1 Concentration Results

We now define an event $\mathcal{E}$ that holds with high probability and will be the main concentration argument in the proof. This event uses a parameter ϵ_{stat} whose value we will set later.

Definition 93 (Deviation Bounds). *Let $\mathcal{E}$ denote the event that for all $h \in [H]$ the total number of calls to* dfslearn(p) *at level h is at most $T_{\max} = MHn_{exp} + M$ during the execution of* MetaAlg *and that for all these calls to* dfslearn(p) *the following deviation bounds hold for all $g \in \mathcal{G}_h$ and $\pi \in \Pi_h$ (where*

$$\epsilon_{\text{stat}} = \epsilon_{\text{sub}} = \epsilon_{\text{feas}} = \frac{\epsilon}{2^6 7 H^2 T_{\max}}$$

$$\phi_h = (H - h + 1)(6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}})$$

$$T_{\max} = MHn_{\text{exp}} + M$$

$$n_{\text{test}} = \frac{\log\left(12KHT_{\max}|\mathcal{G}|/\delta\right)}{2\epsilon_{\text{stat}}^2},$$

$$n_{\text{train}} = \frac{16K\log(12HT_{\max}|\mathcal{G}||\Pi|/\delta)}{\epsilon_{\text{stat}}^2},$$

$$n_{\text{exp}} = \frac{8\log(4MH/\delta)}{\epsilon},$$

$$n_{\text{eval}} = \frac{32\log(8MH/\delta)}{\epsilon^2}$$

Table 6.1: Exact values of parameters of VALOR run with inputs $\epsilon, \delta \in (0,1)$ and $M, K \in \mathbb{N}$.

D'_a is a data set of n_{test} observations sampled from $p \circ a$ in Line 5, and $\tilde{D}$ is the data set of n_{train} samples from Line 11 with stored values $\{V_a\}_{a\in\mathcal{A}}$):

$$\left|\hat{\mathbb{E}}_{D'_a}[g(x_{h+1})] - \mathbb{E}_{p\circ a}[g(x_{h+1})]\right| \le \epsilon_{stat}, \quad \forall a \in \mathcal{A} \tag{6.3}$$

$$\left|\hat{\mathbb{E}}_{\tilde{D}}[g(x_h)] - \mathbb{E}_{p}[g(x_h)]\right| \le \epsilon_{stat}$$

$$\left|\hat{\mathbb{E}}_{\tilde{D}}[K\mathbf{1}\{\pi(x_h) = a_h\}(r_h + V_a)] - \mathbb{E}_{p}[K\mathbf{1}\{\pi(x_h) = a_h\}(r_h + V_a)]\right| \le \epsilon_{stat}. \tag{6.4}$$

In the next Lemma, we bound $\mathbb{P}[\mathcal{E}]$, which is the main concentration argument in the proof. The bound involves a new quantity $T_{\max}$ which is the maximum number of calls to dfslearn. We will control this quantity later.

Lemma 94. *Set*

$$n_{test} \ge \frac{1}{2\epsilon_{stat}^2}\ln\left(\frac{12KHT_{\max}|\mathcal{G}|}{\delta}\right), \qquad n_{train} \ge \frac{16K}{\epsilon_{stat}^2}\ln\left(\frac{12HT_{\max}|\mathcal{G}||\Pi|}{\delta}\right).$$

Then $\mathbb{P}[\mathcal{E}] \ge 1 - \delta/2$.

Proof. Let us denote the total number of calls to dfslearn before the algorithm stops by N_{dfs} (which is random) and first focus on the j-th call to dfslearn. Let $\mathcal{B}_j$ be the sigma-field of all samples collected before the jth call to dfslearn (if it exists, or otherwise the last call to dfslearn) and all intrinsic randomness of the algorithm. The current path is denoted by p_j at level h_j and data sets D'_a, $\tilde{D}$ collected are denoted by $D'_{j,a}$ and $\tilde{D}_j$ respectively. Consider a fix $a \in \mathcal{A}$ and $g \in \mathcal{G}$ and define

$$Y_{i,j} = \begin{cases} 0 & \text{if } j > N_{dfs} \\ g(x_{h+1}^{(i,j)}) - \mathbb{E}_{p_j\circ a}[g(x_{h+1})] & \text{otherwise} \end{cases}$$

which is well-defined even if $j > N_{dfs}$ and where $x_{h+1}^{(i,j)}$ is the i-th sample of x_{h+1} in $D'_{j,a}$. Since $|Y_{i,j}| \leq 1$ and since contexts x_{h+1} are sampled i.i.d. from $p_j \circ a$ conditioned on p_j which is measurable in $\mathcal{B}_j$, we get by Hoeffding's lemma that $\mathbb{E}[\exp(\lambda Y_{i,j})|Y_{1:i-1,j}, \mathcal{B}_j] = \mathbb{E}[\exp(\lambda Y_{i,j})|\mathcal{B}_j] \leq \exp(\lambda^2/2)$ for $\lambda \in \mathbb{R}$. As a result, we have $\mathbb{E}[\exp(\lambda \sum_{i=1}^{n_{\text{test}}} Y_{i,j})] = \mathbb{E}[\mathbb{E}[\exp(\lambda \sum_{i=1}^{n_{\text{test}}} Y_{i,j})|\mathcal{B}_j]] \leq \exp(n_{\text{test}}\lambda^2/2)$ and by Chernoff's bound the following concentration result holds

$$\left|\hat{\mathbb{E}}_{D'_{j,a}}[g(x_{h+1})] - \mathbb{E}_{p_j \circ a}[g(x_{h+1})]\right| \leq \sqrt{\frac{\log(2K|\mathcal{G}|/\delta')}{2n_{\text{test}}}}$$

with probability at least $1 - \frac{\delta'}{K|\mathcal{G}|}$ for a fixed a and g and j as long as $j \leq N_{dfs}$. With a union bound over $\mathcal{A}$ and $\mathcal{G}$, the following statement holds: Given a fix $j \in \mathbb{N}$, with probability at least $1 - \delta'$, if $j \leq N_{dfs}$ then for all $g \in \mathcal{G}_{h+1}$ and $a \in \mathcal{A}$

$$\left|\hat{\mathbb{E}}_{D'_{j,a}}[g(x_{h+1})] - \mathbb{E}_{p_j \circ a}[g(x_{h+1})]\right| \leq \sqrt{\frac{\log(2K|\mathcal{G}|/\delta')}{2n_{\text{test}}}}.$$

Choosing $n_{\text{test}} \geq \frac{1}{2\epsilon_{\text{stat}}^2} \ln\left(\frac{12KHT_{\max}|\mathcal{G}|}{\delta}\right)$ and $\delta' = \frac{\delta}{6HT_{\max}}$ allows us to bound the LHS by ϵ_{stat}. In exactly the same way since the data set $\tilde{D}_j$ consists of n_{train} samples that, given $\mathcal{B}_j$, are sampled i.i.d. from p_j, we have for all $g \in \mathcal{G}_h$

$$\left|\hat{\mathbb{E}}_{\tilde{D}_j}[g(x_h)] - \mathbb{E}_{p_j}[g(x_h)]\right| \leq \sqrt{\frac{\log(2|\mathcal{G}|/\delta')}{2n_{\text{train}}}},$$

with probability $1 - \delta'$ as long as $j \leq N_{dfs}$. As above, our choice of n_{train} ensures that this deviation is bound by ϵ_{stat}.

Finally, for the third inequality we must use Bernstein's inequality. For the random variable $K\mathbf{1}\{\pi(x_h) = a_h\}(r_h + V_{a_h})$, since a_h is chosen uniformly at random, it is not hard to see that both the variance and the range are at most $2K$ (see for example Lemma 14 by Jiang, Krishnamurthy, et al. (2017)). As such, Bernstein's inequality with a union bound over $\pi \in \Pi$ gives that with probability $1 - \delta'$,

$$\left|(\hat{\mathbb{E}}_{\tilde{D}_j} - \mathbb{E}_{p_j})[K\mathbf{1}\{\pi(x_h) = a_h\}(r_h + V_{a_h})]\right| \leq \sqrt{\frac{4K\log(2|\Pi|/\delta')}{n_{\text{train}}}} + \frac{4K}{3n_{\text{train}}}\log(2|\Pi|/\delta') \leq \epsilon_{\text{stat}},$$

since $\{V_a\}$ and p_j can essentially be considered fixed at the time when $\tilde{D}_j$ is collected (a more formal treatment is analogous to the proof of the first two inequalities). Using a union bound, the deviation bounds (6.3)–(6.4) hold for a single call to `dfslearn` with probability $1 - 3\delta'$.

Consider now the event $\mathcal{E}'$ that these bounds hold for the first $T_{\max}$ calls at each level h. Applying a union bound let us bound $\mathbb{P}(\mathcal{E}') \geq 1 - 3HT_{\max}\delta' = 1 - \frac{\delta}{2}$. It remains to show that $\mathcal{E}' \subseteq \mathcal{E}$.

First note that in event $\mathcal{E}'$ in the first $T_{\max}$ calls to `dfslearn`, the algorithm does not call itself recursively if $p \circ a$ leads to a learned state. To see this assume $p \circ a$ leads to a state $s \in \mathcal{S}^{\text{learned}}$. Let D'_a be the data set collected in Line 5 for this action a. Since the subsequent state $s \in \mathcal{S}^{\text{learned}}$, then there is a data set $(D, V, \{V_b\}) \in \mathcal{D}_{h+1}$ sampled from this state (we will only use the first two items in the tuple). This means that D'_a and D are two data sets sampled from the same distribution, and as such, we have

$$\begin{aligned}
V_{opt} - V_{pes} &= \hat{\mathbf{E}}_{D'_a}[g_{opt}(x_{h+1}) - g_{pes}(x_{h+1})] \leq \mathbb{E}_s[g_{opt}(x_{h+1}) - g_{pes}(x_{h+1})] + 2\epsilon_{\text{stat}} \\
&\leq \hat{\mathbf{E}}_D[g_{opt}(x_{h+1}) - g_{pes}(x_{h+1})] + 4\epsilon_{\text{stat}} \\
&\leq V + \phi_{h+1} + \epsilon_{\text{feas}} - V + \phi_{h+1} + \epsilon_{\text{feas}} + 4\epsilon_{\text{stat}} = 2\phi_{h+1} + 4\epsilon_{\text{stat}} + 2\epsilon_{\text{feas}}.
\end{aligned}$$

The last line holds because the constraints for g_{opt} and g_{pes} include the one based on (D, V) (Line 6), so the expectation of g_{opt} and g_{pes} on D can only differ by the amount of the allowed slackness $2\phi_{h+1}$ and the violations of feasibility $2\epsilon_{\text{feas}}$. Therefore the condition in the if clause is satisfied and the algorithm does not call itself recursively. We here assumed that the constrained optimization problem has an approximately feasible solution but if that is not the case, the if condition is trivially satisfied.

Since the number of learned states per level is bounded by M, this means that within the first $T_{\max}$ calls to `dfslearn`, the algorithm can make recursive calls to the level below at most M times. Further note that for any fixed level h the total number of non-recursive calls to `dfslearn` is bounded by MHn_{exp} since `MetaAlg` has at most MH iterations and in each `dfslearn` is called n_{exp} times at each level (but the first). Therefore, in event $\mathcal{E}'$, the total number of calls to `dfslearn` at any level h is bounded by $MHn_{\text{exp}} + M \leq T_{\max}$ and the statement follows. □

6.8.2 Bound on Oracle Calls

Proof of Theorem 82. Consider event $\mathcal{E}$ from Definition 93 which by Lemma 94 has probability at least $1 - \delta/2$. `VALOR` requires two types of nontrivial computations over the function classes. We show that they can be reduced to CSC on Π and LP on $\mathcal{G}$ (recall Sec. 6.3.1), respectively, and hence `VALOR` is oracle-efficient.

First, Line 12 in `dfslearn` involves optimizing $V_D(\pi; \{V_a\})$ (Eq. (6.1)) over Π, which can be reduced to CSC as follows: We first form tuples $(x^{(i)}, a^{(i)}, y^{(i)})$ from D and $\{V_a\}$ on which $V_D(\pi; \{V_a\})$ depends, where we bind x_h to $x^{(i)}$, a_h to $a^{(i)}$, and $r_h + V_{a_h}$ to $y^{(i)}$. From the tuples, we construct a CSC data set $(x^{(i)}, -[K\mathbf{1}\{a = a^{(i)}\}y^{(i)}]_{a \in \mathcal{A}})$, where the second argument is a K-dimensional vector with one non-zero. On this data set, the cost-sensitive risk of any policy (interpreted as a classifier) is exactly $-V_D(\pi; \{V_a\})$, so minimizing risk (which the oracle does) maximizes the original objective.[5]

Second, the optimization in Line 4 in `polvalfun` can be reduced to CSC with the very same argument, except that we now accumulate all CSC inputs for each data set in $\mathcal{D}_h$. Since $|\mathcal{D}_h| \leq T_{\max}$ is polynomial, the total input size is still polynomial.

Third, the state identity test in Line 6 in `dfslearn` requires solving the following problem over the function class $\mathcal{G}$:

$$V_{opt} = \max_{g \in \mathcal{G}} \hat{\mathbf{E}}_{D'}[g(x_h)] \quad \text{(and min for } V_{pes}\text{)}$$
$$\text{s.t. } V - \phi_h \leq \hat{\mathbf{E}}_D[g(x_h)] \leq V + \phi_h, \forall (D, V) \in \mathcal{D}_h.$$

The objective and the constraints are linear functionals of $\mathcal{G}$, all empirical expectations involve polynomially many samples, and the number of constraints is $|\mathcal{D}_h| \leq T_{\max}$ which remains polynomial throughout the execution of the algorithm. Therefore, the LP oracle can directly handle this optimization problem.

Altogether, we showed that all non-trivial computations can be reduced to oracle calls with inputs with polynomial description length. It remains to show that the number of calls is bounded. Since there are at most $T_{\max}$ calls to `dfslearn` at each level $h \in [H]$, the total number of calls to the LP oracle is $T_{\max}HK$. Similarly, the number of CSC oracle calls from `dfslearn` is at most $T_{\max}H$. In addition, there at at most MH calls to the CSC oracle in `polvalfun`. The statement follows with realizing that $T_{\max} = MHn_{\text{exp}} + M = O\left(\frac{MH}{\epsilon} \ln\left(\frac{MH}{\delta}\right)\right)$. □

[5] Note that the inputs to the oracle have polynomial length: D consists of polynomially many (x, a, r, x') tuples, each of which should be assumed to have polynomial description length, and $\{V_a\}$ similarly.

6.8.3 Depth First Search and Estimated Values

In this section, we show that in the high-probability event $\mathcal{E}$ (Definition 93), `dfslearn` produces good estimates of optimal values on learned states. The next lemma first quantifies the error in the value estimate at level h in terms of the estimation error of the values of the next time step $\{V_a\}_a$.

Lemma 95 (Error propagation when learning a state)**.** *Consider a call to* `dfslearn` *with input path p of depth h. Assume that all values $\{V_a\}_{a\in\mathcal{A}}$ in Algorithm 11 satisfy $|V_a - V^\star(p\circ a)| \le \beta$ for some $\beta > 0$. Then in event $\mathcal{E}$, $\tilde{V}$ returned in Line 14 satisfies $|\tilde{V} - V^\star(p)| \le \epsilon_{stat} + \beta + \epsilon_{sub}$.*

Proof. The proof follows a standard analysis of empirical risk minimization (here we are maximizing). Let $\tilde{\pi}$ denote the empirical risk maximizer in Line 12 and let $\pi^\star$ denote the globally optimal policy (which is in our class due to realizability). Then

$$\begin{aligned}\tilde{V} &\le \hat{\mathbf{E}}_{\tilde{D}}[K\mathbf{1}\{\tilde{\pi}(x_h) = a_h\}(r_h + V_{a_h})] \le \mathbb{E}_p[K\mathbf{1}\{\tilde{\pi}(x_h) = a_h\}(r_h + V_{a_h})] + \epsilon_{\text{stat}}\\ &\le \mathbb{E}_p[K\mathbf{1}\{\tilde{\pi}(x_h) = a_h\}(r_h + g^\star(x_{h+1}))] + \beta + \epsilon_{\text{stat}}\\ &\le \mathbb{E}_p[K\mathbf{1}\{\pi^\star(x_h) = a_h\}(r_h + g^\star(x_{h+1}))] + \beta + \epsilon_{\text{stat}} = V^\star(s) + \beta + \epsilon_{\text{stat}}.\end{aligned}$$

The first inequality is the deviation bound, which holds in event $\mathcal{E}$. The second inequality is based on the precondition on $\{V_a\}_{a\in\mathcal{A}}$, linearity of expectation, and the realizability property of $g^\star_{h+1}$. The third inequality uses that $\pi^\star$ is the global and point-wise maximizer of the long-term expected reward, which is precisely $r_h + g^\star$.

Similarly, we can lower bound $\tilde{V}$ by

$$\begin{aligned}\tilde{V} &= \hat{\mathbf{E}}_{\tilde{D}}[K\mathbf{1}\{\tilde{\pi}(x_h) = a_h\}(r_h + V_{a_h})] - \epsilon_{\text{sub}} \ge \hat{\mathbf{E}}_{\tilde{D}}[K\mathbf{1}\{\pi^\star(x_h) = a_h\}(r_h + V_{a_h})] - \epsilon_{\text{sub}}\\ &\ge \mathbb{E}_p[K\mathbf{1}\{\pi^\star(x_h) = a_h\}(r_h + V_{a_h})] - \epsilon_{\text{stat}} - \epsilon_{\text{sub}}\\ &\ge \mathbb{E}_p[K\mathbf{1}\{\pi^\star(x_h) = a_h\}(r_h + g^\star(x_{h+1}))] - \epsilon_{\text{stat}} - \beta - \epsilon_{\text{sub}} = V^\star(s) - \epsilon_{\text{stat}} - \beta - \epsilon_{\text{sub}}.\end{aligned}$$

Here we first use $\tilde{V}$ is optimal up to ϵ_{sub} and then that $\tilde{\pi}$ is the empirical maximizer. Subsequently, we leveraged the deviation bounds of event $\mathcal{E}$ and finally used the assumption about the estimation accuracy from the level below. This proves the claim. □

The goal of the proof is to apply the above lemma inductively so that we can learn all of the values to reasonable accuracy. Before doing so, we need to quantify the estimation error when V_a is set in Line 8 of the algorithm without a recursive call.

Lemma 96 (Error when not recursing)**.** *Consider a call to* `dfslearn` *with input path p of depth h. If $g^\star$ is feasible for Line 6 of* `dfslearn` *and V_a is set in Line 8 of Algorithm 11, then in event $\mathcal{E}$, the value $V_a = \frac{V_{opt}+V_{pes}}{2}$ satisfies $|V_a - V^\star(p\circ a)| \le \phi_{h+1} + 3\epsilon_{stat} + \epsilon_{feas} + \epsilon_{sub}$.*

Proof. Recall that D'_a is the data set sampled in Line 5 for the particular action a in consideration. Since $g^\star_{h+1}$ is feasible for both V_{opt} and V_{pes}, we have

$$V_{pes} - \epsilon_{\text{sub}} = \hat{\mathbf{E}}_{D'_a}[g_{pes}(x_{h+1})] - \epsilon_{\text{sub}} \le \hat{\mathbf{E}}_{D'_a}[g^\star(x_{h+1})] \le \hat{\mathbf{E}}_{D'_a}[g_{opt}(x_{h+1})] + \epsilon_{\text{sub}} = V_{opt} + \epsilon_{\text{sub}}.$$

Without loss of generality, we can assume that $V_{pes} \le V_{opt}$, otherwise we can just exchange them. This implies that $0 \le V_{opt} - V_a = V_a - V_{pes} = \frac{V_{opt}-V_{pes}}{2} \le \phi_{h+1} + 2\epsilon_{\text{stat}} + \epsilon_{\text{feas}}$. Therefore,

$$\begin{aligned}\hat{\mathbf{E}}_{D'}[g^\star(x_{h+1})] - V_a &\le V_{opt} - V_a + \epsilon_{\text{sub}} = \frac{V_{opt} - V_{pes}}{2} + \epsilon_{\text{sub}} \le \phi_{h+1} + 2\epsilon_{\text{stat}} + \epsilon_{\text{feas}} + \epsilon_{\text{sub}}.\\ V_a - \hat{\mathbf{E}}_{D'}[g^\star(x_{h+1})] &\le V_a - V_{pes} + \epsilon_{\text{sub}} = \frac{V_{opt} - V_{pes}}{2} + \epsilon_{\text{sub}} \le \phi_{h+1} + 2\epsilon_{\text{stat}} + \epsilon_{\text{feas}} + \epsilon_{\text{sub}}.\end{aligned}$$

By the triangle inequality

$$\begin{aligned}|V_a - V^\star(p \circ a)| \leq & |\hat{\mathbf{E}}_{D'_a}[g^\star(x_{h+1})] - V_a| + |\hat{\mathbf{E}}_{D'_a}[g^\star(x_{h+1})] - V^\star(p \circ a)| \\ \leq & \phi_{h+1} + 3\epsilon_{\text{stat}} + \epsilon_{\text{feas}} + \epsilon_{\text{sub}}.\end{aligned}$$

The last inequality is the concentration statement, which holds in event $\mathcal{E}$. □

We now are able to apply Lemma 95 inductively in combination with Lemma 96 to obtain the main result of `dfslearn` in this section.

Proposition 97 (Accuracy of learned values)**.** *Assume the realizability condition $g^\star \in \mathcal{G}_h$. Set $\phi_h = (H+1-h)(6\epsilon_{stat} + 2\epsilon_{sub} + \epsilon_{feas})$ for all $h \in [H]$. Then under event $\mathcal{E}$, for any level $h \in [H]$ and any state $s \in \mathcal{S}_h$ all triplets $(D, V, \{V_a\}) \in \mathcal{D}_h$ associated with state s (formally with paths p that lead to s) satisfy*

$$|V - V^\star(s)| \leq \phi_h - 2\epsilon_{stat}, \qquad |V_a - V^\star(s \circ a)| \leq \phi_{h+1} + 3\epsilon_{stat} + \epsilon_{feas} + \epsilon_{sub}.$$

Moreover, under event $\mathcal{E}$, we have $g^\star$ is feasible for Line 6 of `dfslearn` for all h, at all times.

Proof. We prove this statement by induction over h. For $h = H+1$ the statement holds trivially since $\mathcal{G}_{H+1} = \{g^\star_{h+1}\}$ the constant 0 function is the only function in $\mathcal{G}_{H+1}$ and therefore the algorithm always returns on Line 8 and never calls level $H+1$ recursively.

Consider now some data set $(\tilde{D}, \tilde{V}, \{V_a\}) \in \mathcal{D}_h$ at level h associated with state $s \in \mathcal{S}_h$. This data set was obtained by calling `dfslearn` at some path p (pointing to state s). Since when we added this data set, we have not yet exhausted the budget of $T_{\max}$ calls to `dfslearn` (by the preconditions of the lemma), we have that the once we reach Line 11 the inductive hypothesis applies for all data sets at level $h+1$ (which may have been added by recursive calls of this execution). Each of the V_a values can be set in one of two ways.

1. The algorithm did not make a recursive call. Since by the inductive assumption $g^\star$ is feasible for Line 6 of `dfslearn`, we can apply Lemma 96 and get that

$$|V_a - V^\star(s \circ a)| \leq \phi_{h+1} + 3\epsilon_{\text{stat}} + \epsilon_{\text{feas}} + \epsilon_{\text{sub}}.$$

2. The algorithm made a recursive call. Since the value returned was added as a data set at level $h+1$, it satisfies the inductive assumption

$$|V_a - V^\star(s \circ a)| \leq \phi_{h+1} - 2\epsilon_{\text{stat}}.$$

This demonstrates the second inequality in the inductive step. For the first, applying Lemma 95 with $\beta = \phi_{h+1} + 3\epsilon_{\text{stat}} + \epsilon_{\text{feas}} + \epsilon_{\text{sub}}$, we get that $|\tilde{V} - V^\star(s)| \leq \phi_{h+1} + 4\epsilon_{\text{stat}} + \epsilon_{\text{feas}} + 2\epsilon_{\text{sub}} = \phi_h - 2\epsilon_{\text{stat}}$, by definition of ϕ_h. Finally, this also implies that $|\tilde{V} - \hat{\mathbf{E}}_{\tilde{D}}[g^\star_h(x_h)]| \leq |\tilde{V} - V^\star(s)| + |\hat{\mathbf{E}}_{\tilde{D}}[g^\star_h(x_h)] - V^\star(s)| \leq \phi_h$ which means that $g^\star$ is still feasible. □

6.8.4 Policy Performance

In this section, we bound the quality of the policy returned by `polvalfun` in the good event $\mathcal{E}$ by using the fact that `dfslearn` produces accurate estimates of the optimal values (previous section). Before we state the main result of this section in Proposition 99, we prove the following helpful lemma. This Lemma is essentially Lemma 4.3 in Ross and Bagnell (2014).

Lemma 98. *The suboptimality of a policy π can be written as*

$$V^\star - V^\pi = \mathbb{E}\left[\sum_{h=1}^{H}(V^\star(s_h) - Q^\star(s_h, \pi_h)) \mid a_h \sim \pi_h\right].$$

Proof. The difference of values of a policy π compared to the optimal policy in a certain state $s \in \mathcal{S}_h$ can be expressed as

$$\begin{aligned}
V^\star(s) - V^\pi(s) &= V^\star(s) - \mathbb{E}_s[K\mathbf{1}\{\pi_h(x_h) = a_h\}(r_h + V^\pi(x_{h+1})] \\
&= V^\star(s) - \mathbb{E}_s[K\mathbf{1}\{\pi_h(x_h) = a_h\}(r_h + V^\star(x_{h+1}) - V^\star(x_{h+1}) + V^\pi(x_{h+1}))] \\
&= V^\star(s) - Q^\star(s, \pi_h) + \mathbb{E}_s[K\mathbf{1}\{\pi_h(x_h) = a_h\}(V^\star(x_{h+1}) - V^\pi(x_{h+1}))] \\
&= V^\star(s) - Q^\star(s, \pi_h) + \mathbb{E}_s[V^\star(x_{h+1}) - V^\pi(x_{h+1}) \mid a_h \sim \pi_h].
\end{aligned}$$

Therefore, by applying this equality recursively, the suboptimality of π can be written as

$$V^\star(s) - V^\pi = \mathbb{E}\left[\sum_{h=1}^{H}(V^\star(s_h) - Q^\star(s_h, \pi_h)) \mid a_h \sim \hat{\pi}_h\right]. \qquad \square$$

Now we may bound the policy suboptimality.

Proposition 99. *Assume $g_h^\star \in \mathcal{G}_h$ and the we are in event $\mathcal{E}$. Recall the definition $\phi_h = (H+1-h)(6\epsilon_{stat} + 2\epsilon_{sub} + \epsilon_{feas})$ for all $h \in [H]$. Then the policy $\hat{\pi} = \hat{\pi}_{1:H}$ returned by* `polvalfun` *satisfies*

$$V^{\hat{\pi}} \geq V^\star - p_{ul}^{\hat{\pi}} - 2H^2 T_{\max}(7\epsilon_{stat} + 3\epsilon_{sub} + 2\epsilon_{feas})$$

where $p_{ul}^{\hat{\pi}} = \mathbb{P}(\exists h \in [H] \ : \ s_h \notin \mathcal{S}^{learned} \mid a_{1:H} \sim \hat{\pi})$ is the probability of hitting an unlearned state when following $\hat{\pi}$.

Proof. To bound the suboptimality of the learned policy, we bound the difference of how much following $\hat{\pi}_h$ for one time step can hurt per state using Proposition 97. For a state $s \in \mathcal{S}^{\text{learned}}$ at level h, we have

$$\begin{aligned}
&V^\star(s) - Q^\star(s, \hat{\pi}_h) \\
&= \mathbb{E}_s[K(\mathbf{1}\{\pi_h^\star(x_h) = a_h\} - \mathbf{1}\{\hat{\pi}_h(x_h) = a_h\})(r_h + g_{h+1}^\star(x_{h+1})] \\
&\leq \sum_{s \in \mathcal{S}_h^{\text{learned}}} \mathbb{E}_s[K(\mathbf{1}\{\pi_h^\star(x_h) = a_h\} - \mathbf{1}\{\hat{\pi}_h(x_h) = a_h\})(r_h + g_{h+1}^\star(x_{h+1})] \\
&\leq \sum_{(s,_,\{V_a\}) \in \mathcal{D}_h} \Big(\mathbb{E}_s[K(\mathbf{1}\{\pi_h^\star(x_h) = a_h\} - \mathbf{1}\{\hat{\pi}_h(x_h) = a_h\})(r_h + V_{a_h})] \\
&\qquad + 2\phi_{h+1} + 6\epsilon_{\text{stat}} + 2\epsilon_{\text{feas}} + 2\epsilon_{\text{sub}}\Big) \\
&\leq \sum_{(\tilde{D},_,\{V_a\}) \in \mathcal{D}_h} \Big(\mathbb{E}_{\tilde{D}}[K(\mathbf{1}\{\pi_h^\star(x_h) = a_h\} - \mathbf{1}\{\hat{\pi}_h(x_h) = a_h\})(r_h + V_{a_h})] \\
&\qquad + 2\phi_{h+1} + 8\epsilon_{\text{stat}} + 2\epsilon_{\text{feas}} + 2\epsilon_{\text{sub}}\Big) \\
&\leq 2|\mathcal{D}_h|(\phi_{h+1} + 4\epsilon_{\text{stat}} + \epsilon_{\text{feas}} + 2\epsilon_{\text{sub}}).
\end{aligned}$$

Here the first identity is based on expanding definitions. For the first inequality, we use that $s \in \mathcal{S}^{\text{learned}}$ and also that $\pi^\star$ simultaneously maximizes the long term reward from all states, so the terms we added in

are all non-negative. In the second inequality, we introduce the notation $(s, \_, \{V_a\}) \in \mathcal{D}_h$ to denote a data set in $\mathcal{D}_h$ associated with state s with successor values $\{V_a\}$. For this inequality we use Proposition 97 to control the deviation of the successor values. The third inequality uses the deviation bound that holds in event $\mathcal{E}$.

Since per `dfslearn` call, only one data set can be added to $\mathcal{D}_h$, the magnitude $|\mathcal{D}_h| \leq T_{\max}$ is bounded by the total number of calls to `dfslearn` at each level. Using Lemma 98, the suboptimality of $\hat{\pi}$ is therefore at most

$$\begin{aligned}
V^\star - V^{\hat{\pi}} &\leq p_{ul}^{\hat{\pi}} + (1 - p_{ul}^{\hat{\pi}}) \sum_{h=1}^{H} 2|\mathcal{D}_h|(\phi_{h+1} + 4\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}) \\
&\leq p_{ul}^{\hat{\pi}} + 2HT_{\max}(4\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}) + 2T_{\max} \sum_{h=1}^{H} \phi_{h+1} \\
&\leq p_{ul}^{\hat{\pi}} + 2HT_{\max}(4\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}) + 2T_{\max}(6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}) \sum_{h=1}^{H} (H - h) \\
&\leq p_{ul}^{\hat{\pi}} + 2HT_{\max}(4\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}) + H^2 T_{\max}(6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}) \\
&\leq p_{ul}^{\hat{\pi}} + 14H^2 T_{\max}\epsilon_{\text{stat}} + 6H^2 T_{\max}\epsilon_{\text{sub}} + 3H^2 T_{\max}\epsilon_{\text{feas}}.
\end{aligned}$$

This argument is similar to the proof of Lemma 8 in Krishnamurthy, Agarwal, and Langford (2016). Note that we introduce the dependency on $T_{\max}$ since we perform joint policy optimization, which will degrade the sample complexity. □

6.8.5 Meta-Algorithm Analysis

Now that we have the main guarantees for `dfslearn` and `polvalfun`, we may turn to the analysis of `MetaAlg`.

Lemma 100. *Consider running* `MetaAlg` *with* `dfslearn` *and* `polvalfun` *(Algorithm 9 + 10 + 11) with parameters*

$$n_{exp} \geq \frac{8}{\epsilon} \ln\left(\frac{4MH}{\delta}\right), \quad n_{eval} \geq \frac{32}{\epsilon^2} \ln\left(\frac{8MH}{\delta}\right), \quad \epsilon_{stat} = \epsilon_{sub} = \epsilon_{feas} = \frac{\epsilon}{2^6 7 H^2 T_{\max}}$$

Then with probability at least $1 - \delta$, `MetaAlg` *returns a policy that is at least ϵ-optimal after at most MK iterations.*

Proof. First apply Lemma 94 so that the good event $\mathcal{E}$ holds, except with probability $\delta/2$.

In the event $\mathcal{E}$, since before the first execution of `polvalfun`, we called `dfslearn`$(\varnothing)$, by Proposition 97, we know that $|\hat{V}^\star - V^\star| \leq \phi_1 - 2\epsilon_{\text{stat}}$ where $\hat{V}^\star$ is the value stored in the only dataset associated with the root. This value does not change for the remainder of the algorithm, and the choice of $\epsilon_{\text{stat}}, \phi$ ensure that

$$|\hat{V}^\star - V^\star| \leq \phi_1 - 2\epsilon_{\text{stat}} = 6H\epsilon_{\text{stat}} + 2H\epsilon_{\text{sub}} + H\epsilon_{\text{feas}} - 2\epsilon_{\text{stat}} \leq \epsilon/8.$$

This is true for all executions of `polvalfun` (formally all $\hat{V}^{(k)}$ values). Next, since we perform at most MH iterations of the loop in `MetaAlg`, we consider at most MH policies. Via a standard application of Hoeffding's inequality, with probability $1 - \delta/4$, we have that for all $k \in [MH]$

$$|\hat{V}^{\hat{\pi}_k} - V^{\hat{\pi}_k}| \leq \sqrt{\frac{\log(8MH/\delta)}{2n_{\text{eval}}}}.$$

The choice of n_{eval} ensure that this is at most $\epsilon/8$. With these two bounds, if `MetaAlg` terminates, the termination condition implies that

$$V^\star - V^{\hat{\pi}^{(k)}} \leq \hat{V}^{(k)} - \hat{V}^{\hat{\pi}^{(k)}} + \frac{\epsilon}{4} \leq \frac{3}{4}\epsilon \leq \epsilon$$

and hence the returned policy is ϵ-optimal.

On the other hand, if the algorithm does not terminate in iteration k, we have that $\hat{V}^{(k)} - \hat{V}^{\hat{\pi}^{(k)}} > \frac{\epsilon}{2}$ and therefore

$$V^\star - V^{\hat{\pi}^{(k)}} \geq \hat{V}^{(k)} - \hat{V}^{\hat{\pi}^{(k)}} - \frac{\epsilon}{4} \geq \frac{\epsilon}{4}.$$

We now use this fact with Proposition 99 to argue that the policy $\hat{\pi}^{(k)}$ must visit an unlearned state with sufficient probability. Under the conditions here, applying Proposition 99, we get that

$$\frac{\epsilon}{4} \leq V^\star - V^{\hat{\pi}^{(k)}} \leq p_{ul}^{\hat{\pi}^{(k)}} + 2T_{\max}H^2(7\epsilon_{\text{stat}} + 3\epsilon_{\text{sub}} + 2\epsilon_{\text{feas}}).$$

With the choice of ϵ_{stat}, rearranging this inequality reveals that $p_{ul}^{\hat{\pi}^{(k)}} \geq \epsilon/8 > 0$. Hence, if the algorithm does not terminate there must be at least one unlearned state, i.e., $\mathcal{S} \setminus \mathcal{S}^{\text{learned}} \neq \emptyset$.

For the last step of the proof, we argue that since $p_{ul}^{\hat{\pi}^{(k)}}$ is large, the probability of reaching an unlearned state is high, and therefore the additional calls to `dfslearn` in Line 11 with high probability will visit a new state, which we will then learn. Specifically, we will prove that on every non-terminal iteration of `MetaAlg`, we learn at least one previously unlearned state. With this fact, since there are at most MH states, the algorithm must terminate and return a near-optimal policy after at most MH iterations.

In a non-terminal iteration k, the probability that we do not hit an unlearned state in Line 11 is

$$(1 - p_{ul}^{\hat{\pi}^{(k)}})^{n_{\text{exp}}} \leq (1 - \epsilon/8)^{n_{\text{exp}}} \leq \exp(-\epsilon n_{\text{exp}}/8).$$

This follows from independence of the n_{exp} trajectories sampled from $\hat{\pi}^{(k)}$. $n_{\text{exp}} \geq \frac{8}{\epsilon}\ln\left(\frac{4MH}{\delta}\right)$ ensures that the probability of not hitting unlearned states in any of the MH iterations is at most $\delta/4$.

In total, except with probability $\delta/2 + \delta/4 + \delta/4$ (for the three events we considered above), on every iteration, either the algorithm finds a near optimal policy and returns it, or it visits a previously unlearned state, which subsequently becomes learned. Since there are at most MH states, this proves that with probability at least $1 - \delta$, the algorithm returns a policy that is at most ϵ-suboptimal. □

6.8.6 Proof of Sample Complexity: Theorem 83

We now have all parts to complete the proof of Theorem 83. For the calculation, we instantiate all the parameters as

$$\begin{aligned}
\epsilon_{\text{stat}} &= \epsilon_{\text{sub}} = \epsilon_{\text{feas}} = \frac{\epsilon}{2^6 7 H^2 T_{\max}}, \\
\phi_h &= (H - h + 1)(6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}), \quad T_{\max} = MHn_{\text{exp}} + M, \\
n_{\text{test}} &= \frac{\log\left(12KHT_{\max}|\mathcal{G}|/\delta\right)}{2\epsilon_{\text{stat}}^2}, \quad n_{\text{train}} = \frac{16K\log(12HT_{\max}|\mathcal{G}||\Pi|/\delta)}{\epsilon_{\text{stat}}^2}, \\
n_{\text{exp}} &= \frac{8\log(4MH/\delta)}{\epsilon}, \quad n_{\text{eval}} = \frac{32\log(8MH/\delta)}{\epsilon^2}.
\end{aligned}$$

These settings suffice to apply all of the above lemmas and therefore with these settings the algorithm outputs a policy that is at most ϵ-suboptimal, except with probability δ. For the sample complexity, since $T_{\max}$ is an upper bound on the number of data sets we collect (because $T_{\max}$ is an upper bound on the number of execution of `dfslearn` at any level), and we also n_{eval} trajectories for each of the MH iterations of `MetaAlg`, the total sample complexity is

$$\begin{aligned}
&HT_{\max}n_{\text{train}} + KHT_{\max}n_{\text{test}} + MHn_{\text{eval}} \\
&= O\left(\frac{T_{\max}^3 KH^5}{\epsilon^2}\log\left(\frac{MKH}{\epsilon\delta}|\mathcal{G}||\Pi|\log(MH/\delta)\right) + \frac{MH}{\epsilon^2}\log(MH/\delta)\right) \\
&= O\left(\frac{M^3KH^8}{\epsilon^5}\log^3(MH/\delta)\log\left(\frac{MKH}{\epsilon\delta}|\mathcal{G}||\Pi|\log(MH/\delta)\right)\right).
\end{aligned}$$

This proves the theorem. □

6.8.7 Extension: `VALOR` with Constrained Policy Optimization

We note that Theorem 83 suffers relatively high sample complexity compared to the original `LSVEE`. The issue is that `VALOR` pools all the data sets together for policy optimization (Algorithm 10). This implicitly weights all data sets uniformly, and allows some undesired trade-off: the policy that maximizes the objective could sacrifice significant amount of value on one data set (for some hidden state) to gain slightly more value on many others, only to find out later that the sacrificed state is visited very often during execution. This is the well-known distribution mismatch issue of reinforcement learning.

To address this issue and attain better sample complexity results, Algorithm 12 shows an alternative to the policy optimization component of `VALOR` in Algorithm 10. Instead of using an unconstrained optimization problem, it finds the policy through a feasibility problem, and hence avoid the undesired trade-off mentioned above. The computation can be implemented by the multi data set classification oracle defined in Section 6.7.

Algorithm 12: Constrained policy optimization with local values

1 **Function** `polvalfun()`
2 $\quad\hat{V}^\star \leftarrow V$ associated with only dataset in $\mathcal{D}_1$;
3 $\quad$**for** $h = 1 : H$ **do**
4 $\quad\quad$Pick $\hat{\pi}_h$ such that the following constraints are violated at most ϵ_{feas} for all $(D, V, \{V_a\}_a) \in \mathcal{D}_h$: $\hat{\mathbf{E}}_D[K\mathbf{1}\{\pi(x_h) = a_h\}(r_h + V_{a_h})] \geq V - 2\phi_h + 4\epsilon_{\text{stat}} + \epsilon_{\text{sub}}$;
5 $\quad$**return** $\hat{\pi}_{1:H}, \hat{V}^\star$;

Below, we prove a stronger version of Proposition 99 (which is for Algorithm 10) for this approach based on feasibility. First, we show that $\pi^\star$ is always a feasible choice in Line 4 in event $\mathcal{E}$.

Lemma 101. *Assume $g^\star \in \mathcal{G}_h$, $\pi^\star \in \Pi_h$ and $\phi_h = (H + 1 - h)(6\epsilon_{stat} + 2\epsilon_{sub} + \epsilon_{feas})$ for all $h \in [H]$. Then $\pi^\star$ is a valid choice in Line 4 of `polvalfun` in Algorithm 12 in event $\mathcal{E}$.*

Proof. Consider a single data set $(D, V, \{V_a\}_a) \in \mathcal{D}_h$ that is associated with state $s \in \mathcal{S}_h$. Using

Proposition 97, we can bound the deviation of the optimal policy for each constraint as

$$\begin{aligned}
&V - \hat{\mathbf{E}}_D[K\mathbf{1}\{\pi^\star(x_h) = a_h\}(r_h + V_{a_h})] \\
&\leq V^\star(s) + \phi_h - 2\epsilon_{\text{stat}} - \hat{\mathbf{E}}_D[K\mathbf{1}\{\pi^\star(x_h) = a_h\}(r_h + V_{a_h})] \\
&\leq V^\star(s) + \phi_h - 2\epsilon_{\text{stat}} - \mathbb{E}_s[K\mathbf{1}\{\pi^\star(x_h) = a_h\}(r_h + V_{a_h})] + \epsilon_{\text{stat}} \\
&\leq V^\star(s) + \phi_h + 2\epsilon_{\text{stat}} - \mathbb{E}_s[K\mathbf{1}\{\pi^\star(x_h) = a_h\}(r_h + V^\star(s \circ a_h))] + \phi_{h+1} + \epsilon_{\text{sub}} + \epsilon_{\text{feas}} \\
&= \phi_h + 2\epsilon_{\text{stat}} + \phi_{h+1} + \epsilon_{\text{sub}} + \epsilon_{\text{feas}} = 2\phi_h - 4\epsilon_{\text{stat}} - \epsilon_{\text{sub}}.
\end{aligned}$$

Here we first used that V is close to the optimal value $V^\star(s)$, the deviation bounds next and finally leveraged that V_a is a good estimate. Since that inequality holds for all constraints, $\pi^\star$ is feasible. □

We now show that Algorithm 12 produces policies with a better guarantees than its unconstrained counterpart. The difference is that we eliminate the $T_{\max}$ term in the error bound.

Proposition 102 (Improvement over Proposition 99)**.** *Assume $g^\star \in \mathcal{G}_h$ and that we are in event $\mathcal{E}$. Recall the definition $\phi_h = (H + 1 - h)(6\epsilon_{stat} + 2\epsilon_{sub} + \epsilon_{feas})$ for all $h \in [H]$. Then the policy $\hat{\pi} = \hat{\pi}_{1:H}$ returned by* `polvalfun` *in Algorithm 12 satisfies*

$$V^{\hat{\pi}} \geq V^\star - p_{ul}^{\hat{\pi}} - 32H^2(\epsilon_{stat} + \epsilon_{feas} + \epsilon_{sub})$$

where $p_{ul}^{\hat{\pi}} = \mathbb{P}(\exists h \in [H] \ : \ s_h \notin \mathcal{S}^{learned} \mid a_{1:H} \sim \hat{\pi})$ is the probability of hitting an unlearned state when following $\hat{\pi}$.

Proof. We bound the difference of how much following $\hat{\pi}_h$ for one time step can hurt per state using Proposition 97. First note that by Lemma 101, the optimization problem always has a feasible solution in event $\mathcal{E}$, so $\hat{\pi}_h$ is well defined. For a state $s \in \mathcal{S}_h^{\text{learned}}$, we have

$$\begin{aligned}
&V^\star(s) - Q^\star(s, \hat{\pi}_h) \\
&= \mathbb{E}_s[K(\mathbf{1}\{\pi_h^\star(x_h) = a_h\} - \mathbf{1}\{\hat{\pi}_h(x_h) = a_h\})(r_h + g_{h+1}^\star(x_{h+1})] \\
&\leq \mathbb{E}_s[K(\mathbf{1}\{\pi_h^\star(x_h) = a_h\} - \mathbf{1}\{\hat{\pi}_h(x_h) = a_h\})(r_h + V_{a_h})] + 2\phi_{h+1} + 6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + 2\epsilon_{\text{feas}} \\
&\leq \hat{\mathbf{E}}_D[K(\mathbf{1}\{\pi_h^\star(x_h) = a_h\} - \mathbf{1}\{\hat{\pi}_h(x_h) = a_h\})(r_h + V_{a_h})] + 2\phi_{h+1} + 8\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + 2\epsilon_{\text{feas}} \\
&\leq V + \epsilon_{\text{sub}} - V + 2\phi_h - 4\epsilon_{\text{stat}} - \epsilon_{\text{sub}} + \epsilon_{\text{feas}} + 2\phi_{h+1} + 8\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + 2\epsilon_{\text{feas}} \\
&= 4\phi_{h+1} + 16\epsilon_{\text{stat}} + 5\epsilon_{\text{feas}} + 6\epsilon_{\text{sub}} = 4\phi_h - 8\epsilon_{\text{stat}} - 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}.
\end{aligned}$$

Here $(D, V, \{V_a\})$ is one of the data sets in $\mathcal{D}_h$ that is associated with s, which has optimal policy value V by construction. We first applied definitions and then used that V_a are good value estimates. Subsequently we applied the deviation bounds and finally leveraged the definition of V and the approximate feasibility of $\hat{\pi}_h$. Using Lemma 98, the suboptimality of $\hat{\pi}$ is therefore at most

$$\begin{aligned}
V^\star - V^{\hat{\pi}} &\leq p_{ul}^{\hat{\pi}} + (1 - p_{ul}^{\hat{\pi}}) \sum_{h=1}^{H} (4\phi_{h+1} + 16\epsilon_{\text{stat}} + 5\epsilon_{\text{feas}} + 6\epsilon_{\text{sub}}) \\
&\leq p_{ul}^{\hat{\pi}} + 16H\epsilon_{\text{stat}} + 6H\epsilon_{\text{sub}} + 5H\epsilon_{\text{feas}} + 4\sum_{h=1}^{H} \phi_{h+1} \\
&\leq p_{ul}^{\hat{\pi}} + 16H\epsilon_{\text{stat}} + 6H\epsilon_{\text{sub}} + 5H\epsilon_{\text{feas}} + 4(6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}) \sum_{h=1}^{H} (H - h) \\
&\leq p_{ul}^{\hat{\pi}} + 16H\epsilon_{\text{stat}} + 6H\epsilon_{\text{sub}} + 5H\epsilon_{\text{feas}} + 2H^2(6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}) \\
&\leq p_{ul}^{\hat{\pi}} + 32H^2(\epsilon_{\text{stat}} + \epsilon_{\text{feas}} + \epsilon_{\text{sub}}).
\end{aligned}$$

□

Using this improved policy guarantee, we obtain a tighter analysis of `MetaAlg` that does not have a dependency on $T_{\max}$ in ϵ_{stat}.

Lemma 103. *Consider running* `MetaAlg` *with* `dfslearn` *and* `polvalfun` *(Algorithm 9 + 12 + 11) with parameters*

$$n_{exp} \geq \frac{8}{\epsilon}\ln\left(\frac{4MH}{\delta}\right), \quad n_{eval} \geq \frac{32}{\epsilon^2}\ln\left(\frac{8MH}{\delta}\right), \quad \epsilon_{stat} = \epsilon_{feas} = \epsilon_{sub} = \frac{\epsilon}{2^{10}H^2}.$$

Then with probability at least $1-\delta$, `MetaAlg` *returns a policy that is at least ϵ-optimal after at most MK iterations.*

Proof. The proof is identical to the proof of Lemma 100 except using Proposition 102 in place of Proposition 99, and using Lemma 101 to guarantee that the optimization problem in Line 4 is always feasible, in event $\mathcal{E}$. □

Finally, we are ready to assemble all statements to the following sample-complexity bound:

Theorem 104. *Consider a Markovian CDP with deterministic dynamics over M hidden states, as described in Section 6.3. When $\pi^\star \in \Pi$ and $g^\star \in \mathcal{G}$ (Assumptions 87 and 88 hold), for any $\epsilon, \delta \in (0,1)$, the local value algorithm with constrained policy optimization (Algorithm 9 + 12 + 11) returns a policy π such that $V^\star - V^\pi \leq \epsilon$ with probability at least $1-\delta$, after collecting at most $\tilde{O}\left(\frac{MKH^6}{\epsilon^3}\log(|\mathcal{G}||\Pi|/\delta)\log(1/\delta)\right)$ trajectories.*

Proof. We now have all parts to complete the proof of Theorem 83. For the calculation, we instantiate all the parameters as

$$\begin{aligned}
&\epsilon_{\text{stat}} = \epsilon_{\text{feas}} = \epsilon_{\text{sub}} = \frac{\epsilon}{2^{10}H^2}, \quad \phi_h = (H-h+1)(6\epsilon_{\text{stat}} + 2\epsilon_{\text{sub}} + \epsilon_{\text{feas}}),\\
&T_{\max} = MHn_{\text{exp}} + M,\\
&n_{\text{test}} = \frac{\log\left(12KHT_{\max}|\mathcal{G}|/\delta\right)}{2\epsilon_{\text{stat}}^2}, \quad n_{\text{train}} = \frac{16K\log(12HT_{\max}|\mathcal{G}||\Pi|/\delta)}{\epsilon_{\text{stat}}^2},\\
&n_{\text{exp}} = \frac{8\log(4MH/\delta)}{\epsilon}, \quad n_{\text{eval}} = \frac{32\log(8MH/\delta)}{\epsilon^2}.
\end{aligned}$$

These settings suffice to apply all of the above lemmas for these algorithms and therefore with these settings the algorithm outputs a policy that is at most ϵ-suboptimal, except with probability δ. For the sample complexity, since $T_{\max}$ is an upper bound on the number of data sets we collect (because $T_{\max}$ is an upper bound on the number of execution of `dfslearn` at any level), and we also n_{eval} trajectories for each of the MH iterations of `MetaAlg`, the total sample complexity is

$$\begin{aligned}
&HT_{\max}n_{\text{train}} + KHT_{\max}n_{\text{test}} + MHn_{\text{eval}}\\
&= O\left(\frac{T_{\max}KH^5}{\epsilon^2}\log\left(\frac{MKH}{\epsilon\delta}|\mathcal{G}||\Pi|\log(MH/\delta)\right) + \frac{MH}{\epsilon^2}\log(MH/\delta)\right)\\
&= O\left(\frac{MKH^6}{\epsilon^3}\log(MH/\delta)\log\left(\frac{MKH}{\epsilon\delta}|\mathcal{G}||\Pi|\log(MH/\delta)\right)\right).
\end{aligned}$$

□

6.9 Alternative Algorithms

Theorem 105 (Informal statement)**.** *Under Assumption 89 or Assumptions 88+90, there exist oracle-efficient algorithms with polynomial sample complexity in CDPs (contextual decision processes) with deterministic dynamics over small hidden states. These algorithms do not store or use local values.*

6.9.1 Algorithm with Two-Sample State-Identity Test

See Algorithm 9 + 13. The algorithm uses a novel state identity test which compares two distributions using a two-sample test (Gretton et al., 2012) in Line 10 (recall that $\mathcal{G}_h = \mathcal{G}$ for $h \in [H]$ and $\mathcal{G}_{H+1} = \{x \mapsto 0\}$). Such an identity test mechanism is very different from the one used in the VALOR algorithm, and the two mechanisms have very different behavior. For example, if $\mathcal{G} = \{g^\star\}$, the local value algorithm will claim every state s as "not new" because it knows the optimal value $V^\star(s)$, whereas the two-sample test may still declare a state s to be new if $\mathbb{E}_s[g^\star(x)] \neq \mathbb{E}_{s'}[g^\star(x)]$ for any previously visited s'. On the other hand, the two-sample test algorithm may not have learned $V^\star(s)$ at all when it claims that a state s is not new. Given the novelty of the mechanism, we believe analyzing the two-sample test algorithm and understanding its computational and statistical properties enriches our toolkit for dealing with the challenges addressed in this paper.

Algorithm 13: Algorithm with Two-Sample State-Identity Test

1 **Function** `polvalfun()`
2 $\quad \hat{g}_{H+1} \leftarrow 0$;
3 $\quad$ **for** $h = H : 1$ **do**
4 $\quad\quad \hat{\pi}_h \leftarrow \operatorname{argmax}_{\pi \in \Pi_h} \sum_{D \in \mathcal{D}_h^{\text{learned}}} \hat{\mathbf{E}}_D[K\mathbf{1}\{\pi(x_h) = a_h\}(r_h + \hat{g}_{h+1}(x_{h+1}))]$;
5 $\quad\quad \hat{g}_h \leftarrow \operatorname{argmin}_{g \in \mathcal{G}_h} \sum_{D \in \mathcal{D}_h^{\text{val}}} \hat{\mathbf{E}}_D[K\mathbf{1}\{\hat{\pi}_h(x_h) = a_h\}(g(x_h) - r_h - \hat{g}_{h+1}(x_{h+1}))^2]$;
6 $\quad \hat{V}^\star \leftarrow \hat{\mathbf{E}}_D[\hat{g}_1(x_1)]$ where D is the only distribution in $\mathcal{D}_1^{\text{val}}$;
7 $\quad$ **return** $\hat{\pi}_{1:H}, \hat{V}^\star$;
8 **Function** `dfslearn(`$a_{1:h-1}$`)`
9 $\quad \tilde{D} \leftarrow$ sample $x_h \sim a_{1:h-1}, a_h \sim \text{Unif}(\mathcal{A}), r_h, x_{h+1}$;
10 $\quad d_{\text{MMD}} \leftarrow \min_{D \in \mathcal{D}_h^{\text{val}}} \sup_{g \in \mathcal{G}_h} \left| \hat{\mathbf{E}}_D[g(x_h)] - \hat{\mathbf{E}}_{\tilde{D}}[g(x_h)] \right|$;
11 $\quad$ **if** $d_{MMD} \leq 2\tau$ ***and*** IS_RECURSIVE_CALL **then**
12 $\quad\quad$ **return**
13 $\quad$ **if** $d_{MMD} > 2\tau$ **then**
14 $\quad\quad$ Add $\tilde{D}$ to $\mathcal{D}_h^{\text{val}}$
15 $\quad$ Add $\tilde{D}$ to $\mathcal{D}_h^{\text{learned}}$;
16 $\quad$ **for** $a \in \mathcal{A}$ **do**
17 $\quad\quad$ `dfslearn(`$a_{1:h-1} \circ a$`)` ;

Computational considerations

The two-sample test algorithm requires three types nontrivial computation. Line 4 requires importance weighted policy optimization, which is simply a call to the CSC oracles. Line 5 performs squared-loss regression on $\mathcal{G}_h$, which is a call to a LS oracle.

The slightly unusual computation occurs on Line 10: we compute the (empirical) Maximum Mean Discrepancy (MMD) between D and $\tilde{D}$ against the function class $\mathcal{G}_h$, and take the minimum over $D \in \mathcal{D}^{\text{val}}$. First, since $|\mathcal{D}_h^{\text{val}}|$ remains small over the execution of the algorithm, the minimization over $D \in \mathcal{D}_h^{\text{val}}$ can be done by enumeration. Then, for a fixed D, computing the MMD is a linear optimization problem over $\mathcal{G}_h$. In the special case where $\mathcal{G}_h$ is the unit ball in a Reproducing Kernel Hilbert Space (RKHS) (Schölkopf

and Smola, 2002), MMD can be computed in closed form by $O(n^2)$ kernel evaluations, where n is the number of data points involved (Gretton et al., 2012).

To unclutter the sample-complexity analysis, we assume that perfect oracles, i.e., $\epsilon_{\text{feas}} = \epsilon_{\text{sub}} = 0$.

Sample complexity

Theorem 106. *Consider the same Markovian CDP setting as in Theorem 83 but we explicitly require here that the process is an MDP over $\mathcal{X}$. Under Assumption 89, for any $\epsilon, \delta \in (0,1)$, the two-sample state-identity test algorithm (Algorithm 9+13) returns a policy π such that $V^\star - V^\pi \leq \epsilon$ with probability at least $1-\delta$, after collecting at most $\tilde{O}\left(\frac{M^2K^2H^6}{\epsilon^4}\log(|\mathcal{G}||\Pi|/\delta)\log^2(1/\delta)\right)$ trajectories.*

For this algorithm, we use the following notion of learned state:

Definition 107 (Learned states)**.** *Denote the sequence of states whose data sets are added to $\mathcal{D}_h^{learned}$ as $\mathcal{S}_h^{learned}$. States that are in $\mathcal{S}^{learned} = \bigcup_{h\in[H]} \mathcal{S}_h^{learned}$ are called learned. The sequence of states whose data sets are added to $\mathcal{D}_h^{val}$ are denoted by $\mathcal{S}_h^{val}$. Let $\mathcal{S}_h^{check}$ denote the set of all states that have been reached by any previous* `dfslearn` *call at level h.*

Fact 108. *We have $\mathcal{S}_h^{val} \subseteq \mathcal{S}_h^{learned} \subseteq \mathcal{S}_h^{check}$. Furthermore, $\forall s \in \mathcal{S}_h^{learned}$ and $a \in \mathcal{A}$, $s \circ a \in \mathcal{S}_{h+1}^{check}$.*

Define the following short-hand notations for the objective functions used in Algorithm 13:

$$
\begin{aligned}
V_D(\pi; g') &:= \hat{\mathbf{E}}_D[K\mathbf{1}\{\pi(x) = a\}(r + g'(x'))]. \\
V_{\mathcal{D}_h^{\text{learned}}}(\pi; g') &:= \sum_{D \in \mathcal{D}_h^{\text{learned}}} V_D(\pi; g'). \\
L_D(g; \pi, g') &:= \hat{\mathbf{E}}_D[K\mathbf{1}\{\pi(x) = a\}(g(x) - r - g'(x'))^2]. \\
L_{\mathcal{D}_h^{\text{val}}}(g; \pi, g') &:= \sum_{D \in \mathcal{D}_h^{\text{val}}} L_D(g; \pi, g').
\end{aligned}
$$

Also define V_s, $V_{\mathcal{S}_h^{\text{learned}}}$, L_s, $L_{\mathcal{S}_h^{\text{val}}}$ as the population version of V_D, $V_{\mathcal{D}_h^{\text{learned}}}$, L_D, $L_{\mathcal{D}_h^{\text{val}}}$, respectively.

Concentration Results. For our analysis we rely on the following concentration bounds that define the good event $\mathcal{E}$. This definition involves parameters τ, τ_L, τ_V whose values we will set later.

Definition 109. *Let $\mathcal{E}$ denote the event that for all $h \in [H]$ the total number of calls to* `dfslearn`(p) *at level h is at most $T_{\max} = M(K+1)(1 + Hn_{exp})$ during the execution of* `MetaAlg` *and that for all these calls to* `dfslearn`(p) *the following deviation bounds hold for all $g \in \mathcal{G}_h, g' \in \mathcal{G}_{h+1}$ and $\pi \in \Pi_h$ (where $\tilde{D}$ is the data set of n_{train} samples from Line 9 and s is the state reached by p):*

$$
\begin{aligned}
|\hat{\mathbf{E}}_{\tilde{D}}[g(x)] - \mathbb{E}_s[g(x)]| &\leq \tau && (6.5) \\
|V_{\tilde{D}}(\pi; g') - V_s(\pi; g')| &\leq \tau_V \\
|L_{\tilde{D}}(g; \pi, g') - L_s(g; \pi, g')| &\leq \tau_L. && (6.6)
\end{aligned}
$$

We now show that this event has high probability.

Lemma 110. *Set n_{train} so that*

$$n_{train} \geq \max\Big\{ \frac{1}{2\tau^2} \ln\left(\frac{12HT_{\max}|\mathcal{G}|}{\delta}\right), \\ \frac{16K}{\tau_V^2} \ln\left(\frac{12HT_{\max}|\mathcal{G}||\Pi|}{\delta}\right), \\ \frac{32K}{\tau_L^2} \ln\left(\frac{12HT_{\max}|\mathcal{G}|^2|\Pi|}{\delta}\right) \Big\}.$$

Then $\mathbb{P}[\mathcal{E}] \geq 1 - \delta/2$ where $\mathcal{E}$ is defined in Definition 109. In addition, in event $\mathcal{E}$, during all calls the sequences are bounded as $|\mathcal{S}_h^{val}| \leq M$ and $|\mathcal{S}_h^{learned}| \leq T_{\max}$.

Proof. Let us first focus on one call to `dfslearn`, say at path p at level h. First, observe that the data set $\tilde{D}$ is a set of n_{train} transitions sampled i.i.d. from the state s that is reached by p. By Hoeffding's inequality and a union bound, with probability $1 - \delta'$, for all $g \in \mathcal{G}_h$

$$\left|\hat{\mathbf{E}}_{\tilde{D}}[g(x_h)] - \mathbb{E}_s[g(x_h)]\right| \leq \sqrt{\frac{\log(2|\mathcal{G}|/\delta')}{2n_{\text{train}}}}.$$

With $\delta' = \frac{\delta}{6HT_{\max}}$ the choice for n_{train} let us bound the LHS by τ.

For the random variable $K\mathbf{1}\{\pi(x_h) = a_h\}(r_h + g'(x_{h+1}))$, since a_h is chosen uniformly at random, it is not hard to see that both the variance and the range are at most $2K$ (see for example Lemma 14 by Jiang, Krishnamurthy, et al. (2017)). Applying Bernstein's inequality and a union bound, for all $\pi \in \Pi_h$ and $g \in \mathcal{G}_{h+1}$, we have

$$|V_{\tilde{D}}(\pi; g') - V_s(\pi; g')| \leq \sqrt{\frac{4K\log(2|\mathcal{G}||\Pi|/\delta')}{n_{\text{train}}}} + \frac{4K}{3n_{\text{train}}}\log(2|\mathcal{G}||\Pi|/\delta')$$

with probability $1 - \delta'$. As above, with $\delta' = \frac{\delta}{6HT_{\max}}$ our choice of n_{train} ensures that this deviation is bound by τ_V.

Similarly, we apply Bernstein's inequality to the random variable $K\mathbf{1}\{\pi(x_h) = a_h\}(g(x_h) - r_h - g'(x_{h+1}))^2$ which has range and variance at most $4K$. Combined with a union bound over all $g \in \mathcal{G}_h, g' \in \mathcal{G}_{h+1}, \pi \in \Pi_h$ we have that with probability $1 - \delta'$,

$$|L_{\tilde{D}}(g; \pi, g') - L_s(g; \pi, g')| \leq \sqrt{\frac{8K\log(2|\mathcal{G}|^2|\Pi|/\delta')}{n_{\text{train}}}} + \frac{2K}{n_{\text{train}}}\log(2|\mathcal{G}|^2|\Pi|/\delta') \leq \tau_L.$$

This last inequality is based on the choice for n_{train} and $\delta' = \frac{\delta}{6HT_{\max}}$. For details on this concentration bound see for example Lemma 14 by Jiang, Krishnamurthy, et al. (2017). Using a union bound, the deviation bounds (6.5)–(6.6) hold for a single call to `dfslearn` with probability $1 - 3\delta'$.

Consider now the event $\mathcal{E}'$ that these bounds hold for the first $T_{\max}$ calls at each level h. Applying a union bound let us bound $\mathbb{P}(\mathcal{E}') \geq 1 - 3HT_{\max}\delta' = 1 - \frac{\delta}{2}$. It remains to show that $\mathcal{E}' \subseteq \mathcal{E}$.

First note that in event $\mathcal{E}'$ in the first $T_{\max}$ calls to `dfslearn` at level h, the algorithm does not call itself recursively during a recursive call if p leads to a state $s \in \mathcal{S}_h^{\text{val}}$. To see this assume p leads to a state $s \in \mathcal{S}_h^{\text{val}}$ and let $D \in \mathcal{D}_h^{\text{val}}$ be a data set sampled from this state. This means that $\tilde{D}$ and D are sampled from the same distribution, and as such, we have for every $g \in \mathcal{G}_h$

$$|\hat{\mathbf{E}}_{\tilde{D}}[g(x)] - \hat{\mathbf{E}}_D[g(x)]| \leq |\hat{\mathbf{E}}_{\tilde{D}}[g(x)] - \mathbb{E}_s[g(x)]| + |\mathbb{E}_s[g(x)] - \hat{\mathbf{E}}_D[g(x)]| \leq 2\tau. \tag{6.7}$$

Therefore $d_{MMD} \leq 2\tau$, the condition in the first clause is satisfied, and the algorithm does not recurse. If this condition is not satisfied, the algorithm adds $\tilde{D}$ to $\mathcal{D}_h^{\text{val}}$. Therefore, the initial call to `dfslearn` at the root can result in at most MK recursive calls per level, since the identity tests must return true on identical states.

Further, for any fixed level, we issue at most MHn_{exp} additional calls to `dfslearn`, since `MetaAlg` has at most MH iterations and in each one, `dfslearn` is called n_{exp} times per level. Any new state that we visit in this process was already counted by the MK calls per level in the initial execution of `dfslearn`. On the other hand, these calls always descend to the children, so the number of calls to old states is at most $M(1+K)Hn_{\text{exp}}$ per level. In total the number of calls to `dfslearn` per level is at most $M(1+K)Hn_{\text{exp}} + MK \leq T_{\max}$, and $\mathbb{P}(\mathcal{E}) \leq \delta/2$ follows.

Further, the bound $|\mathcal{S}_h^{\text{learned}}| \leq T_{\max}$ follows from the fact that per call only one state can be added to $\mathcal{S}_h^{\text{learned}}$ and there are at most $T_{\max}$ calls. The bound $|\mathcal{S}_h^{\text{val}}| \leq M$ follows from the fact that in $\mathcal{E}$ no state can be added twice to $\mathcal{S}_h^{\text{val}}$ since as soon as it is in $\mathcal{S}_h^{\text{val}}$ once, $d_{MMD} \leq 2\tau$ holds (see Eq.(6.7)) and the current data set is not added to $\mathcal{D}_h^{\text{val}}$. □

Depth-first search and learning optimal values. We now prove that `polvalfun` and `dfslearn` produce good value function estimates.

Proposition 111. *In event $\mathcal{E}$, consider an execution of* `polvalfun` *and let $\{\hat{g}_h, \hat{\pi}_h\}_{h\in[H]}$ denote the learned value functions and policies. Then every state s in $\mathcal{S}_h^{\text{check}}$ satisfies*

$$|\mathbb{E}_s[\hat{g}_h(x_h)] - \mathbb{E}_s[g^\star(x_h)]| \leq (H+1-h)(2M\tau_V + \sqrt{4M^2\tau_V + 2T_{\max}\tau_L} + 8\tau), \tag{6.8}$$

and every learned state $s \in \mathcal{S}_h^{\text{learned}}$ satisfies

$$V^\star(s) - Q^\star(s, \hat{\pi}_h) \leq 2M\tau_V + 2(H-h)(2M\tau_V + \sqrt{4M^2\tau_V + 2T_{\max}\tau_L} + 8\tau). \tag{6.9}$$

Proof. We prove both inequalities simultaneously by induction over h. For convenience, we use the following short hand notations: $\epsilon_V = M\tau_V$ and $\epsilon_L = T_{\max}\tau_L$. Using this notation, in event $\mathcal{E}$, $|V_{\mathcal{D}_h^{\text{val}}}(\pi; g') - V_{\mathcal{S}_h^{\text{val}}}(\pi; g')| \leq \epsilon_V$ and $|L_{\mathcal{D}_h^{\text{learned}}}(g; \pi, g') - L_{\mathcal{S}_h^{\text{learned}}}(g; \pi, g')| \leq \epsilon_L$ hold for all g, g' and π.

Base case: Both statement holds trivially for $h = H+1$ since the LHS is 0 and the RHS is non-negative. In particular there are no actions, so Eq. (6.9) is trivial.

Inductive case: Assume that Eq. (6.8) holds on level $h+1$. For any learned $s \in \mathcal{S}_h^{\text{learned}}$, we first show that $\hat{\pi}_h$ achieves high value compared to $\pi^\star_{\hat{g}_{h+1}}$ (recall its definition from Assumption 89) under $V_s(\cdot; \hat{g}_{h+1})$:

$$\begin{aligned} V_s(\pi^\star_{\hat{g}_{h+1}}; \hat{g}_{h+1}) - V_s(\hat{\pi}_h; \hat{g}_{h+1}) &\leq \sum_{s\in\mathcal{S}_h^{\text{learned}}} V_s(\pi^\star_{\hat{g}_{h+1}}; \hat{g}_{h+1}) - V_s(\hat{\pi}_h; \hat{g}_{h+1}) \\ &= V_{\mathcal{S}_h^{\text{learned}}}(\pi^\star_{\hat{g}_{h+1}}; \hat{g}_{h+1}) - V_{\mathcal{S}_h^{\text{learned}}}(\hat{\pi}_h; \hat{g}_{h+1}) \\ &\leq V_{\mathcal{D}_h^{\text{learned}}}(\pi^\star_{\hat{g}_{h+1}}; \hat{g}_{h+1}) - V_{\mathcal{D}_h^{\text{learned}}}(\hat{\pi}_h; \hat{g}_{h+1}) + 2\epsilon_V \leq 2\epsilon_V. \end{aligned}$$

Eq. (6.9) follows as a corollary:

$$\begin{aligned}
&V^\star(s) - Q^\star(s, \hat{\pi}_h) \\
&= V_s(\pi^\star; g^\star) - V_s(\hat{\pi}_h; g^\star) \\
&\le V_s(\pi^\star; \hat{g}_{h+1}) - V_s(\hat{\pi}_h; \hat{g}_{h+1}) + 2 \sup_{s' \text{ being child of } s} |\mathbb{E}_{x' \sim s'}[\hat{g}_{h+1}(x_{h+1}) - g^\star(x_{h+1})]| \\
&\le V_s(\pi^\star_{\hat{g}_{h+1}}; \hat{g}_{h+1}) - V_s(\hat{\pi}_h; \hat{g}_{h+1}) + 2(H-h)(2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 8\tau) \\
&\le 2\epsilon_V + 2(H-h)(2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 8\tau).
\end{aligned}$$

This proves Eq. (6.9) at level h. The rest of the proof proves Eq.(6.8). First we introduce and recall the definitions:

$$\begin{aligned}
g_{\hat{\pi}_h, \hat{g}_{h+1}}(x) &= \mathbb{E}[r + \hat{g}_{h+1}(x_{h+1}) \mid x_h = x, a_h = \hat{\pi}_h(x)], \\
g_{\star, \hat{g}_{h+1}}(x) &= \mathbb{E}[r + \hat{g}_{h+1}(x_{h+1}) \mid x_h = x, a_h = \pi^\star_{\hat{g}_{h+1}}(x)].
\end{aligned}$$

Note that $g_{\hat{\pi}_h, \hat{g}_{h+1}} \notin \mathcal{G}_h$ in general, but it is the Bayes optimal predictor for the squared losses $L_s(\cdot; \hat{\pi}_h, \hat{g}_{h+1})$ for all s simultaneously. On the other hand, Assumption 89 guarantees that $g_{\star, \hat{g}_{h+1}} \in \mathcal{G}_h$, for any $\hat{g}_{h+1}$.

The LHS of Eq.(6.8) can be bounded as

$$|\mathbb{E}_s[g^\star(x_h)] - \mathbb{E}_s[\hat{g}_h(x_h)]| \le \left|\mathbb{E}_s[g^\star(x_h)] - \mathbb{E}_s[g_{\hat{\pi}_h, \hat{g}_{h+1}}(x_h)]\right| + \left|\mathbb{E}_s[g_{\hat{\pi}_h, \hat{g}_{h+1}}(x_h)] - \mathbb{E}_s[\hat{g}_h(x_h)]\right| \quad (6.15)$$

To bound the first term in Eq.(6.15),

$$\begin{aligned}
\left|\mathbb{E}_s[g^\star(x_h)] - \mathbb{E}_s[g_{\hat{\pi}_h, \hat{g}_{h+1}}(x_h)]\right| &\le \left|\mathbb{E}_s[g^\star(x_h)] - \mathbb{E}_s[g_{\star, \hat{g}_{h+1}}(x_h)]\right| + \left|\mathbb{E}_s[g_{\star, \hat{g}_{h+1}}(x_h)] - \mathbb{E}_s[g_{\hat{\pi}_h, \hat{g}_{h+1}}(x_h)]\right| \\
&= \left|\mathbb{E}_s\left[g^\star(x_h) - g_{\star, \hat{g}_{h+1}}(x_h)\right]\right| + \left|V_s(\pi^\star_{\hat{g}_{h+1}}; \hat{g}_{h+1}) - V_s(\hat{\pi}_h; \hat{g}_{h+1})\right| \\
&\le \left|\mathbb{E}_s\left[g^\star(x_h) - g_{\star, \hat{g}_{h+1}}(x_h)\right]\right| + 2\epsilon_V.
\end{aligned}$$

Now consider each individual context x_h emitted in $s \in \mathcal{S}_h$:

$$\begin{aligned}
&g^\star(x_h) - g_{\star, \hat{g}_{h+1}}(x_h) \\
&= \mathbb{E}_{r_h \sim R(x_h, \pi^\star(x_h))}[r_h] + \mathbb{E}_{s \circ \pi^\star(x_h)}[g^\star(x_{h+1})] - \max_{a \in \mathcal{A}} \left(\mathbb{E}_{r_h \sim R(x_h, a)}[r_h] + \mathbb{E}_{s \circ a}[\hat{g}_{h+1}(x_{h+1})]\right) \\
&\le \mathbb{E}_{r_h \sim R(x_h, \pi^\star(x_h))}[r_h] + \mathbb{E}_{s \circ \pi^\star(x_h)}[\hat{g}_{h+1}(x_h)] \\
&\quad - \max_{a \in \mathcal{A}} \left(\mathbb{E}_{r_h \sim R(x_h, a)}[r] + \mathbb{E}_{s \circ a}[\hat{g}_{h+1}(x_h)]\right) + |\mathbb{E}_{s \circ \pi^\star(x_h)}[\hat{g}_{h+1}(x_{h+1}) - g^\star(x_{h+1})]| \\
&\le |\mathbb{E}_{s \circ \pi^\star(x_h)}[\hat{g}_{h+1}(x_{h+1}) - g^\star(x_{h+1})]| \\
&\le (H-h)(2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 8\tau).
\end{aligned}$$

The second inequality is true since the second term optimizes over $a \in \mathcal{A}$ and the first term is the special case of $a = \pi^\star(x_h)$. The last inequality follows from the fact that if $s \in \mathcal{S}_h^{\text{learned}} \Rightarrow s \circ a \in \mathcal{S}_{h+1}^{\text{check}}$ and we can therefore apply the induction hypothesis. We can use the same argument to lower bound the above quantity. This gives

$$\left|\mathbb{E}_s[g^\star(x_h)] - \mathbb{E}_s[g_{\hat{\pi}_h, \hat{g}_{h+1}}(x_h)]\right| \le (H-h)(2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 8\tau) + 2\epsilon_V.$$

Next, we work with the second term in Equation (6.15):

$$\left|\mathbb{E}_s[\hat{g}_h(x_h)] - \mathbb{E}_s[g_{\hat{\pi}_h, \hat{g}_{h+1}}(x_h)]\right|$$

$$
\begin{aligned}
&\leq \sqrt{\mathbb{E}_s[\big(\hat{g}_h(x_h) - g_{\hat{\pi}_h,\hat{g}_{h+1}}(x_h)\big)^2]} \\
&= \sqrt{L_s(\hat{g}_h;\hat{\pi}_h,\hat{g}_{h+1}) - L_s(g_{\hat{\pi}_h,\hat{g}_{h+1}};\hat{\pi}_h,\hat{g}_{h+1})} \\
&\leq \sqrt{L_{\mathcal{S}_h^{\text{val}}}(\hat{g}_h;\hat{\pi}_h,\hat{g}_{h+1}) - L_{\mathcal{S}_h^{\text{val}}}(g_{\hat{\pi}_h,\hat{g}_{h+1}};\hat{\pi}_h,\hat{g}_{h+1})} \\
&\leq \sqrt{L_{\mathcal{D}_h^{\text{val}}}(\hat{g}_h;\hat{\pi}_h,\hat{g}_{h+1}) - L_{\mathcal{S}_h^{\text{val}}}(g_{\hat{\pi}_h,\hat{g}_{h+1}};\hat{\pi}_h,\hat{g}_{h+1}) + \epsilon_L} \\
&\leq \sqrt{L_{\mathcal{D}_h^{\text{val}}}(g_{\star,\hat{g}_{h+1}};\hat{\pi}_h,\hat{g}_{h+1}) - L_{\mathcal{S}_h^{\text{val}}}(g_{\hat{\pi}_h,\hat{g}_{h+1}};\hat{\pi}_h,\hat{g}_{h+1}) + \epsilon_L} \\
&\leq \sqrt{L_{\mathcal{S}_h^{\text{val}}}(g_{\star,\hat{g}_{h+1}};\hat{\pi}_h,\hat{g}_{h+1}) - L_{\mathcal{S}_h^{\text{val}}}(g_{\hat{\pi}_h,\hat{g}_{h+1}};\hat{\pi}_h,\hat{g}_{h+1}) + 2\epsilon_L} \\
&= \sqrt{\sum_{s\in\mathcal{S}_h^{\text{val}}} \mathbb{E}_{x\sim s}[(g_{\star,\hat{g}_{h+1}}(x) - g_{\hat{\pi}_h,\hat{g}_{h+1}}(x))^2] + 2\epsilon_L} \\
&\leq \sqrt{\sum_{s\in\mathcal{S}_h^{\text{val}}} \mathbb{E}_{x\sim s}[2|g_{\star,\hat{g}_{h+1}}(x) - g_{\hat{\pi}_h,\hat{g}_{h+1}}(x)|] + 2\epsilon_L} \\
&= \sqrt{\sum_{s\in\mathcal{S}_h^{\text{val}}} 2\mathbb{E}_{x\sim s}[g_{\star,\hat{g}_{h+1}}(x) - g_{\hat{\pi}_h,\hat{g}_{h+1}}(x)] + 2\epsilon_L} \\
&= \sqrt{\sum_{s\in\mathcal{S}_h^{\text{val}}} 2(V_s(\pi^\star_{\hat{g}_{h+1}};\hat{g}_{h+1}) - V_s(\hat{\pi}_h;\hat{g}_{h+1})) + 2\epsilon_L} \\
&\leq \sqrt{4M\epsilon_V + 2\epsilon_L}.
\end{aligned}
$$

Put together, we get the desired result for states $s \in \mathcal{S}_h^{\text{val}}$:

$$
|\mathbb{E}_s[g^\star(x_h)] - \mathbb{E}_s[\hat{g}_h(x_h)]| \leq (H-h)(2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 8\tau) + 2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L}.
$$

It remains to deal with states $s \in \mathcal{S}_h^{\text{check}} \setminus \mathcal{S}_h^{\text{val}}$. According to the algorithm, this only happens when the MMD test suggests that the data set $\tilde{D}$ drawn from s looks very similar to a previous data set $D \in \mathcal{D}_h^{\text{val}}$, which corresponds to some $s' \in \mathcal{S}_h^{\text{val}}$. So,

$$
\begin{aligned}
&|\mathbb{E}_s[\hat{g}_h(x_h)] - \mathbb{E}_s[g^\star(x_h)]| \\
&\leq |\mathbb{E}_{s'}[\hat{g}_h(x_h)] - \mathbb{E}_{s'}[g^\star(x_h)]| + |\mathbb{E}_{s'}[\hat{g}_h(x_h)] - \mathbb{E}_s[\hat{g}_h(x_h)]| + |\mathbb{E}_s[g^\star(x_h)] - \mathbb{E}_{s'}[g^\star(x_h)]| \\
&\leq (H-h)(2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 8\tau) + 2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} \\
&\quad + |\hat{\mathbf{E}}_D[\hat{g}_h(x_h)] - \hat{\mathbf{E}}_{\tilde{D}}[\hat{g}_h(x_h)]| + |\hat{\mathbf{E}}_D[g^\star(x_h)] - \mathbb{E}_{\tilde{D}}[g^\star(x_h)]| + 4\tau \\
&\leq (H-h)(2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 8\tau) + 2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 2\tau + 2\tau + 4\tau \\
&= (H+1-h)(2\epsilon_V + \sqrt{4M\epsilon_V + 2\epsilon_L} + 8\tau).
\end{aligned}
$$

□

Quality of Learned Policies and Meta-Algorithm Analysis. After quantifying the estimation error of the value function returned by `polvalfun`, it remains to translate that into a bound on the suboptimality of the returned policy:

Proposition 112. *Assume we are in event $\mathcal{E}$. Then the policy $\hat{\pi} = \hat{\pi}_{1:H}$ returned by* `polvalfun` *in Algorithm 13 satisfies*

$$
V^{\hat{\pi}} \geq V^\star - p_{ul}^{\hat{\pi}} - 2HM\tau_V - H^2(2M\tau_V + \sqrt{4M^2\tau_V + 2T_{\max}\tau_L} + 8\tau)
$$

where $p_{ul}^{\hat{\pi}} = \mathbb{P}(\exists h \in [H] \ : \ s_h \notin \mathcal{S}^{learned} \mid a_{1:H} \sim \hat{\pi})$ *is the probability of hitting an unlearned state when following* $\hat{\pi}$.

Proof. Proposition 111 states that for every learned state $s \in \mathcal{S}_h^{\text{learned}}$

$$V^\star(s) - Q^\star(s, \hat{\pi}_h) \leq 2M\tau_V + 2(H-h)(2M\tau_V + \sqrt{4M^2\tau_V + 2T_{\max}\tau_L} + 8\tau).$$

Using Lemma 98, we can show that $\hat{\pi}$ yields expected return that is optimal up to

$$\begin{aligned}
V^\star - V^{\hat{\pi}} \leq & p_{ul}^{\hat{\pi}} + (1 - p_{ul}^{\hat{\pi}}) \sum_{h=1}^{H} (2M\tau_V + 2(H-h)(2M\tau_V + \sqrt{4M^2\tau_V + 2T_{\max}\tau_L} + 8\tau)) \\
\leq & p_{ul}^{\hat{\pi}} + 2HM\tau_V + 2(2M\tau_V + \sqrt{4M^2\tau_V + 2T_{\max}\tau_L} + 8\tau) \sum_{h=1}^{H} (H-h) \\
\leq & p_{ul}^{\hat{\pi}} + 2HM\tau_V + H^2(2M\tau_V + \sqrt{4M^2\tau_V + 2T_{\max}\tau_L} + 8\tau).
\end{aligned}$$

□

Lemma 113. *Consider running* `MetaAlg` *with* `dfslearn` *and* `polvalfun` *(Algorithm 9 + 13) with parameters*

$$\begin{aligned}
n_{exp} \geq & \frac{8}{\epsilon} \ln\left(\frac{4MH}{\delta}\right), \qquad n_{eval} \geq \frac{32}{\epsilon^2} \ln\left(\frac{8MH}{\delta}\right), \\
\tau = & \frac{\epsilon}{2^6 3 H^2}, \quad \tau_V = \frac{\epsilon^2}{2^8 3^4 M^2 H^4}, \quad \tau_L = \frac{\epsilon^2}{2^7 3^2 H^4 T_{\max}}
\end{aligned}$$

Then with probability at least $1 - \delta$, `MetaAlg` *returns a policy that is at least* ϵ*-optimal after at most* MK *iterations.*

Proof. The proof is completely analogous to the proof of Lemma 100 except with using Proposition 112 instead of Proposition 99. We set the parameters τ, τ_L and τ_V so that the policy guarantee of Proposition 112 is $V^{\hat{\pi}} \geq V^\star - p_{ul}^{\hat{\pi}} - \epsilon/8$. More specifically, we bound the guaranteed gap as

$$\begin{aligned}
& 2HM\tau_V + H^2(2M\tau_V + \sqrt{4M^2\tau_V + 2T_{\max}\tau_L} + 8\tau) \\
\leq & 2MH\tau_V + 2MH^2\tau_V + 2MH^2\sqrt{\tau_V} + H^2\sqrt{2T_{\max}\tau_L} + 8H^2\tau \\
\leq & 6MH^2\sqrt{\tau_V} + H^2\sqrt{2T_{\max}\tau_L} + 8H^2\tau
\end{aligned}$$

and then set τ, τ_L and τ_V so that each terms evaluates to $\epsilon/24$. □

Proof of Theorem 106. We now have all parts to complete the proof of Theorem 106.

Proof. For the calculation, we instantiate all the parameters as

$$\begin{aligned}
n_{\text{exp}} = & \frac{8}{\epsilon} \ln\left(\frac{4MH}{\delta}\right), \quad n_{\text{eval}} = \frac{32}{\epsilon^2} \ln\left(\frac{8MH}{\delta}\right), \quad n_{\text{train}} = 16K\left(\frac{2}{\tau_L^2} + \frac{1}{\tau_V^2}\right) \ln\left(\frac{12HT_{\max}|\mathcal{G}|^2|\Pi|}{\delta}\right). \\
\tau = & \frac{\epsilon}{2^6 3 H^2}, \quad \tau_V = \frac{\epsilon^2}{2^8 3^4 M^2 H^4}, \quad \tau_L = \frac{\epsilon^2}{2^7 3^2 H^4 T_{\max}} \quad T_{\max} = M(K+1)(1 + Hn_{\text{exp}}).
\end{aligned}$$

These settings suffice to apply all of the above lemmas for these algorithms and therefore with these settings the algorithm outputs a policy that is at most ϵ-suboptimal, except with probability δ. For the sample

complexity, since $T_{\max}$ is an upper bound on the number of datasets we collect (because $T_{\max}$ is an upper bound on the number of execution of `dfslearn` at any level), and we also n_{eval} trajectories for each of the MH iterations of `MetaAlg`, the total sample complexity is

$$HT_{\max}n_{\text{train}} + MHn_{\text{eval}} = \tilde{O}\left(\frac{M^5H^{12}K^4}{\epsilon^7}\log(|\mathcal{G}||\Pi|/\delta)\log^3(1/\delta)\right). \qquad \square$$

6.9.2 Global Policy Algorithm

See Algorithm 14. As the other algorithms, this method learns states using depth-first search. The state identity test is similar to that of `VALOR` at a high level: for any new path p, we derive an upper bound and a lower bound on $V^\star(p)$, and prune the path if the gap is small. Unlike in `VALOR` where both bounds are derived using the value function class $\mathcal{G}$, here only the upper bound is from a value function (see Line 11), and the lower bound comes from *Monte-Carlo roll-out* with a near-optimal policy, which avoids the need for on-demand exploration.

More specifically, the global policy algorithm does not store data sets but maintains a global policy, a set of learned paths, and a set of pruned paths, all of which are updated over time. We always guarantee that $\hat{\pi}_{h:H}$ is near-optimal for any learned state at level h, and leverage this property to conduct state-identity test: if a new path p leads to the same state as a learned path q, then Eq.(6.26) yields a tight upper bound on $V^\star(p)$, which can be achieved by $\hat{\pi}_{h:H}$ up to some small error and we check by Monte-Carlo roll-outs. If the test succeeds, the path p is added to the set $\textsc{Pruned}(h)$. Otherwise, all successor states are learned (or pruned) in a recursive manner, after which the state itself becomes learned (i.e., p added to $\textsc{Learned}(h)$). Then, the policy at level h is updated to be near-optimal for the newly learned state in addition to the previous ones (Line 25). Once we change the global policy, however, all the pruned states need to be re-checked (Line 26), as their optimal values are only guaranteed to be realized by the previous global policy and not necessarily by the new policy.

Computational efficiency

The algorithm contains three non-trivial computational components. In Eq.(6.26), a linear program is solved to determine the optimal value estimate of the current path given the value of one learned state (LP oracle). In Line 24, computing the value of each learned path can be reduced to multi-class cost-sensitive classification as in the other two algorithms (CSC oracle). Finally, fitting the global policy in Line (25) requires the same problem as the policy fitting procedure discussed in Section 6.8.7 (multi data set classification oracle).

As with the previous algorithm, we assume no error in the oracles ($\epsilon_{\text{feas}} = \epsilon_{\text{sub}} = 0$) in the following to simplify the analysis.

Sample complexity

Theorem 114. *Consider a Markovian contextual decision process with deterministic dynamics over M hidden states, as described in Section 6.3. When Assumption 90 and 88 hold, for any $\epsilon, \delta \in (0,1)$, the global policy algorithm (Algorithm 14) returns a policy π such that $V^\star - V^\pi \le \epsilon$ with probability at least $1-\delta$, after collecting at most $\tilde{O}\left(\frac{M^3H^3K}{\epsilon^2}\log\left(|\Pi||\mathcal{G}|/\delta\right)\right)$ trajectories.*

In the following, we prove this statement but first introduce helpful notation:

Algorithm 14: Global Policy Algorithm

Function `main`
  Global LEARNED(h), $h \in [H]$;
  Global PRUNED(h), $h \in [H]$;
  Global $\{\hat{\pi}_h\}_{h\in[H]}$;
  `dfslearn` $(\circ)$;
  return $\{\hat{\pi}_h\}_{h\in[H]}$;
Function `TestLearned` (p, h)
  Collect dataset $D = \{(x_h, \bar{r})\}$ of size n_{test} where $x_h \sim p$, $a_{h:H} \sim \hat{\pi}_{h:H}$, $\bar{r} = \sum_{h'=h}^{H} r_{h'}$;
  for $q \in$ LEARNED(h) **do**
    Collect dataset $D'_q = \{(x_h, \bar{r})\}$ of size n_{test} where $x_h \sim q$, $a_{h:H} \sim \hat{\pi}_{h:H}$, $\bar{r} = \sum_{h'=h}^{H} r_{h'}$;
    Solve

$$V_{opt} = \max_{g\in\mathcal{G}} \mathbb{E}_D[g(x_h)] \text{ s.t. } \mathbb{E}_{D'_q}[g(x_h) - \bar{r}] \le \phi_h + 2\tau_{val} \tag{6.26}$$

    if $V_{opt} \le \mathbb{E}_{D'_q}[\bar{r}] + \phi_h + 4\tau_{val}$ *and* $\mathbb{E}_D[\bar{r}] \ge \mathbb{E}_{D'_q}[\bar{r}] - 2\tau_{val}$ **then**
      return *true*;
  return *false*;
Function `dfslearn` (p)
  Let $h = |p| - 1$ the current level;
  if *not_called_from_Line_28* *and* `TestLearned` (p, h) **then**
    Add p to PRUNED(h);
    return ;
  for $a \in \mathcal{A}$ **do**
    `dfslearn` $(p \circ a)$;
  Add p to LEARNED(h);
  for $q \in$ LEARNED(h) **do**
    Collect dataset $D_q = \{(x_h, a_h, \bar{r})\}$ of size n_{train} where $x_h \sim q$, $a_h \sim$ Unif, $a_{h+1:H} \sim \hat{\pi}_{h+1:H}$, $\bar{r} = \sum_{h'=h}^{H} r_{h'}$;
    $\hat{V}(q) \leftarrow \max_{\pi\in\Pi} \mathbb{E}_{D_q}[K\mathbf{1}\{a_h = \pi(x_h)\}\bar{r}]$;
  Update $\hat{\pi}_h$ to be any policy satisfying

$$\forall q \in \text{LEARNED}(h) \quad \hat{\mathbb{E}}_{D_q}[K\mathbf{1}\{a_h = \pi(x_h)\}\bar{r}] \ge \hat{V}(q) - 2\tau_{pol}$$

  for $q \in$ PRUNED(h) **do**
    if `TestLearned`$(q, h) = false$ **then**
      remove q from PRUNED(h);
      `dfslearn` (q) ;
  return ;

Definition 115 (Deviation Bounds)**.** *We say the deviation bound holds for a data set of n_{train} observations sampled from q in Line 23 during a call to* `dfslearn` *if for all $\pi \in \Pi_h$*

$$|\hat{\mathbf{E}}_{D_q}[K\mathbf{1}\{a = \pi(x)\}\bar{r}] - \mathbb{E}_{q,\hat{\pi}_{h+1:H}}[K\mathbf{1}\{a_h = \pi(x_h)\}\bar{r}]| \leq \tau_{pol},$$

where we use $\mathbb{E}_{q,\hat{\pi}_{h+1:H}}[\cdot]$ as shorthand for $\mathbb{E}[\cdot|s_h = s, a_h \sim \mathit{Uniform}(K), a_{h+1:H} \sim \hat{\pi}_{h+1:H}]$ with s being the state reached by p and $\hat{\pi}_{h+1:H}$ being the current policy when the data set was collected. We say the deviation bound holds for a data set of n_{test} observations sampled in Line 8 during a call to `TestLearned` *if for all $g \in \mathcal{G}_h$:*

$$|\hat{\mathbf{E}}_D[g(x_h)] - \mathbb{E}_p[g(x_h)]| \leq \tau_{val}, \qquad |\hat{\mathbf{E}}_D[\bar{r}] - V^{\hat{\pi}_{h+1:H}}(p)| \leq \tau_{val}.$$

We say the deviation bound holds for a data set of n_{test} observations sampled in Line 10 during a call to `TestLearned` *if for all $g \in \mathcal{G}_h$:*

$$|\hat{\mathbf{E}}_{D'_q}[g(x_h)] - \mathbb{E}_q[g(x_h)]| \leq \tau_{val}, \qquad |\hat{\mathbf{E}}_{D'_q}[\bar{r}] - V^{\hat{\pi}_{h+1:H}}(q)| \leq \tau_{val}.$$

Learning Values using Depth First Search. We first show that if the current policy is close to optimal for all learned states, then the policy is also good on all states for which `TestLearned` returns true.

Lemma 116 (Policy on Tested States)**.** *Consider a call of* `TestLearned` *at path p and level h and assume the deviation bounds of Definition 115 hold for all data sets collected during this and all prior calls. Assume further that $\hat{\pi}_{h:H}$ satisfies $V^{\hat{\pi}_{h:H}}(q) \geq V^\star(q) - \phi_h$ for all $q \in$ LEARNED(h). Then $g^\star$ is always feasible for the program in Equation (6.26) and if* `TestLearned` *returns true, then the current policy $\hat{\pi}_{h:H}$ is near optimal for p, that is $V^{\hat{\pi}_{h:H}}(p) \geq V^\star(p) - \phi_h - 8\tau_{val}$.*

Proof. The optimal value function $g^\star$ is always feasible since

$$\hat{\mathbf{E}}_{D'}[g^\star(x) - \bar{r}] \leq V^\star(q) - V^{\hat{\pi}_{h:H}}(q) + 2\tau_{val} \leq \phi_h + 2\tau_{val}.$$

Here, we first used the deviation bounds and then the assumption about the performance of the current policy on learned states. Therefore, $V_{opt} \geq \hat{\mathbf{E}}_D[g^\star(x)] \geq V^\star(p) - \tau_{val}$ cannot underestimate the optimal value of p by much. Consider finally the performance of the current policy on p if `TestLearned` returns true:

$$\begin{aligned} V^{\hat{\pi}_{h:H}}(p) &\geq \hat{\mathbf{E}}_D[\bar{r}] - \tau_{val} \geq \hat{\mathbf{E}}_{D'}[\bar{r}] - 3\tau_{val} \\ &\geq V_{opt} - 3\tau_{val} - 4\tau_{val} - \phi_h \geq V^\star(p) - 8\tau_{val} - \phi_h. \end{aligned}$$

Here, the first inequality follows from the deviation bounds, the second from the second condition of the if-clause in `TestLearned`, the third from the first condition of the if-clause and finally the fact that V_{opt} is an accurate estimate of the optimal value of p. □

Thus, the `TestLearned` routine can identify paths where the current policy is close to optimal if this policy's performance on all learned states is good. Next, we prove that the policy has near-optimal performance on all the learned states.

Lemma 117 (Global policy fitting)**.** *Consider a call of* `dfslearn` *(p) at level h and assume the deviation bounds hold for all data sets collected during this and all prior calls. Then the program in Line 25 is always feasible and after executing that line, we have $\forall q \in$ LEARNED(h),*

$$Q^{\hat{\pi}_{h+1:H}}(q, \hat{\pi}_h) \geq Q^{\hat{\pi}_{h+1:H}}(q, \star) - 3\tau_{pol},$$

where $\star$ is a shorthand for $\pi^\star_{\hat\pi_{h+1:H}}$, the policy defined in Assumption 90 w.r.t. the current policy $\hat\pi_{h+1:H}$. This implies that if all children nodes q' of q satisfy $V^{\hat\pi_{h+1:H}}(q') \geq V^\star(q') - \beta$ for some β, then $V^{\hat\pi_{h:H}}(q) \geq V^\star(q) - \beta - 3\tau_{pol}$.

Proof. We prove feasibility by showing that $\pi^\star_{\hat\pi_{h+1:H}}$ is always feasible. For each $q \in \textsc{Learned}(h)$, let $\hat\pi_h^q$ denote the policy that achieves the maximum in computing $\hat V(q)$. Then

$$\hat{\mathbf{E}}_{D_q}[K\mathbf{1}\{a_h = \pi^\star_{\hat\pi_{h+1:H}}(x_h)\}\bar r] \geq Q^{\hat\pi_{h+1:H}}(q,\star) - \tau_{pol} \geq Q^{\hat\pi_{h+1:H}}(q,\hat\pi_h^q) - \tau_{pol} \geq \hat V(q) - 2\tau_{pol}.$$

The first and last inequality are due to the deviation bounds and the second inequality follows from definition of $\pi^\star_{\hat\pi_{h+1:H}}$. This proves the feasibility. Now, using this inequality along with $\hat V(q) = \max_{\pi\in\Pi} \mathbb{E}_{D_q}[K\mathbf{1}\{a_h = \pi(x_h)\}\bar r]$, we can relate $\hat V(q)$ and $Q^{\hat\pi_{h+1:H}}(q,\star)$:

$$\hat V(q) \geq \hat{\mathbf{E}}_{D_q}[K\mathbf{1}\{a_h = \pi^\star_{\hat\pi_{h+1:H}}(x_h)\}\bar r] \geq Q^{\hat\pi_{h+1:H}}(q,\star) - \tau_{pol}.$$

Finally, since $\hat\pi_h$ is feasible in Line 25,

$$V^{\hat\pi_{h:H}}(q) = Q^{\hat\pi_{h+1:H}}(q,\hat\pi_h) \geq \hat V(q) - 2\tau_{pol} \geq Q^{\hat\pi_{h+1:H}}(q,\star) - 3\tau_{pol}.$$

To prove the implication, consider the case where for $a \in \mathcal{A}$, all paths $q' = q \circ a$ satisfy $V^{\hat\pi_{h+1:H}}(q') \geq V^\star(q') - \beta$. Then

$$\begin{aligned} V^\star(q) - V^{\hat\pi_{h:H}}(q) &\leq V^\star(q) - Q^{\hat\pi_{h+1:H}}(q,\star) + 3\tau_{pol} \leq V^\star(q) - Q^{\hat\pi_{h+1:H}}(q,\pi^\star) + 3\tau_{pol} \\ &= \mathbb{E}_{q'\sim q\circ\pi^\star}[V^\star(q') - V^{\hat\pi_{h+1:H}}(q')] + 3\tau_{pol} \leq \beta + 3\tau_{pol}, \end{aligned}$$

where we first used the inequality from above and then the fact that $\pi^\star_{\hat\pi_{h+1:H}}$ is optimal given the fixed policy $\hat\pi_{h+1:H}$. The equality holds since both $V^\star(q) - Q^{\hat\pi_{h+1:H}}(q,\pi^\star)$ both are with respect to $a_h \sim \pi^\star_h$ and finally we apply the assumption. □

We are now ready to apply both lemmas above recursively to control the performance of the current policy on all learned and pruned paths:

Lemma 118. *Set $\phi_h = (H-h+1)(8\tau_{val} + 3\tau_{pol})$ and consider a call to* `dfslearn`(p) *at level h. Assume the deviation bounds hold for all data sets collected until this call terminates. Then for all $p \in \textsc{Learned}(h)$, the current policy satisfies*

$$V^{\hat\pi_{h:H}}(p) \geq V^\star(p) - \phi_h$$

at all times except between adding a new path and updating the policy. Further, for all $p \in \textsc{Pruned}(h)$ the currently policy satisfies

$$V^{\hat\pi_{h:H}}(p) \geq V^\star(p) - \phi_h - 8\tau_{val}$$

whenever `dfslearn` *returns from level h to $h-1$.*

Proof. We prove the claim inductively. For $h = H+1$ the statement is trivially true since there are no actions left to take and therefore the value of all policies is identical 0 by definition.

Assume now the statement holds for $h+1$. We first study the learned states. To that end, consider a call to `dfslearn`(p) at level h that does not terminate in Line 18 and performs a policy update. Since `dfslearn` is called recursively for all $p \circ a$ with $a \in \mathcal{A}$ before p is added to $\textsc{Learned}(h)$ and every path

that `dfslearn` is called with either makes that path learned or pruned, all successor states of p are in $\textsc{Pruned}(h)$ or $\textsc{Learned}(h)$ when p is added. Since the statement holds for $h+1$, for all successor paths p' we have $V^{\hat{\pi}_{h+1:H}}(p') \geq V^\star(p') - \phi_{h+1} - 8\tau_{val}$. We can apply Lemma 117 and obtain that after changing $\hat{\pi}_h$, it holds that for all $q \in \textsc{Learned}(h)$ $V^{\hat{\pi}_{h:H}}(q) \geq V^\star(q) - \phi_{h+1} - 8\tau_{val} - 3\tau_{pol} = V^\star(q) - \phi_h$. Since that is the only place where the policy changes or a state is added to $\textsc{Learned}(h)$, this proves the first part of the statement for level h.

For the second part, we can apply Lemma 116 which claims that for all paths q for which $\texttt{TestLearned}(q,h)$ returns true, it holds that $V^{\hat{\pi}_{h:H}}(q) \geq V^\star(q) - \phi_h - 8\tau_{val}$. It remains to show that whenever `dfslearn` returns to a higher level, for all paths $q \in \textsc{Pruned}(h)$, $\texttt{TestLearned}(q,h)$ evaluates to true. This condition can only be violated when we add a new state to $\textsc{Pruned}(h)$ or change the policy $\hat{\pi}_{h:H}$.

For the later case, we explicitly check the condition in Lines 26-28 after we change the policy before returning. Therefore `dfslearn` can only return after Line 28 without further recursive calls to `dfslearn` if `TestLearned` evaluated to true for all $q \in \textsc{Pruned}(h)$. The statement is therefore true if the algorithm returns after Line 28. Further, a path can only be added to $\textsc{Pruned}(h)$ after we explicitly checked that `TestLearned` evaluates true for it before we return in Line 18. Hence, the second part of the statement also holds for h which completes the proof. □

Lemma 119 (Termination)**.** *Assume the deviation bounds hold for all Data sets collected during the first $T_{\max} = 3M^2HK$ calls of* `dfslearn` *and* `TestLearned`*. The algorithm terminates during these calls and at all times for all $h \in [H]$ it holds $|\textsc{Learned}(h)| \leq M$. Moreover, the number of paths that have ever been added to $\textsc{Pruned}(h)$ (that is, counting those removed in Line 26) is at most KM.*

Proof. Consider a call to $\texttt{TestLearned}(p,h)$ where p leads to the same state as a $q \in \textsc{Learned}(h)$. Assume the deviation bounds hold for all data sets collected during this call and before, and we can show that `TestLearned` must evaluate to true: Using Lemma 118 we get that on all learned paths p it holds that $V^{\hat{\pi}_{h:H}}(p) \geq V^\star(p) - \phi_h$. Therefore, $g^\star$ is feasible in (6.26) since $\hat{\mathbf{E}}_{D'}[g^\star(x) - \bar{r}] \leq V^\star(q) - V^{\hat{\pi}_{h:H}}(q) + 2\tau_{val} \leq \phi_h + 2\tau_{val}$. This allows us to relate V_{opt} to the optimal value as

$$V_{opt} \geq \hat{\mathbf{E}}_D[g^\star(x)] \geq V^\star(p) - \tau_{val}.$$

It further holds that

$$\hat{\mathbf{E}}_D[\bar{r}] \geq V^{\hat{\pi}_{h:H}}(p) - \tau_{val} = V^{\hat{\pi}_{h:H}}(q) - \tau_{val} \geq \hat{\mathbf{E}}_{D'}[\bar{r}] - 2\tau_{val}.$$

and so the second condition in the if-clause holds. For the first condition, let $\hat{g}$ be the function that achieves the maximum in the computation of V_{opt}. Then

$$\begin{aligned} V_{opt} &= \hat{\mathbf{E}}_D[\hat{g}(x_h)] \leq \mathbb{E}_s[\hat{g}(x_h)] + \tau_{val} \leq \hat{\mathbf{E}}_{D'_q}[\hat{g}(x_h)] + 2\tau_{val} \\ &\leq \hat{\mathbf{E}}_{D'_q}[\bar{r}] + \phi_h + 2\tau_{val} + 2\tau_{val} = \hat{\mathbf{E}}_{D'_q}[\bar{r}] + \phi_h + 4\tau_{val}. \end{aligned}$$

Then the first condition is also true and `TestLearned` returns true. Therefore, `TestLearned` evaluates to true for all paths that reach the same state as a learned path. As a consequence, if `dfslearn` is called with such a path it returns in Line 18. Furthermore, as long as all deviation bounds hold, the number of learned paths per level is bounded by $|\textsc{Learned}(h)| \leq M$.

We next show that the number of paths that have ever appeared in $\textsc{Pruned}(h)$ is at most KM. This is true since there are at most KM recursive calls to `dfslearn` at level h from level $h-1$ and only during those calls a path can be added to $\textsc{Pruned}(h)$ that has not been in $\textsc{Pruned}(h)$ before.

Assume the deviation bounds hold for all data sets collected during the first $T_{\max}$ calls of `dfslearn`. There can be at most MH calls of `dfslearn` in which a path is learned. Since the recursive call in

Line 28 always learns a new state at the next level, the only way to grow $\textsc{Pruned}(h)$ is via the recursive call on Line 20, which occurs at most MKH times. Therefore the algorithm terminates after at most $MH + MHK$ calls to `dfslearn`. Each of these calls can make at most 1 call to `TestLearned` unless it learns a new state and calls `TestLearned` up to $|\textsc{Pruned}(h)| + 1 \leq MK + 1$ times. Therefore, the total number of calls to `TestLearned` is bounded by $MH(MK + 1) + MHK$. The lemma follows by noticing that both numbers of calls are bounded by $T_{\max}$. □

Lemma 120. *Let $\mathcal{E}$ be the event that the deviation bounds in Definition 115 hold for all data sets collected during Algorithm 14. Set n_{train} and n_{test} such that*

$$n_{train} \geq \frac{16K}{\tau_{pol}^2} \log\left(\frac{16T_{\max}M|\Pi||\mathcal{G}|}{\delta}\right)$$
$$n_{test} \geq \frac{1}{2\tau_{val}^2} \log\left(\frac{16T_{\max}M|\Pi||\mathcal{G}|}{\delta}\right)$$

Then $\mathbb{P}(\bar{\mathcal{E}}) \leq \delta$.

Proof. Consider a single data set D_q collected in `dfslearn`(p) at level h where p is learned for $q \in \textsc{Learned}(h)$. For the random variable $K\mathbf{1}\{\pi(x_h) = a_h\}\bar{r}$, since a_h is chosen uniformly at random, it is not hard to see that both the variance and the range are upper-bounded by $2K$ (see for example Lemma 14 by Jiang, Krishnamurthy, et al. (2017)). As such, Bernstein's inequality and a union bound over all $\pi \in \Pi_h$ gives that with probability $1 - \delta'$,

$$|\hat{\mathbf{E}}_{D_q}[K\mathbf{1}\{a = \pi(x)\}\bar{r}] - \mathbb{E}_{q,\hat{\pi}_{h+1:H}}[K\mathbf{1}\{a_h = \pi(x_h)\}\bar{r}]| \leq \sqrt{\frac{4K\log(2|\Pi|/\delta')}{n_{\text{train}}}} + \frac{4K}{3n_{\text{train}}}\log(2|\Pi|/\delta').$$

Consider a single data set D collected in `TestLearned`(p, h). By Hoeffding's inequality and a union bound, with probability $1 - \delta'$, for all $g \in \mathcal{G}_h$

$$|\hat{\mathbf{E}}_D[g(x_h)] - \mathbb{E}_p[g(x_h)]| \leq \sqrt{\frac{\log(2|\mathcal{G}|/\delta')}{2n_{\text{test}}}}$$

Analogously, for a data set D'_q collected during `TestLearned`(p, h) with $q \in \textsc{Learned}(q)$, we have with probability at least $1 - \delta'$ that

$$|\hat{\mathbf{E}}_{D'_q}[g(x_h)] - \mathbb{E}_q[g(x_h)]| \leq \sqrt{\frac{\log(2|\mathcal{G}|/\delta')}{2n_{\text{test}}}}$$

Further, again by Hoeffding's inequality and a union bound we get that for a single data set D collected in `TestLearned`(p, h) and a single data set D'_q collected during `TestLearned`(p, h) with $q \in \textsc{Learned}(q)$ with probability at least $1 - \delta'$ it holds

$$|\hat{\mathbf{E}}_{D'_q}[\bar{r}] - V^{\hat{\pi}_{h+1:H}}(q)| \leq \sqrt{\frac{\log(4/\delta')}{2n_{\text{test}}}} \quad \text{and}$$
$$|\hat{\mathbf{E}}_D[\bar{r}] - V^{\hat{\pi}_{h+1:H}}(p)| \leq \sqrt{\frac{\log(4/\delta')}{2n_{\text{test}}}}.$$

Combining all these bounds with a union bound and using $\delta' = \frac{\delta}{4MT_{\max}}$, we get that the deviation bounds hold for the first $MT_{\max}$ data sets of the form D'_q and D_q and D with probability at least $1 - \delta$. Using Lemma 119, this is sufficient to show that $\mathbb{P}(\bar{\mathcal{E}}) \leq \delta$. □

Proof of Theorem 114. We now have all parts to complete the proof of Theorem 83.

Proof. For the calculation, we instantiate all the parameters as

$$\tau_{pol} = \frac{\epsilon}{6H}, \quad \tau_{val} = \frac{\epsilon}{6H}, \quad \phi_h = (H-h+1)(8\tau_{val} + 3\tau_{pol}), \quad T_{\max} = 3M^2HK,$$
$$n_{\text{test}} = \frac{\log(16T_{\max}M|\Pi||\mathcal{G}|/\delta)}{2\tau_{val}^2}, \quad n_{\text{train}} = \frac{16K\log(16T_{\max}M|\Pi||\mathcal{G}|/\delta)}{\tau_{pol}^2}.$$

These settings suffice to apply all of the above lemmas for these algorithms and therefore with these settings the algorithm outputs a policy that is at most ϵ-suboptimal, except with probability δ. For the sample complexity, since $T_{\max}$ is an upper bound on the number of calls to `TestLearned` and at most M states are learned per level $h \in [H]$, we collect a total of at most the following number of episodes:

$$\begin{aligned}
&(1+M)T_{\max}n_{\text{test}} + M^2Hn_{\text{train}} \\
&= \tilde{O}\left(\frac{T_{\max}MH^2}{\epsilon^2}\log(|\Pi||\mathcal{G}|/\delta) + \frac{M^2H^3K}{\epsilon^2}\log(|\Pi||\mathcal{G}|/\delta)\right) \\
&= \tilde{O}\left(\frac{M^3KH^3}{\epsilon^2}\log(|\Pi||\mathcal{G}|/\delta)\right).
\end{aligned}$$

□

6.10 Oracle-Inefficiency of OLIVE

As explained in Section 6.5 Theorem 84 follows directly from Theorem 85 and Proposition 86 by proof by contradiction with $P \neq NP$. In the following two sections, we first prove Proposition 86 and then Theorem 85.

6.10.1 Proof for Polynomial Time of Oracles

Proof of Proposition 86. We prove the claim for each oracle separately

1. **CSC-Oracle:** For tabular functions, the objective can be decomposed as

$$n^{-1}\sum_{i=1}^{n} c^{(i)}(\pi(x^{(i)})) = \sum_{x\in\mathcal{X}} n^{-1}\sum_{i=1}^{n}\mathbf{1}\{x = x^{(i)}\}c^{(i)}(\pi(x)).$$

   Each of the $|\mathcal{X}|$ terms only depend on $\pi(x)$ but not on any action chosen for different observations. Hence, since $\Pi = (\mathcal{X} \to \mathcal{A}) \triangleq \mathcal{A}^{|\mathcal{X}|}$, the action chosen by $\hat{\pi} = n^{-1}\operatorname{argmin}_{\pi\in\Pi}\sum_{i=1}^{n} c^{(i)}(\pi(x^{(i)}))$ for $x \in \mathcal{X}$ is $\operatorname{argmin}_{a\in\mathcal{A}}\sum_{i=1}^{n}\mathbf{1}\{x = x^{(i)}\}c^{(i)}(\pi(x))$. To compute $\hat{\pi}$, we first compute for each x the total cost vector $\sum_{i=1}^{n}\mathbf{1}\{x = x^{(i)}\}c^{(i)}(\pi(x))$ and then pick the smallest entry as the action for $\hat{\pi}(x)$. Per x, this takes $O(Kn)$ operations and therefore, the total runtime for this oracle is $O(nK|\mathcal{X}|)$.
2. **LS-Oracle:** Similarly to the CSC objective, the least-squares objective can be decomposed as

$$\sum_{i=1}^{n}(v^{(i)} - g(x^{(i)}))^2 = \sum_{x\in\mathcal{X}}\sum_{i=1}^{n}\mathbf{1}\{x = x^{(i)}\}(v^{(i)} - g(x))^2$$

   and therefore $\hat{g} = \operatorname{argmin} g \in \mathcal{G}\sum_{i=1}^{n}(v^{(i)} - g(x^{(i)}))^2$ can be computed for each observation separately. A minimizer per observation x of $\sum_{i=1}^{n}\mathbf{1}\{x = x^{(i)}\}(v^{(i)} - g(x))^2$ is $\hat{g}(x) = \frac{\sum_{i=1}^{n}\mathbf{1}\{x=x^{(i)}\}v^{(i)}}{\sum_{i=1}^{n}\mathbf{1}\{x=x^{(i)}\}}$, where we set $\hat{g}(x)$ arbitrarily if $\sum_{i=1}^{n}\mathbf{1}\{x = x^{(i)}\} = 0$. This can be computed with $O(n)$ operations and therefore the total runtime of the LS-oracle is $O(|\mathcal{X}|n)$.

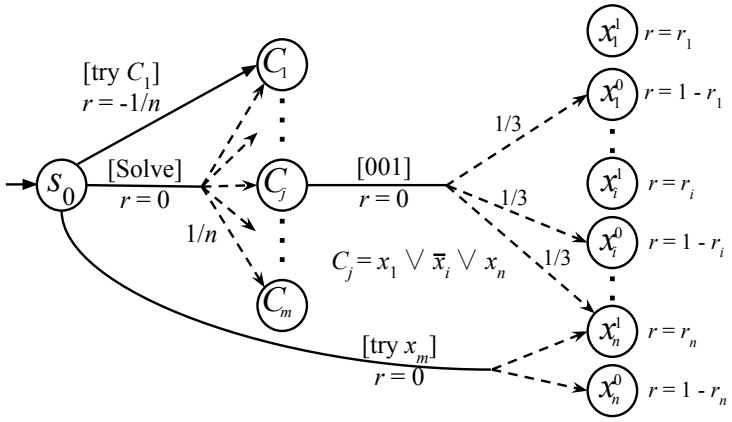

Figure 6.2: Family of MDPs that are determined up to terminal rewards $r_1, \ldots, r_n \in [0, 1]$. Finding the optimal value of the most optimistic MDP in this family solves the encoded 3-SAT instance. Solid arrows represent actions and dashed arrows represent random transitions.

3. **LP-Oracle:** We parameterize $g \in \mathcal{G}$ by vectors $\theta \in \mathbb{R}^{|\mathcal{X}|}$ where each the value of g for each $x \in \mathcal{X}$ is associated with a particular entry θ_x of θ. Then the LP problem reduces to a standard linear program in $\mathbb{R}^{|\mathcal{X}|}$. Khachiyan (1980) and Grötschel, Lovász, and Schrijver (1981) have shown using the ellipsoid method, these problems can be solved approximately in polynomial time. Note that the initial ellipsoid can be set to any ellipsoid containing $[0, 1]^{|\mathcal{X}|}$ due to the normalization of rewards. Further, the volume of the smallest ellipsoid can be upper bounded by a polynomial in ϵ_{feas} using the fact that we only require a solution that is feasible up to ϵ_{feas} and applying the ellipsoid method to the extended polytope with all constraints relaxed by ϵ_{feas}. □

6.10.2 `OLIVE` is NP-hard in tabular MDPs

Instead of showing Theorem 85 directly, we first show the following simpler version:

Theorem 121. *Let $\mathcal{P}$ denote the family of problems of the form (6.2), parameterized by $(\mathcal{X}, \mathcal{A}, D_0, \mathcal{D})$ with implicit $\mathcal{G} = (\mathcal{X} \to [0, 1])$ and $\Pi = (\mathcal{X} \to \mathcal{A})$ (i.e., the tabular value-function and policy classes) and with $\phi = 0$. $\mathcal{P}$ is NP-hard.*

Some remarks are in order about this statement

1. Our proof actually shows that it is NP-hard to find an ϵ-approximate solution to these optimization problems, for polynomially small ϵ.
2. The two theorems differ in whether the data sets ($D_i \in \mathcal{D}$) are chosen adversarially (Theorem 121), or induced naturally from an actual run of `OLIVE` (Theorem 85). Therefore, Theorem 85 is strictly stronger.
3. At a high level, these results imply that `OLIVE` in general must solve NP-hard optimization problems, presenting a barrier for computational tractability.
4. These results also hold with imperfect expectations and polynomially small ϕ.
5. We use the $(\mathcal{G}, \Pi)$ representation here but similar results hold with $\mathcal{F}$ representation (i.e., approximating the Q-function; see Theorems 122 and 123).

For intuition we first sketch the proof of Theorem 121. The complete proof follows below.

Proof Sketch of Theorem 121. We reduce from 3-SAT. Let ψ be a 3-SAT formula on n variables $x_1, \ldots, x_n$ with m clauses $c_1, \ldots, c_m$. We construct a family of MDPs as shown in Figure 6.2 that encodes the 3-SAT problem for this formula as follows: For each variable x_i there are two terminal states x_i^1 and x_i^0 corresponding to the Boolean assignment to the variable. For each variable, the reward in either x_i^1 or x_i^0 is 1 and 0 in the other. The family of MDPs contains all possible combinations of such terminal rewards. There is also one state per clause c_j and one start state s_0. From each clause, there are 7 actions, one for each binary string of length 3 except "000." These actions all receive zero instantaneous reward. For clause $c_\ell = x_i \vee \bar{x}_j \vee \bar{x}_k$, the action "$b_1 b_2 b_3$" transitions to states $x_i^{b_1}, x_j^{1-b_2}$, or $x_k^{1-b_3}$, each with probability $1/3$. The intuition is that the action describes which literals evaluate to true for this clause. From the start state, there are $n + m + 1$ actions. For each variable x_i, there is a [try x_i] action that transitions uniformly to x_i^0, x_i^1 and receives 0 instantaneous reward. For each clause c_j there is a [try c_j] action that transitions deterministically to the state for clause c_j, but receives reward $-1/n$. And finally there is a [solve] action that transitions to a clause state uniformly at random.

For each x_i, we introduce a constraint into Problem (6.2) corresponding to the [try x_i] action. These constraints impose that the optimal $\hat{g} \in \mathcal{G}$ satisfies $\forall i \in [m] : \hat{g}(x_i^0) + \hat{g}(x_i^1) = 1$. We also introduce constraints for the [try c_j] actions and from s_0. Recall that values must be in $[0, 1]$.

With these constraints, if the 3-SAT formula has a satisfying assignment, then the optimal value from the start state is 1, and it is not hard to see that there exists function $\hat{g} \in \mathcal{G}$ that achieves this optimal value, while satisfying all constraints with a $\hat{\pi} \in \Pi$. Conversely, if the value of the start date is 1, we claim that the 3-SAT formula is satisfiable. In more detail, the policy must choose the [solve] action, and the value function must predict that each clause state has value 1, then the literal constraints enforce that exactly one of x_i^0, x_i^1 has value 1 for each i. Thus the optimistic value function encodes a satisfying assignment, completing the reduction. □

Proof of Theorem 121

In this section, we prove that the optimization problem solved by `OLIVE` is NP-hard. The proofs rely on the fact that `OLIVE` only adds a constraint for a single time step h that has high average Bellman error. However, using an extended construction, one can show similar statements for a version of `OLIVE` that adds constraints for all time steps if there is high average Bellman error in any time step.

For notational simplicity, we do not prove Theorem 121 and Theorem 85 directly, but versions of these statements below with a tabular Q-function representation $\mathcal{F}$ instead of the $(\mathcal{G}, \Pi)$ version presented in the paper. For this formulation, `OLIVE` picks the policy for the next round as the greedy policy $\pi_{\hat{f}_k}$ of the Q-function that maximizes

$$\begin{aligned} &\hat{f}_k = \operatorname*{argmax}_{f \in \mathcal{F}} \hat{\mathbf{E}}_{D_0}[f(x, \pi_f(x))] \\ &\text{s.t. } \forall\, D_i \in \mathcal{D} : \\ &|\hat{\mathbf{E}}_{D_i}[\mathbf{1}\{a = \pi_f(x)\}(f(x,a) - r - f(x', \pi_f(x')))]| \leq \phi. \end{aligned} \tag{6.27}$$

This proof naturally extends to the $(\mathcal{G}, \Pi)$ representation: note that `OLIVE` runs in a completely equivalent way if it takes a set of (g, π) pairs induced by $\mathcal{F}$ as inputs, i.e., $\{(x \mapsto f(x, \pi_f(x)), x \mapsto \pi_f(x)) : f \in \mathcal{F}\}$ (Jiang, Krishnamurthy, et al., 2017, see Appendix A.2,). When $\mathcal{F}$ is the tabular Q-function class, it is easy to verify that the induced set is the same as $\mathcal{G} \times \Pi$ where $\mathcal{G}$ and Π are the tabular value-function / policy classes respectively. Therefore, the proof for Theorem 121 just requires a simple substitution where $f(x, \pi_f(x))$ is replaced by $g(x)$ and $\pi_f(x)$ is replaced by π.

We first prove the simpler NP-hardness claim.

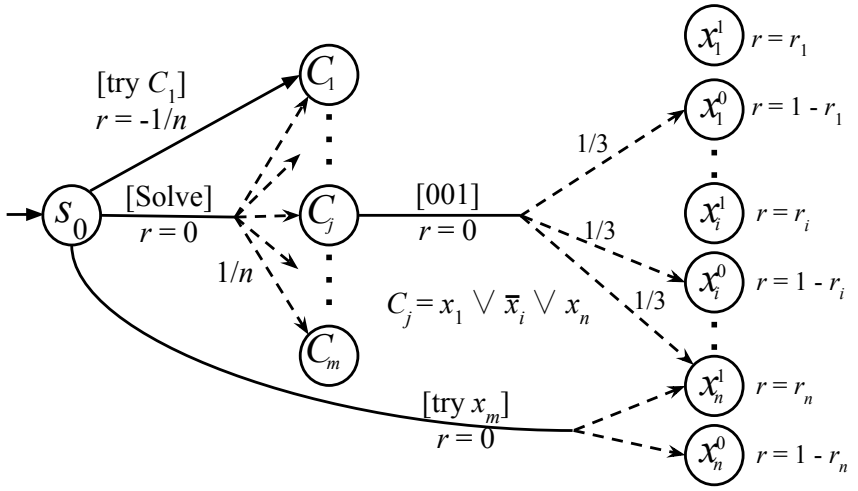

Figure 6.3: Family of MDPs $\mathcal{M}$ for a specific instance of a 3-SAT problem.

Theorem 122 ($\mathcal{F}$-Version of Theorem 121)**.** *Let $\mathcal{P}$ denote the family of problems of the form (6.27), parameterized by $(\mathcal{X}, \mathcal{A}, D_0, \mathcal{D})$ with implicit $\mathcal{F} = (\mathcal{X} \times \mathcal{A} \to [0,1])$ (i.e., the tabular Q-function class) and with $\phi = 0$. $\mathcal{P}$ is NP-hard.*

Proof. For the ease of presentation, we show the statement for $\mathcal{F} = (\mathcal{X} \times \mathcal{A} \to [-1,1])$ and all values scaled to be in $[-1,1]$. By linearly transforming all rewards accordingly, one obtains a proof for the statement with all values in $[0,1]$.

We demonstrate a reduction from 3-SAT. Recall that an instance of 3-SAT is a Boolean formula ψ on n variables can be described by a list of clauses $C_1, \dots C_m$ each containing at 3 literals (a variable x_i or its negation $\bar{x}_i$), e.g. $C_1 = (\bar{x}_2 \vee x_3 \vee \bar{x}_5)$. As notation let $o^1_{j,i}$ for $i \in \{1,2,3\}$ denote the i^{th} literal in the j^{th} clause and $o^0_{j,i}$ its negation (e.g. $o^1_{1,3} = \bar{x}_5$ and $o^0_{1,3} = x_5$). Given a 3-SAT instance with m clauses $C_{1:m}$ and n variables $x_{1:n}$, we define a class of finite episodic MDPs $\mathcal{M}$. This class contains (among others) 2^n MDPs that correspond each to an assignment of Boolean values to $x_{1:n}$.

The proof proceeds as follows: First we describe the construction of this class of MDPs. Then we will demonstrate a set of constraints for the `OLIVE` program. Importantly, these constraints do not distinguish between the 2^n MDPs in the class $\mathcal{M}$ corresponding to the binary assignments to the variables $x_{1:n}$, so the optimistic planning step in `OLIVE` needs to reason about all possible assignments. Finally, we show that with the function class $\mathcal{F} = (\mathcal{X} \times \mathcal{A}) \to [-1,1]$, the solution to the optimization problem (6.27) determines whether ψ is satisfiable or not.

For simplicity, the MDPs in $\mathcal{M}$ have different actions available in different states and rewards are in $[-1,1]$ instead of the usual $[0,1]$. We can however find equivalent MDPs that satisfy the formal requirements of `OLIVE`.

MDP structure. Let ψ be the 3-SAT instance with variables $x_{1:n}$ and clauses $C_{1:m}$. The state space for MDPs in $\mathcal{M}$ consists of $m + 2n + 1$ states, two for each variable, one for each clause, and one additional starting state. For each variable x_i, there are two states x^0_i, x^1_i corresponding to the variable and its negation. Each clause C_j has a state C_j, and the starting state is denoted s_0.

The transitions are as follows: The states x^0_i, x^1_i corresponding to the literals are terminal, with just a single action. The class $\mathcal{M}$ differs only in how it assigns rewards to these terminal states. Specifically let

$y \in \{0,1\}^n$ be a binary vector, then there is an MDP $M_y \in \mathcal{M}$ where for all $i \in [n]$ the reward for literal $x_i^{y_i} = 1$ and $x_i^{1-y_i} = 0$. Specifically, all MDPs in $\mathcal{M}$ have values that satisfy $V(x_i^1) + V(x_i^0) = 1$ for all $i \in [n]$.

Each clause state C_j has 7 actions, indexed by $b \in \{0,1\}^3 \setminus \{\text{“000”}\}$, each corresponding to an assignment of the variables that would satisfy the clause. Taking an action b transitions the agent to three literal states with equal probability $1/3$ and the agent receives no immediate reward. Which literals is determined by the clause. Assume the clause consists of $C_t = (\bar{x}_i \vee x_j \vee \bar{x}_k)$. Then

$$\begin{aligned}
\mathbb{P}(x_i^1|c_t,b) =& \frac{1}{3}\mathbf{1}\{b_1 = 0\}, \quad \mathbb{P}(x_i^0|c_t,b) = \frac{1}{3}\mathbf{1}\{b_1 = 1\} \\
\mathbb{P}(x_j^1|c_t,b) =& \frac{1}{3}\mathbf{1}\{b_2 = 1\}, \quad \mathbb{P}(x_j^0|c_t,b) = \frac{1}{3}\mathbf{1}\{b_2 = 0\} \\
\mathbb{P}(x_k^1|c_t,b) =& \frac{1}{3}\mathbf{1}\{b_3 = 0\}, \quad \mathbb{P}(x_k^0|c_t,b) = \frac{1}{3}\mathbf{1}\{b_3 = 1\}.
\end{aligned}$$

For example, taking action 011 in clause state $C_1 = (\bar{x}_2 \vee x_3 \vee \bar{x}_5)$ transitions with equal probability to x_2^1 (since the first component of the action is 0), x_3^1 (second component is 1) and x_5^0 (last component is 1).

The initial state has $n+m+1$ actions. The first set of actions are labeled `[try` x_i`]`, for $i \in [n]$. They receive zero instantaneous reward and transition uniform to x_i^1, x_i^0. The second set of actions are labeled `[try` C_j`]` (for $j \in [m]$), which receives $1/m$ instantaneous reward and transitions deterministically to c_j. Finally there is a `[solve]` action that transitions uniformly to the $\{C_j\}_{j=1}^m$ states and receives zero instantaneous reward.

`OLIVE` Constraints. We introduce constraints at the start state s_0, all of the constraint states C_j, and the distributions induced when taking the `[try` x_i`]` action. Since the literal states x_i^1, x_i^0 have no actions, we omit the second argument from the Q-functions f. We list these constraints in the following writing out the constraints for each optimal action that are implied by the indicator of the original constraints in Problem (6.27): From initial state:

$$\begin{aligned}
f(s_0, \texttt{[try } c_j\texttt{]}) &= \max_b f(c_1, b) - 1/m && \text{if } \pi_f(s_0) = \texttt{[try } c_j\texttt{]} \qquad (6.28)\\
f(s_0, \texttt{[solve]}) &= \frac{1}{m}\sum_{i=1}^m \max_b f(C_j, b) && \text{if } \pi_f(s_0) = \texttt{[solve]} \\
f(s_0, \texttt{[try } x_i\texttt{]}) &= \frac{f(x_i^0) + f(x_i^1)}{2} && \text{if } \pi_f(s_0) = \texttt{[try } x_i\texttt{]}
\end{aligned}$$

From clause j after `[try` C_j`]`:

$$f(C_j, b) = \frac{f(o_{j,1}^{b(1)}) + f(o_{j,2}^{b(2)}) + f(o_{j,3}^{b(3)})}{3} \qquad \text{if } \pi_f(C_j) = b$$

From variable i after `[try` x_i`]`:

$$\frac{f(x_i^1) + f(x_i^0)}{2} = \frac{1}{2} \qquad (6.29)$$

Note that all appearances of f on the LHS could be replaced by $f(\cdot, \pi_f(\cdot))$. There are other types of constraints involving literal states that could be imposed, specifically constraints of the form

$$\sum_{i=1}^m w_{2i-1} f(x_i^1) + w_{2i} f(x_i^0) = V \qquad (6.30)$$

for some V and $w \in \Delta([2m])$, which appears by first applying `[solve]` or `[try` C_j`]` and then various actions at the clause states to arrive at a distribution over the literal states. It is important here that constraints of this type are *not* included in the optimization problem, since it distinguishes elements of the family $\mathcal{M}$.

The Optimal Value. Consider the `OLIVE` optimization problem (6.27) on the family of MDPs $\mathcal{M}$ with constraints described above. Note that all MDPs in the family generate identical constraints, so formulating the optimization problem does not require determining whether ψ has a satisfying assignment or not.

Now, if ψ has a satisfying assignment, say $y^\star \in \{0,1\}^n$, then the MDP $M_{y^\star} \in \mathcal{M}$ has optimal value 1. Moreover since the function class $\mathcal{F}$ is entirely unconstrained, this function class can achieve this value, which is the solution to Problem (6.27). To see why $M_{y^\star}$ has optimal value 1, consider the policy that chooses the `[solve]` action and from each clause chooses the 3-bit string that transitions to the literal states that have value 1. Importantly, since ψ has a satisfying assignment, this must be true for one of the 7 actions.

Conversely, suppose that Problem (6.27), with all the constraints described above, has value 1. We argue that this implies ψ has a satisfying assignment. Let $\hat{f}, \hat{\pi}$ correspond to the Q-value and policy that achieve the optimal value in the program. First, due to the constraints on the `[try` x_i`]` distributions and the immediate negative rewards for taking `[try` C_j`]` actions, we must have $\hat{\pi}(s_0) =$ `[solve]` and $\hat{f}(s_0, \texttt{[solve]}) = 1$. The constraints on $\hat{f}$ now imply that for each clause C_j there exists a action $b_j \in \{0,1\}^3 \setminus \{000\}$ such that $\hat{f}(C_j, b_j) = 1$. Proceeding one level further, if b_j satisfies $\hat{f}(C_j, b_j) = 1$ then we must have that $\hat{f}(o_{j,k}^{b_j(k)}) = 1$ for all $k \in \{1,2,3\}$. And due to the boundedness conditions on $\hat{f}$ along with the constraint that $\hat{f}(x_i^0) + \hat{f}(x_i^0) = 1$, one of these values must be 1, while the other is zero. Therefore, for any variable that appears in some clause the corresponding literal states must have predicted value that is binary. Since the constraints corresponding to the clauses are all satisfied (or else we could not have value 1 at s_0), the predicted values at the literal states encodes a satisfying assignment to ψ. □

Proof of Theorem 85

After showing that Problem (6.27) is NP-hard when constraints are chosen adversarially, we extend this result to the class of problems encountered by running `OLIVE`. Again, we prove a version of the statement with $\mathcal{F}$ representation but the proof for Theorem 85 is completely analogous.

Theorem 123 ($\mathcal{F}$ Version of Theorem 85)**.** *Let $\mathcal{P}_{\texttt{OLIVE}}$ denote the family of problems of the form (6.27), parameterized by $(\mathcal{X}, \mathcal{A}, Env, t)$, which describes the optimization problem induced by running `OLIVE` in the MDP environment Env (with states $\mathcal{X}$, actions $\mathcal{A}$ and perfect evaluation of expectations) for t iterations with $\mathcal{F} = (\mathcal{X} \times \mathcal{A} \to [0,1])$ and with $\phi = 0$. $\mathcal{P}_{\texttt{OLIVE}}$ is NP-hard.*

Proof. The proof uses the same family of MDPs $\mathcal{M}$ and set of constraints as the proof of Theorem 122 above. As mentioned there, it is crucial that constraints in Equations (6.28)-(6.29) are added for all clauses and literals but none of the possible constraints of the form in Equation (6.30) that arise from distributions over literal states after taking actions `[try` C_j`]` or `[Solve]`. To prove that `OLIVE` can encounter NP-hard problems, it therefore remains to show that running `OLIVE` on any MDP in $\mathcal{M}$ can generate the exact set of constraints in Equations (6.28)-(6.29).

The specification of `OLIVE` by Jiang, Krishnamurthy, et al. (2017) only prescribes that a constraint for one time step h among all that have sufficiently large average Bellman error is added. It however leaves open how exactly h is chosen and which $f \in \mathcal{F}$ is chosen among all that maximize Problem (6.27). Since this component of the algorithm is under-specified, we choose h and $f \in \mathcal{F}$ in an adversarial manner within the specification, which amounts to adversarial tie breaking in the optimization.

We now provide a run of OLIVE on an arbitrary MDP in $\mathcal{M}$ that generates exactly the set of constraints in Equations (6.28)-(6.29):

- For the first $t \in [m]$ iterations, OLIVE picks any Q-function $f_t \in \mathcal{F}$ with $f_t(s_0, b) = \mathbf{1}\{b = \texttt{[try } C_t\texttt{]}\}$ and $f_t(C_t, b) = 1$ and $f_t(x_i^0, \pi_{f_t}(x_i^0)) = f_t(x_i^1, \pi_{f_t}(x_i^1)) = 0$ for all actions b and $i \in [n]$ and chooses to add constraints for $h = 2$. Since the context distributions is a different C_t for every iteration t, this is a valid choice and generates constraints

$$f(C_t, b) = \frac{f(o_{t,1}^{b(1)}) + f(o_{t,2}^{b(2)}) + f(o_{t,3}^{b(3)})}{3} \qquad \text{if } \pi_f(C_t) = b$$

  for all b.
- For the next n iterations $t = m+1, m+2, \ldots m+n$, OLIVE picks any Q-function $f_t \in \mathcal{F}$ with $f_t(s_0, b) = \mathbf{1}\{b = \texttt{[try } x_{t-m}\texttt{]}\}$ and $f_t(x_{t-m}^0, \pi_{f_t}(x_{t-m}^0)) = f_t(x_{t-m}^1, \pi_{f_t}(x_{t-m}^1)) = 1$ for all b. The only positive average Bellman error occurs in the mixture over literal states at $h = 2$ and therefore constraints

$$\frac{f(x_{t-m}^1, \pi_f(x_{t-m}^1)) + f(x_{t-m}^0, \pi_f(x_{t-m}^0))}{2} = \frac{1}{2}$$

  are added.
- Finally, in iteration $t = m+n+1$, OLIVE picks any $f_t \in \mathcal{F}$ with $f_t(s_0, b) = \mathbf{1}\{b = \texttt{[try } x_1\texttt{]}\}$ and $f_t(x_1^0, \pi_{f_t}(x_1^0)) = f_t(x_1^1, \pi_{f_t}(x_1^1)) = 1/2$ for all actions b. Now there is positive average Bellman error in the initial state s_0 and with $h_t = 1$ the following constraints are added

$$f(s_0, \texttt{[try } c_j\texttt{]}) = \max_b f(C_1, b) - 1/m \qquad \text{if } \pi_f(s_0) = \texttt{[try } c_j\texttt{]}$$

$$f(s_0, \texttt{[solve]}) = \frac{1}{m}\sum_{i=1}^{m} \max_b f(C_j, b) \qquad \text{if } \pi_f(s_0) = \texttt{[solve]}$$

$$f(s_0, \texttt{[try } x_i\texttt{]}) = \frac{f(x_i^0) + f(x_i^1)}{2} \qquad \text{if } \pi_f(s_0) = \texttt{[try } x_i\texttt{]}$$

  for all $i \in [n]$ and $j \in [m]$.

Since at iteration $t = m+n+2$, the set of constraints matches exactly the one in the proof of Theorem 122, OLIVE solves exactly the problem instance described there which solves the given 3-SAT instance. □

Chapter 7

Conclusion

7.1 Future Research Possibilities

We now briefly outline a few immediate directions for future research opportunities based on this dissertation.

Minimax-optimal RL in tabular non-episodic MDPs: All algorithms in this dissertation have been developed and analyzed for the episodic setting with a fixed horizon. For other important non-episodic settings such as discounted infinite horizon MDPs, it is still unknown how to explore in a near-optimal way. To the best of our knowledge, the best known PAC bounds for discounted RL are those by Szita and Szepesvári (2010) and Lattimore and Hutter (2012) which are suboptimal in the dominant ϵ^{-2} term, either in their horizon or state space dependency. It would be interesting to investigate to what extend the insights developed in this dissertation can be transferred to this setting to achieve minimax-optimal regret and PAC bounds.

Problem-dependent guarantees: While the sample-complexity bounds derived in this dissertation are close to the best achievable ones for the finite episodic MDPs, they are still problem-independent bounds. This makes them easily computable without any unknown property of the specific problem instance at hand. A logical next step is to move on and provide similarly strong problem-dependent bounds that can reveal what kind of MDP is more easy and or hard to solve for our algorithms. There has been some very recent initial work on problem-dependent regret bounds for this setting (Zanette and Brunskill, 2018; Zanette and Brunskill, 2019; Simchowitz and Jamieson, 2019) (building in parts on the results of this dissertation). These bounds for example depend on the value gaps between the actions in the states. While highly encouraging, it is not clear yet whether these can be tightened and whether problem-dependent (Uniform-)PAC bounds can be achieved.

Policy certificates in model-free approaches and function approximation: Our `ORLC` and `ORLC-SI` algorithms, the only known methods so far that provide certificates during learning, are both model-based. That means they explicitly maintain a model of the environment (transition probabilities and average instantaneous rewards). This is very sample-efficient but model-free approaches that e.g. only maintain value estimates are typically computationally cheaper and are easier to extend to the function approximation setting. Jin et al. (2018) have shown strong regret bounds for an optimistic version of model-free Q-learning and recently, Efroni et al. (2019) reduced the computational complexity of model-based OFU methods while still maintaining strong sample-efficiency guarantees. However, none of them provides certificates. It would

be highly interesting to investigate how one can provide certificates in a computationally and statistically feasible manner for model-free methods. This is non-trivial because while the optimal Q-function stays fixed throughout all episodes and can be estimated in a optimistic and model-free fashion, the Q-function of the current policy (which we want to lower bound in certificates) can change drastically between episodes. It is therefore an interesting challenge to devise an approach that can quickly adjust to this moving target without access to an explicit empirical model.

Developing such a model-free approach in the tabular setting can also be a good pathway into extending policy certificates to the function-approximation setting. One might not be able to achieve certificates with guaranteed accuracy or guaranteed coverage but they could still provide a very useful tool to make deep reinforcement learning approaches more accountable.

Certificates for other properties of the return distribution: This dissertation proposed policy certificates that tell the user what expected return the current policy will achieve at least and how far from optimal it can be. This is a natural starting point and highly relevant for a wide range of applications. However, there are also high-stakes applications where one might be interested in different properties of the sum of reward distribution, especially those characterizing lower-tails to quantify risk. For example, in a health-care application one might care about how good the outcome is at least in the worst 10% of cases for a particular patient. It would therefore be highly desirable to have policy certificates for such risk-aware properties. However, there is very little existing work on how to strategically explore to optimize such properties in MDPs. In Keramati et al. (2019) we have initial results that provide a principled way for optimism (and pessimism) with respect to a common risk-aware metric called conditional value at risk (CVaR). This is a necessary foundation for policy certificates for CVaR in the future.

No-regret learning in problems with rich observations: The `VALOR` algorithm in this dissertation as well as the OLIVE (Jiang, Krishnamurthy, et al., 2017) algorithm and other methods for problems with rich observations and low Bellman rank all enjoy PAC bounds. By the results in this dissertation, this can be translated to a $T^{2/3}$ high-probability regret bound but to the best of our knowledge, there is no algorithm that achieves $\sqrt{T}$ regret in problems with low Bellman rank. It is an exciting direction to develop such a method both from a theoretical perspective but also from a practical lens. By combining ideas from OLIVE with those from regret minimization in contextual bandits (Agarwal, Hsu, et al., 2014) one can hope to achieve an algorithm with $\sqrt{T}$ regret and which has a more practical nature than the existing approaches for this setting.

7.2 Summary of Contributions

We have motivated the work in this dissertation by making reinforcement learning more sample-efficient and accountable which are key requirements of applying RL in high-stakes applications. Tabular Markov decision processes have long been a staple of theoretical reinforcement learning research as they allow us to focus on the key challenges of RL, partial feedback and long-term consequences, without needing to deal with generalization. There has been a long line of work on developing algorithms with better sample-complexity bounds for tabular MDPs. However, most earlier work focuses on infinite-horizon problems while many high-stakes applications such as treatment optimization or automated tutoring systems have a rather episodic nature (one patient, customer, student corresponds to one episode). For this reason, this dissertation focused on episodic tasks, starting with tabular environments.

In Chapter 3, we began by establishing the first lower bound on the sample complexity of reinforcement learning in tabular episodic MDPs to set a reference point for how sample-efficient we can hope to be. We

have further made a initial attempt at matching this goal with a theoretical algorithm called UCFH that achieves a PAC bound with the optimal scaling with the episode, improving upon any prior work when applied to this setting. The key to this improved sample-complexity is the use of empirical Bernstein concentration bounds to accurately quantify the variance of state transitions. However, this algorithm is far from a practical method as it explicitly depends on loose constants and still has a suboptimal sample complexity in the number of states.

In Chapter 4, we then focused on the state space dependency. We proposed a new algorithm called UBEV and proved that its sample-complexity scales optimally with the number of unique states up to lower-order terms and logarithmic factors. This was made possible because we directly bounded the uncertainty of average optimal next state value instead of the transition probabilities. Unlike UCFH this algorithm turns out to have rather simple structure and essentially uses planning in the empirical model with reward bonuses. The algorithm itself further does not depend on loose constants and leverages time-uniform concentration bounds that achieve the optimal scaling with the number of observations in the reward bonuses. These concentration bounds were the key to be able to prove a new type of sample-complexity bound called Uniform-PAC. We introduced Uniform-PAC for a unified analysis of regret and PAC bounds, i.e., a Uniform-PAC bound implies both, a (mistake-style) PAC and a regret bound. This was necessary because we proved that – against a common belief – converting regret and PAC bounds for the same algorithm only yields suboptimal guarantees, making a direct comparison difficult.

While UBEV is a much more practical algorithm compared to UCFH, it scales worse with the horizon. Azar, Osband, and Munos (2017) have used a combination of the techniques in Chapters 3–4 in their UCBVI algorithm whose regret bound scales better in the horizon than UBEV's and is minimax-optimal but only if the problem horizon is small. In Chapter 5 we addressed this limitation and presented an algorithm called ORLC that achieves minimax-optimal PAC and regret bounds up to lower-order terms even if the horizon is large. To the best of our knowledge, this algorithm has better problem-independent sample-efficiency guarantees than any other algorithm. The overall structure of ORLC is similar to UBEV but it also computes lower confidence bounds on the current value function in addition to the usual upper confidence bounds on the optimal value function. The lower and upper bounds tighten each other and are the key to minimax-optimal reinforcement learning. In a sense, ORLC closes for the episodic setting a long chapter of works that propose improved tabular RL algorithms with tighter problem-independent regret or PAC guarantees.

However, improving sample-efficiency is not the only benefit to computing both upper and lower confidence bounds. They can also be used to output what we call policy certificates before each episode. These certificates tell the user how suboptimal the algorithm can perform in the next episode an allow to intervene if necessary. This makes RL algorithms accountable and we introduced a new learning framework called IPOC to ensure not only the sample-efficiency of policy learning but also accuracy of these certificates. As demonstrated by the guarantees proved for ORLC, IPOC bounds imply Uniform-PAC, PAC and regret bounds and are a first step toward theory for accountable reinforcement learning.

Finally, in Chapter 6, we went beyond the tabular setting and considered problems with rich observations such as images or text where generalization is key to any sample-efficient learning. Here, we presented a new provably sample-efficient algorithm called VALOR for environments with deterministic hidden state dynamics and stochastic rich observations. It operates in an oracle model of computation – accessing policy and value function classes exclusively through standard optimization primitives – and therefore represents computationally efficient alternatives to prior algorithms that require enumeration. Further, with stochastic hidden state dynamics, we proved that the only known sample-efficient algorithm at the time, OLIVE, cannot be implemented in the oracle model. This highlights the computational challenges of provably sample-efficient reinforcement learning in rich observation settings. To conclude while much work remains

this dissertation provides several concrete contributions towards efficient and accountable reinforcement learning.

Bibliography

[1] Yasin Abbasi-Yadkori and Gergely Neu. "Online learning in MDPs with side information". In: *arXiv preprint arXiv:1406.6812* (2014).

[2] Alekh Agarwal, Sarah Bird, Markus Cozowicz, Luong Hoang, John Langford, Stephen Lee, Jiaji Li, Dan Melamed, Gal Oshri, Oswaldo Ribas, et al. "Making contextual decisions with low technical debt". In: *arXiv preprint arXiv:1606.03966* (2016).

[3] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. "Taming the monster: A fast and simple algorithm for contextual bandits". In: *International Conference on Machine Learning*. 2014, pp. 1638–1646.

[4] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. "Reducing multiclass to binary: A unifying approach for margin classifiers". In: *Journal of machine learning research* (2000).

[5] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 2009.

[6] András Antos, Csaba Szepesvári, and Rémi Munos. "Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path". In: *Machine Learning* (2008).

[7] Sanjeev Arora, Elad Hazan, and Satyen Kale. "The Multiplicative Weights Update Method: a Meta-Algorithm and Applications." In: *Theory of Computing* (2012).

[8] Richard C Atkinson. "Optimizing the learning of a second-language vocabulary." In: *Journal of experimental psychology* 96.1 (1972), p. 124.

[9] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. "Exploration–exploitation tradeoff using variance estimates in multi-armed bandits". In: *Theoretical Computer Science* 410.19 (2009), pp. 1876–1902.

[10] Peter Auer. "Using upper confidence bounds for online learning". In: *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*. IEEE. 2000, pp. 270–279.

[11] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* 47.2-3 (2002), pp. 235–256.

[12] Peter Auer, Thomas Jaksch, and Ronald Ortner. "Near-optimal regret bounds for reinforcement learning". In: *Advances in Neural Information Processing Systems*. 2009.

[13] Peter Auer and Ronald Ortner. "Online regret bounds for a new reinforcement learning algorithm". In: (2005).

[14] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. "On the sample complexity of reinforcement learning with a generative model". In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Omnipress. 2012, pp. 1707–1714.

[15] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. "Minimax Regret Bounds for Reinforcement Learning". In: *International Conference on Machine Learning*. 2017, pp. 263–272.

[16] Kamyar Azizzadenesheli, Alessandro Lazaric, and Animashree Anandkumar. “Reinforcement Learning in Rich-Observation MDPs using Spectral Methods”. In: *arXiv:1611.03907* (2016).

[17] Kamyar Azizzadenesheli, Alessandro Lazaric, and Animashree Anandkumar. “Reinforcement learning of POMDPs using spectral methods”. In: *Conference on Learning Theory*. 2016.

[18] J Andrew Bagnell, Sham M Kakade, Jeff G Schneider, and Andrew Y Ng. “Policy search by dynamic programming”. In: *Advances in neural information processing systems*. 2004, pp. 831–838.

[19] Akshay Balsubramani and Aaditya Ramdas. “Sequential nonparametric testing with the law of the iterated logarithm”. In: *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. AUAI Press. 2016, pp. 42–51.

[20] Peter L Bartlett and Ambuj Tewari. “REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs”. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press. 2009, pp. 35–42.

[21] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. “Unifying count-based exploration and intrinsic motivation”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 1471–1479.

[22] Alina Beygelzimer, John Langford, and Pradeep Ravikumar. “Error-correcting tournaments”. In: *International Conference on Algorithmic Learning Theory*. Springer. 2009, pp. 247–262.

[23] Alberto Bietti, Alekh Agarwal, and John Langford. “A contextual bandit bake-off”. In: *arXiv preprint arXiv:1802.04064* (2018).

[24] Jonathan Binas, Leonie Luginbuehl, and Yoshua Bengio. “Reinforcement Learning for Sustainable Agriculture”. In: *ICML 2019 Workshop Climate Change: How Can AI Help?* 2019.

[25] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[26] Ronen I. Brafman and Moshe Tennenholtz. “R-max – a general polynomial time algorithm for near-optimal reinforcement learning”. In: *Journal of Machine Learning Research* (2003).

[27] Ronen I Brafman and Moshe Tennenholtz. “R-max-a general polynomial time algorithm for near-optimal reinforcement learning”. In: *Journal of Machine Learning Research* 3.Oct (2002), pp. 213–231.

[28] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. “Regret analysis of stochastic and nonstochastic multi-armed bandit problems”. In: *Foundations and Trends® in Machine Learning* 5.1 (2012), pp. 1–122.

[29] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. “Large-scale study of curiosity-driven learning”. In: *arXiv preprint arXiv:1808.04355* (2018).

[30] Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume III, and John Langford. “Learning to search better than your teacher”. In: *International Conference on Machine Learning*. 2015.

[31] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. “Top-k off-policy correction for a REINFORCE recommender system”. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM. 2019, pp. 456–464.

[32] Fan Chung and Linyuan Lu. “Concentration inequalities and martingale inequalities: a survey”. In: *Internet Mathematics* 3.1 (2006), pp. 79–127.

[33] Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. “SBEED: Convergent Reinforcement Learning with Nonlinear Function Approximation”. In: *International Conference on Machine Learning*. 2018, pp. 1133–1142.

[34] Christoph Dann and Emma Brunskill. “Sample complexity of episodic fixed-horizon reinforcement learning”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 2818–2826.

[35] Christoph Dann, Katja Hofmann, and Sebastian Nowozin. “Memory Lens: How Much Memory Does an Agent Use?” In: *arXiv preprint arXiv:1611.06928* (2016).

[36] Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. “On Oracle-Efficient PAC Reinforcement Learning with Rich Observations”. In: *Advances in neural information processing systems* (2018).

[37] Christoph Dann, Tor Lattimore, and Emma Brunskill. “Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5713–5723.

[38] Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. “Policy Certificates: Towards Accountable Reinforcement Learning”. In: *International Conference on Machine Learning* (2019).

[39] Markus R Dann and Christoph Dann. “Automated matching of pipeline corrosion features from in-line inspection data”. In: *Reliability Engineering & System Safety* 162 (2017), pp. 40–50.

[40] Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. “Provably efficient RL with Rich Observations via Latent State Decoding”. In: *International Conference on Machine Learning*. 2019, pp. 1665–1674.

[41] Rick Durrett. *Probability: theory and examples*. Cambridge university press, 2010.

[42] Yonathan Efroni, Nadav Merlis, Mohammad Ghavamzadeh, and Shie Mannor. “Tight Regret Bounds for Model-Based Reinforcement Learning with Greedy Policies”. In: *arXiv preprint arXiv:1905.11527* (2019).

[43] Claude-Nicolas Fiechter. “Efficient reinforcement learning”. In: *Proceedings of the seventh annual conference on Computational learning theory*. ACM. 1994, pp. 88–97.

[44] Claude-Nicolas Fiechter. “Expected mistake bound model for on-line reinforcement learning”. In: *ICML*. Vol. 97. 1997, pp. 116–124.

[45] Vincent François-Lavet, Raphael Fonteneau, and Damien Ernst. “How to discount deep reinforcement learning: Towards new dynamic strategies”. In: *arXiv preprint arXiv:1512.02011* (2015).

[46] Aurélien Garivier and Olivier Cappé. “The KL-UCB algorithm for bounded stochastic bandits and beyond”. In: *Proceedings of the 24th annual conference on learning theory*. 2011, pp. 359–376.

[47] Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. “On explore-then-commit strategies”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 784–792.

[48] Mohammad Ghavamzadeh, Marek Petrik, and Yinlam Chow. “Safe policy improvement by minimizing robust baseline regret”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 2298–2306.

[49] Karan Goel, Christoph Dann, and Emma Brunskill. “Sample efficient policy search for optimal stopping domains”. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press. 2017, pp. 1711–1717.

[50] Robert Grande, Thomas Walsh, and Jonathan How. “Sample efficient reinforcement learning with gaussian processes”. In: *International Conference on Machine Learning*. 2014, pp. 1332–1340.

[51] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. “A kernel two-sample test”. In: *Journal of Machine Learning Research* (2012).

[52] Martin Grötschel, László Lovász, and Alexander Schrijver. “The ellipsoid method and its consequences in combinatorial optimization”. In: *Combinatorica* 1.2 (1981), pp. 169–197.

[53] Zhaohan Daniel Guo, Shayan Doroudi, and Emma Brunskill. “A pac rl algorithm for episodic pomdps”. In: *Artificial Intelligence and Statistics*. 2016, pp. 510–518.

[54] Assaf Hallak, Dotan Di Castro, and Shie Mannor. “Contextual Markov Decision Processes”. In: *arXiv:1502.02259* (2015).

[55] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. “Deep reinforcement learning that matters”. In: *arXiv preprint arXiv:1709.06560* (2017).

[56] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. “Vime: Variational information maximizing exploration”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 1109–1117.

[57] Steven R. Howard, Aaditya Ramdas, Jon Mc Auliffe, and Jasjeet Sekhon. “Uniform, nonparametric, non-asymptotic confidence sequences”. In: *arXiv preprint arXiv:1810.08240* (2018).

[58] Daniel Joseph Hsu. “Algorithms for active learning”. PhD thesis. UC San Diego, 2010.

[59] Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. “Fair learning in Markovian environments”. In: *arXiv preprint arXiv:1611.03071* (2016).

[60] Thomas Jaksch, Ronald Ortner, and Peter Auer. “Near-optimal regret bounds for reinforcement learning”. In: *Journal of Machine Learning Research* 11.Apr (2010), pp. 1563–1600.

[61] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. “lil’ucb: An optimal exploration algorithm for multi-armed bandits”. In: *Conference on Learning Theory*. 2014, pp. 423–439.

[62] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. “Contextual Decision Processes with low Bellman rank are PAC-Learnable”. In: *International Conference on Machine Learning*. 2017, pp. 1704–1713.

[63] Nan Jiang and Lihong Li. “Doubly robust off-policy value evaluation for reinforcement learning”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*. JMLR. org. 2016, pp. 652–661.

[64] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. “Is Q-learning Provably Efficient?” In: *arXiv preprint arXiv:1807.03765* (2018).

[65] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. “The Malmo Platform for artificial intelligence experimentation”. In: *International Joint Conference on Artificial Intelligence*. 2016.

[66] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. “Fairness in learning: Classic and contextual bandits”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 325–333.

[67] Sham Kakade. “On the sample complexity of reinforcement learning”. PhD thesis. University College London, 2003.

[68] Sham M. Kakade and John Langford. “Approximately optimal approximate reinforcement learning”. In: *International Conference on Machine Learning*. 2002.

[69] Sham Kakade, Michael J Kearns, and John Langford. “Exploration in metric state spaces”. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003, pp. 306–312.

[70] Sampath Kannan, Michael Kearns, Jamie Morgenstern, Mallesh Pai, Aaron Roth, Rakesh Vohra, and Zhiwei Steven Wu. “Fairness incentives for myopic agents”. In: *Proceedings of the 2017 ACM Conference on Economics and Computation*. ACM. 2017, pp. 369–386.

[71] Michael Kearns and Daphne Koller. “Efficient reinforcement learning in factored MDPs”. In: *International Joint Conference on Artificial Intelligence*. 1999.

[72] Michael Kearns and Satinder Singh. “Near-optimal reinforcement learning in polynomial time”. In: *Machine Learning* (2002).

[73] Ramtin Keramati, Alex Tamkin, Christoph Dann, and Emma Brunskill. "Being Optimistic to Be Conservative: Quickly Learning a CVaR Policy". In: *in preparation* (2019).

[74] Leonid G Khachiyan. "Polynomial algorithms in linear programming". In: *USSR Computational Mathematics and Mathematical Physics* (1980).

[75] J Zico Kolter and Andrew Y Ng. "Near-Bayesian exploration in polynomial time". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 513–520.

[76] Akshay Krishnamurthy, Alekh Agarwal, and John Langford. "PAC Reinforcement learning with rich observations". In: *Advances in Neural Information Processing Systems*. 2016.

[77] John Langford and Alina Beygelzimer. "Sensitive error correcting output codes". In: *International Conference on Computational Learning Theory*. Springer. 2005, pp. 158–172.

[78] John Langford and Tong Zhang. "The epoch-greedy algorithm for multi-armed bandits with side information". In: *Advances in Neural Information Processing Systems*. 2008.

[79] Tor Lattimore and Csaba Czepesvari. *Bandit Algorithms*. Cambridge University Press, 2018.

[80] Tor Lattimore and Marcus Hutter. "PAC bounds for discounted MDPs". In: *International Conference on Algorithmic Learning Theory*. Springer. 2012, pp. 320–334.

[81] Huitian Lei, Inbal Nahum-Shani, K Lynch, David Oslin, and Susan A Murphy. "A" SMART" design for building individualized treatment sequences". In: *Annual review of clinical psychology* 8 (2012), pp. 21–48.

[82] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. "A contextual-bandit approach to personalized news article recommendation". In: *Proceedings of the 19th international conference on World wide web*. ACM. 2010, pp. 661–670.

[83] Lihong Li, Michael L Littman, and Thomas J Walsh. "Knows what it knows: a framework for self-aware learning". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 568–575.

[84] Lihong Li, Thomas J. Walsh, and Michael L. Littman. "Towards a unified theory of state abstraction for MDPs". In: *International Symposium on Artificial Intelligence and Mathematics*. 2006.

[85] Yao Liu, Omer Gottesman, Aniruddh Raghu, Matthieu Komorowski, Aldo A Faisal, Finale Doshi-Velez, and Emma Brunskill. "Representation balancing mdps for off-policy policy evaluation". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2644–2653.

[86] Ashique Rupam Mahmood, Huizhen Yu, and Richard S Sutton. "Multi-step off-policy learning without importance sampling ratios". In: *arXiv preprint arXiv:1702.03006* (2017).

[87] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. "Offline policy evaluation across representations with applications to educational games". In: *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2014, pp. 1077–1084.

[88] Shie Mannor and John N Tsitsiklis. "The sample complexity of exploration in the multi-armed bandit problem". In: *Journal of Machine Learning Research* 5.Jun (2004), pp. 623–648.

[89] Pascal Massart. *Concentration inequalities and model selection*. Springer, 2007.

[90] Andreas Maurer and Massimiliano Pontil. "Empirical Bernstein bounds and sample variance penalization". In: *arXiv preprint arXiv:0907.3740* (2009).

[91] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), p. 529.

[92] Aditya Modi, Nan Jiang, Satinder Singh, and Ambuj Tewari. "Markov Decision Processes with Continuous Side Information". In: *Algorithmic Learning Theory*. 2018, pp. 597–618.

[93] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[94] Rémi Munos and Csaba Szepesvári. "Finite-time bounds for fitted value iteration". In: *Journal of Machine Learning Research* (2008).

[95] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. "Deep exploration via bootstrapped DQN". In: *Advances in neural information processing systems*. 2016, pp. 4026–4034.

[96] Ian Osband, Daniel Russo, and Benjamin Van Roy. "(More) efficient reinforcement learning via posterior sampling". In: *Advances in Neural Information Processing Systems*. 2013, pp. 3003–3011.

[97] Ian Osband and Benjamin Van Roy. "Model-based reinforcement learning and the eluder dimension". In: *Advances in Neural Information Processing Systems*. 2014.

[98] Ian Osband and Benjamin Van Roy. "Why is posterior sampling better than optimism for reinforcement learning?" In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 2701–2710.

[99] Ian Osband, Benjamin Van Roy, Daniel Russo, and Zheng Wen. "Deep exploration via randomized value functions". In: *arXiv preprint arXiv:1703.07608* (2017).

[100] Ian Osband, Benjamin Van Roy, and Zheng Wen. "Generalization and Exploration via Randomized Value Functions". In: *International Conference on Machine Learning*. 2016, pp. 2377–2386.

[101] Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. "Count-based exploration with neural density models". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 2721–2730.

[102] Jason Pazis and Ronald Parr. "Efficient PAC-Optimal Exploration in Concurrent, Continuous State MDPs with Delayed Updates." In: *AAAI*. 2016, pp. 1977–1985.

[103] Jason Pazis and Ronald Parr. "PAC Optimal Exploration in Continuous Space Markov Decision Processes." In: *AAAI*. 2013.

[104] Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. "Safe Policy Iteration". In: *PInternational Conference on Machine learning*. 2013, pp. 307–315.

[105] Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

[106] Manish Raghavan, Aleksandrs Slivkins, Jennifer Wortman Vaughan, and Zhiwei Steven Wu. "The Externalities of Exploration and How Data Diversity Helps Exploitation". In: *arXiv preprint arXiv:1806.00543* (2018).

[107] Spyros Reveliotis and Theologos Bountourelis. "Efficient PAC learning for episodic tasks with acyclic state spaces". In: *Discrete Event Dynamic Systems* (2007).

[108] Stephane Ross. "Interactive learning for sequential decisions and predictions". PhD thesis. Carnegie Mellon University, 2013.

[109] Stephane Ross and J Andrew Bagnell. "Reinforcement and imitation learning via interactive no-regret learning". In: *arXiv:1406.5979* (2014).

[110] Dan Russo and Benjamin Van Roy. "Eluder dimension and the sample complexity of optimistic exploration". In: *Advances in Neural Information Processing Systems*. 2013.

[111] Daniel Russo. "Worst-Case Regret Bounds for Exploration via Randomized Value Functions". In: *arXiv preprint arXiv:1906.02870* (2019).

[112] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. "A tutorial on thompson sampling". In: *Foundations and Trends® in Machine Learning* 11.1 (2018), pp. 1–96.

[113] Daniel Russo and Benjamin Van Roy. "Learning to optimize via information-directed sampling". In: *Advances in Neural Information Processing Systems*. 2014, pp. 1583–1591.

[114] Touqir Sajed, Wesley Chung, and Martha White. "High-confidence error estimates for learned value functions". In: *arXiv preprint arXiv:1808.09127* (2018).

[115] Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[116] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. "Mastering the game of Go without human knowledge". In: *Nature* 550.7676 (2017), p. 354.

[117] Max Simchowitz and Kevin Jamieson. "Non-Asymptotic Gap-Dependent Regret Bounds for Tabular MDPs". In: *arXiv preprint arXiv:1905.03814* (2019).

[118] Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. "Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 105–133.

[119] Matthew J Sobel. "The variance of discounted Markov decision processes". In: *Journal of Applied Probability* 19.4 (1982), pp. 794–802.

[120] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. "Gaussian process optimization in the bandit setting: No regret and experimental design". In: *Proceedings of the International Conference on Machine Learning, 2010*. 2010.

[121] Alexander L. Strehl, Lihong Li, and Michael L. Littman. "Reinforcement learning in finite MDPs: PAC analysis". In: *Journal of Machine Learning Research* (2009).

[122] Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. "PAC model-free reinforcement learning". In: *International Conference on Machine Learning*. 2006.

[123] Alexander L. Strehl and Michael L. Littman. "A theoretical analysis of model-based interval estimation". In: *International Conference on Machine learning*. 2005.

[124] Alexander L Strehl and Michael L Littman. "An analysis of model-based interval estimation for Markov decision processes". In: *Journal of Computer and System Sciences* 74.8 (2008), pp. 1309–1331.

[125] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. "Model-based reinforcement learning in contextual decision processes". In: *arXiv preprint arXiv:1811.08540* (2018).

[126] István Szita and Csaba Szepesvári. "Model-based reinforcement learning with nearly tight exploration complexity bounds". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 1031–1038.

[127] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. "# Exploration: A study of count-based exploration for deep reinforcement learning". In: *Advances in neural information processing systems*. 2017, pp. 2753–2762.

[128] Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. "High-confidence off-policy evaluation". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[129] Philip Thomas and Emma Brunskill. "Data-efficient off-policy policy evaluation for reinforcement learning". In: *International Conference on Machine Learning*. 2016, pp. 2139–2148.

[130] Philip Thomas, Christoph Dann, and Emma Brunskill. "Decoupling Gradient-Like Learning Rules from Representations". In: *International Conference on Machine Learning*. 2018, pp. 4924–4932.

[131] Philip Thomas, Bruno Castro Silva, Christoph Dann, and Emma Brunskill. "Energetic natural gradient descent". In: *International Conference on Machine Learning*. 2016, pp. 2887–2895.

[132] Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. "High confidence policy improvement". In: *International Conference on Machine Learning*. 2015, pp. 2380–2388.

[133] Leslie G Valiant. "A theory of the learnable". In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM. 1984, pp. 436–445.

[134] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M. Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Yuhuai Wu, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. *AlphaStar: Mastering the Real-Time Strategy Game StarCraft II*. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/. 2019.

[135] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. "Inequalities for the L1 deviation of the empirical distribution". In: *Hewlett-Packard Labs, Tech. Rep* (2003).

[136] Zheng Wen and Benjamin Van Roy. "Efficient exploration and value function generalization in deterministic systems". In: *Advances in Neural Information Processing Systems*. 2013.

[137] Zheng Wen and Benjamin Van Roy. "Efficient Reinforcement Learning in Deterministic Systems with Value Function Generalization". In: *Mathematics of Operations Research* (2017).

[138] Andrew G Wilson, Christoph Dann, Chris Lucas, and Eric P Xing. "The human kernel". In: *Advances in neural information processing systems*. 2015, pp. 2854–2862.

[139] Andrew Gordon Wilson, Christoph Dann, and Hannes Nickisch. "Thoughts on massively scalable Gaussian processes". In: *arXiv preprint arXiv:1511.01870* (2015).

[140] A. Zanette and E. Brunskill. "Tighter Problem-Dependent Regret Bounds in Reinforcement Learning without Domain Knowledge using Value Function Bounds". In: *https://arxiv.org/abs/1901.00210* (2019).

[141] Andrea Zanette and Emma Brunskill. "Problem dependent reinforcement learning bounds which can identify bandit structure in mdps". In: *International Conference on Machine Learning*. 2018, pp. 5732–5740.

[142] Shlomo Zilberstein and Stuart Russell. "Optimal composition of real-time systems". In: *Artificial Intelligence* 82.1-2 (1996), pp. 181–213.

[143] Barret Zoph and Quoc V Le. "Neural architecture search with reinforcement learning". In: *arXiv preprint arXiv:1611.01578* (2016).